

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Острозька академія»
Економічний факультет

Кафедра економіко-математичного моделювання та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА/ПРОЄКТ
на здобуття освітнього ступеня бакалавра

на тему: **«РОЗРОБКА ІГРОВОГО ЗАСТОСУНКУ В ЖАНРІ ROGUELIKE ПІД
МОБІЛЬНУ СИСТЕМУ ANDROID НА РУШІЮ UNITY»**

Виконав: студент 4 курсу, групи КН-41
першого (бакалаврського) рівня вищої освіти
спеціальності 122 Комп'ютерні науки
освітньо-професійної програми «Комп'ютерні науки»
Лящук Сергій Олександрович

Керівник:
Гаврильчик Леонід Сергійович

Рецензент:
Місай Володимир Віталійович

РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ

Завідувач кафедри економіко-математичного моделювання та інформаційних
технологій _____ (проф., д.е.н. Кривицька О.Р.)

Протокол № ____ від « ____ » _____ 2022 р.

Міністерство освіти і науки України
Національний університет «Острозька академія»

Факультет: економічний

Кафедра: економіко-математичного моделювання та інформаційних технологій

Спеціальність: 122 Комп'ютерні науки

Освітньо-професійна програма: Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри

Ольга КРИВИЦЬКА

«___» _____ 20__ р.

ЗАВДАННЯ
на кваліфікаційну роботу/проєкт студента

Лящука Сергія Олександровича

1. Тема роботи: Розробка ігрового застосунку в жанрі roguelike під мобільну систему Android на рушію Unity

керівник роботи/проєкту: Гаврильчик Леонід Сергійович

Затверджено наказом ректора НаУОА від 29 жовтня 2021 року №110

2. Термін здачі студентом закінченої роботи/проєкту: 03 червня 2022 року

3. Вихідні дані до роботи/проєкту: постановка задачі, аналіз наявних аналогів, вихідні дані.

4. Перелік завдань, які належить виконати: опис предметного середовища; огляд наявних аналогів; постановку задачі; опис архітектури рішення, що розробляється; опис коду та інтерфейсу програми; тестування застосунку.

5. Перелік графічного матеріалу: макети, рисунки, таблиці, діаграми.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Гаврильчик Л.С.	01.12.2021р.	01.12.2021р.
2	Гаврильчик Л.С.	01.12.2021р.	01.12.2021р.
3	Гаврильчик Л.С.	01.12.2021р.	01.12.2021р.

7. Дата видачі завдання: 01.12.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи/проєкту	Строк виконання етапів	Примітка
1	Затвердження теми роботи/проєкту	до 01.11.2021 р.	
2	Постановка завдання	до 01.12. 2021 р.	
3	Розробка архітектури та загальної структури системи	до 01.02.2022 р.	
4	Розробка структур окремих підсистем	до 01.03. 2022 р.	
5	Програмна реалізація системи	до 01.05.2022 р.	
6	Попередній захист кваліфікаційної роботи/проєкту	до 01.06.2022р.	
7	Здача кваліфікаційної роботи/проєкту на кафедру	03.06.2022 р.	

Студент: _____ Сергій ЛЯЦУК

Керівник кваліфікаційної роботи/проєкту: _____ Леонід ГАВРИЛЬЧИК

АНОТАЦІЯ
кваліфікаційної роботи
на здобуття освітнього ступеня бакалавра

Тема: *Розробка ігрового застосунку в жанрі roguelike під мобільну систему Android на рушію Unity*

Автор: *Лящук Сергій Олександрович*

Науковий керівник: *Гаврильчик Леонід Сергійович*

Захищена «.....»..... 20__ року.

Пояснювальна записка до кваліфікаційної роботи: *47 с., 25 рис., 2 табл., 14 джерел.*

Ключові слова: *Unity, 2D, ігровий застосунок, ігровий рушій*

Короткий зміст праці:

У кваліфікаційній роботі на тему: «Розробка ігрового застосунку в жанрі roguelike під мобільну систему Android на рушію Unity» описано створення 2D гри з можливістю користування у системі Android. Дана тема була обрана у зв'язку щорічним ростом популярності мобільних ігрових додатків. В процесі розробки використано наступні засоби: ігровий рушій Unity, середовище розробки Microsoft Visual Studio 2022, мова програмування C#.

The qualification work on the topic: "Development of a game application in the genre of roguelike for the Android mobile system on the Unity engine" describes the creation of a 2D game with the ability to use in the Android system. This topic was chosen in connection with the annual growth in popularity of mobile gaming applications. The following tools were used in the development process: Unity game engine, Microsoft Visual Studio 2022 development environment, C # programming language.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ	10
1.1 Обґрунтування вибору середовища розробки	10
1.1.1 Вибір ігрового рушія	10
1.1.2 Асети для ігрового застосунку	16
1.1.2 Порівняльний аналіз з обраного програмного забезпечення	16
1.2. Опис предметного середовища	17
1.3. Огляд існуючих платформ	18
1.4. Огляд наявних аналогів	19
Висновок до розділу 1	24
РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	25
2.1 Аналіз предметної області	25
2.2 Постановка задачі	26
2.3 Розробка архітектури ігрового додатку	26
2.4 Проектування інтерфейсу гри	27
2.5. Система меню	28
Висновок до розділу 2	32
РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	33
3.1. Засоби розробки	33
3.2. Вимоги до технічного та програмного забезпечення	33
3.3 Опис програмної реалізації	33
3.3.1. Створення меню	33
3.3.2. Створення кімнат ігрового рівня	35
3.3.3. Інтерфейс	36
3.3.4. Розробка Player та TouchAndGo	37
3.3.5. Камера та звуки	40

	6
3.3.6. Розробка Енету	41
3.3.7. Бойова система	41
3.3.8. Робота над логікою гри	44
3.3.9. Ліцензії та авторські права на асет текстур, музику	45
Висновок до розділу 3	46
Розділ 4	
Тестування	47
4.1 Функціональне тестування	47
4.2. Usability Test	48
Висновок до розділу 4	48
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Debug — відлагоджувач, комп'ютерна програма, яка використовується для тестування і виправлення вад інших програм.

IDE — інтегроване середовище розробки.

UML — уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування.

Асет — це компоненти, які є графікою, звуковим супроводом або скриптами. Вони прикріплюються до об'єктів і становлять важливу частину гри.

Префаб — це набір заздалегідь готових ігрових об'єктів та компонентів, які використовуються більш ніж один раз за всю гру.

Скрипт (сценарій) — це невелика програма, яка містить послідовність дій, створених для автоматичного виконання завдання.

Спрайт — це двомірне зображення, що застосовується в комп'ютерній графіці.

Triple-A — це умовна множина ігор середніх та великих компаній з великим бюджетом.

ВСТУП

Життя людини дуже стрімко почало змінюватись в другій половині 17-го століття з початком промислової революції, адже людина почала заміняти ручну працю автоматизованими станками та машинами. Ще швидше життя людини почало змінюватись починаючи з 40-х років 20-го століття, коли було винайдено перший комп'ютер який дав змогу здійснювати точні розрахунки швидше та якісніше, уникаючи багатьох помилок, що позитивно позначилось на розвитку технологій в цілому. В 1958 році фізик Віллі Хігінботем створив першу відеогру це було першим кроком до зародження відео ігрової індустрії.

З кожним роком комп'ютерні технології все більше розвивали, вони ставали все досконалішими, доступнішими та більш технологічними і в 1972 році було представлено громадськості першу ігрову приставку Magnavox Odyssey, при підключенні до телевізора вона давала змогу грати в 12 вбудованих ігор. Так з'явився новий вид розваг який нам відомий як відео ігри. Згодом комп'ютери стали доступні для як побутова техніка, це позитивно вплинуло на їх розвиток, адже більше людей стало користуватись ними. В 1980-х роках почали з'являтися так звані зали з аркадними ігровими автоматами, де можна було пограти в певні ігри. Популярність комп'ютерних ігор швидко росла так само ріс і ринок ігрової індустрії.

Швидкість темпу життя людини у 21 столітті значно виросла, адже з'явилися нові потреби та нові професії, було автоматизовано значну кількість виробничих та повсякденних справ, людина намагається заповнити кожну хвилину свого часу потрібними справами та це також торкається і відпочинку. Галузь мобільних телефонів дуже розвинута, тепер завдяки ним можна не тільки спілкуватись, а також фотографувати, слухати музику, здійснювати пошук в інтернеті та навіть грати. Найпоширенішою операційною системою для мобільних телефонів є система Android, тому, що вона є безкоштовною, постійно вдосконалюється, підтримується на телефонах різних виробників. У зв'язку з високою популярністю мобільних телефонів, стали популярними ігрові додатки. Ринок мобільних ігрових додатків

росте щороку, у 2020 році він займав 33% від усього ринку ігор. Саме тому розробка ігрових додатків є дуже прибутковою та популярною.

Мета дослідження – створення ігрового програмного продукту.

Задачі дослідження:

- Аналіз предметної області;
- проектування концепту ігрового застосунку;
- вибір інструментів для розробки, методів реалізації та тестування програмного продукту.

Об'єктом дослідження є процес аналізу мобільного геймінгу у жанрі roguelike, вивчення ігрових механік.

Предметом дослідження є технології: ігровий рушій Unity, мова програмування C#.

Для вирішення завдання, реалізації мобільного ігрового додатка необхідно здійснити:

- опис предметного середовища;
- огляд наявних аналогів;
- постановку задачі;
- опис архітектурних рішень;
- опис інтерфейсу, коду програми;
- тестування.

РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Обґрунтування вибору середовища розробки

1.1.1 Вибір ігрового рушія

Ігровий рушій (Game engine) — це програма яка є центральною частиною будь-якої відеогри, вона відповідає за всю технічну сторону, забезпечує взаємодію основних технологій, спрощує розробку гри, дозволяє здійснювати перевірку та тестування на етапі розробки. Також важливо зазначити, що деякі ігрові рушії дають змогу розробляти кросплатформені застосунки які можна запускати на різних пристроях таких як мобільні телефони, планшети, персональні комп'ютери, ігрові консолі.

Першим і одним з головних питань які виникають на початку розробки є питання вибору середовища та платформи. Для створення ігрового додатку є велика кількість, ігрових рушіїв, з кожним роком їх стає все більше. Не великий перелік ігрових рушіїв: Unreal Engine, CryEngine, Havok, Godot, Unity, всі вони мають певні переваги та недоліки. За користування певними ігровими рушіями необхідно сплачувати щомісячну підписку або відсоток від реалізації продукту, також є безкоштовні рішення. Деякі дають змогу створювати кросплатформені ігрові застосунки, це означає, створений продукт на такому ігровому рушію можна перенести на іншу платформу. Багато компаній створюють власні ігрові рушії, які більш орієнтовані на потреби їх продукції, це робить компанію не залежною від технологій інших компаній проте має і свої мінуси, часто це обмежує платформу і вимагає від працівників вивчення нового підходу та нової технології.

Розглянемо наступні ігрові рушії для створення ігрових додатків: Godot, Unreal Engine, CryEngine, Unity.

Godot – це ігровий рушій загального призначення який був розроблений для підтримки різного роду проектів. Він є багатofункціональним, кросплатформним, надає можливість створювати 2D та 3D ігри, підтримує повний набір інструментів, це допомагає користувачу створити бажаний продукт. Засобами Godot користувач

може легко експортувати гру на потрібну йому платформу, включаючи мобільні платформи такі як iOS, Android, настільні платформи Windows, macOS, Linux, консолі та веб-платформу HTML5.

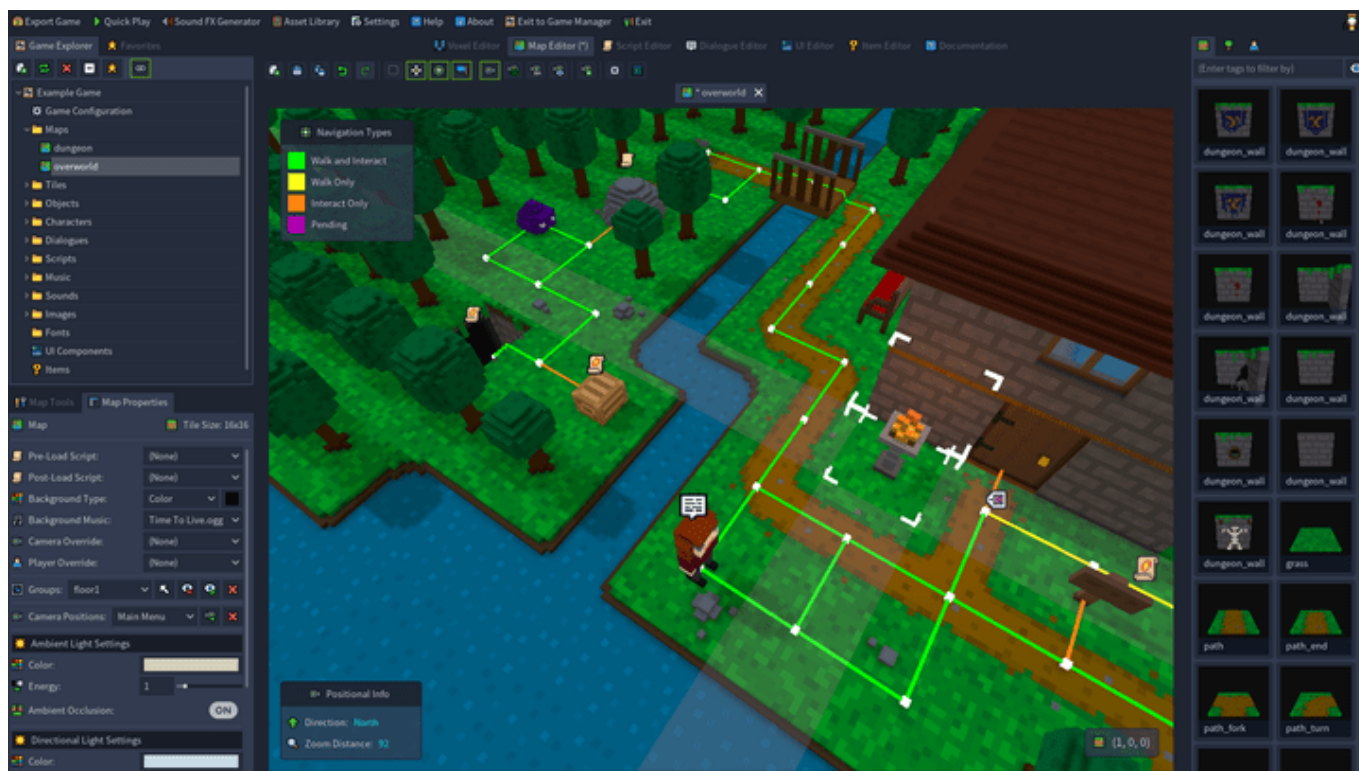


Рис. 1.1 – Інтерфейс Godot

До переваг Godot можна зазначити те, що він має відкритий вихідний код, та поширюється за безкоштовною ліцензією MIT. Згідно ліцензії MIT ігри розробників належать розробникам, в свою чергу Godot вимагає лише розміщення наступного тексту будь де у грі:

«Ця гра використовує Godot Engine, доступний за такою ліцензією:

Авторське право (с) 2007-2022 Хуан Лінецький, Аріель Манзур. Авторське право (с) 2014-2022 учасників Godot Engine.»

Розробкою Godot займається спільнота користувачів це робить її повністю незалежною і дає змогу розвивати та формувати рушій згідно їхніх потреб та очікувань.

До особливостей слід зазначити що Godot завдяки зручному та простому інтерфейсу який є інтуїтивно зрозумілим, має велику та зручну документацію переведену різними мовами. Скрипти для Godot можна писати за допомогою C#, GDScript або використовуючи візуальний скриптинг. GDScript є вбудованою мовою програмування для Godot, її перевагами є простота, синтаксис найбільш схожий на Python, вона легко піддається вивченню, це полегшує поріг входження. Тому розробляти додатки за допомогою Godot можуть як досвідчені розробники які володіють мовою C# так і початківці які тільки знайомляться з розробкою цифрових ігрових продуктів.

Unreal Engine – це ігровий рушій який був розроблений та підтримується компанією Epic Games. На даний момент є одним з найбільш прогресивних ігрових рушіїв на ринку, в основному використовується великими компаніями для створення Triple-A проєктів, оскільки за допомогою його засобів можна створити будь-яку гру, 2D чи 3D, дає змогу додати мережеву гру, вміє взаємодіяти з текстурами високої якості, підтримує можливість вдосконалення завдяки встановленню плагінів має хорошу оптимізацію, надає можливість створювати ігрові застосунки для більшості операційних систем, мобільних платформ та консолей. Для написання скриптів на ігровому рушію Unreal Engine можна використовувати такі мови програмування як UnrealScript, C++, C#, також можна використовувати візуальні редактори такі як BluePrint. BluePrint. З недоліків можна виділити умови ліцензії які змушують сплачувати певний відсоток від прибутку який залежить від суми прибутку, документація написана на англійській мові і є не дуже зручною, до масиву інструментів доведеться довго звикати адже в них дуже легко загубитись, поріг входження для користувача знаходиться на вищому рівні на відміну від інших ігрових рушіїв. Новачку який не володіє навичками програмування буде складніше почати своє знайомство з усіма функціями Unreal Engine, для професіоналів з галузі розробки ігор які вже володіють досвідом в розробці, знають декілька мов програмування та бажають створити Triple-A проєкт буде складно знайти кращий інструмент.

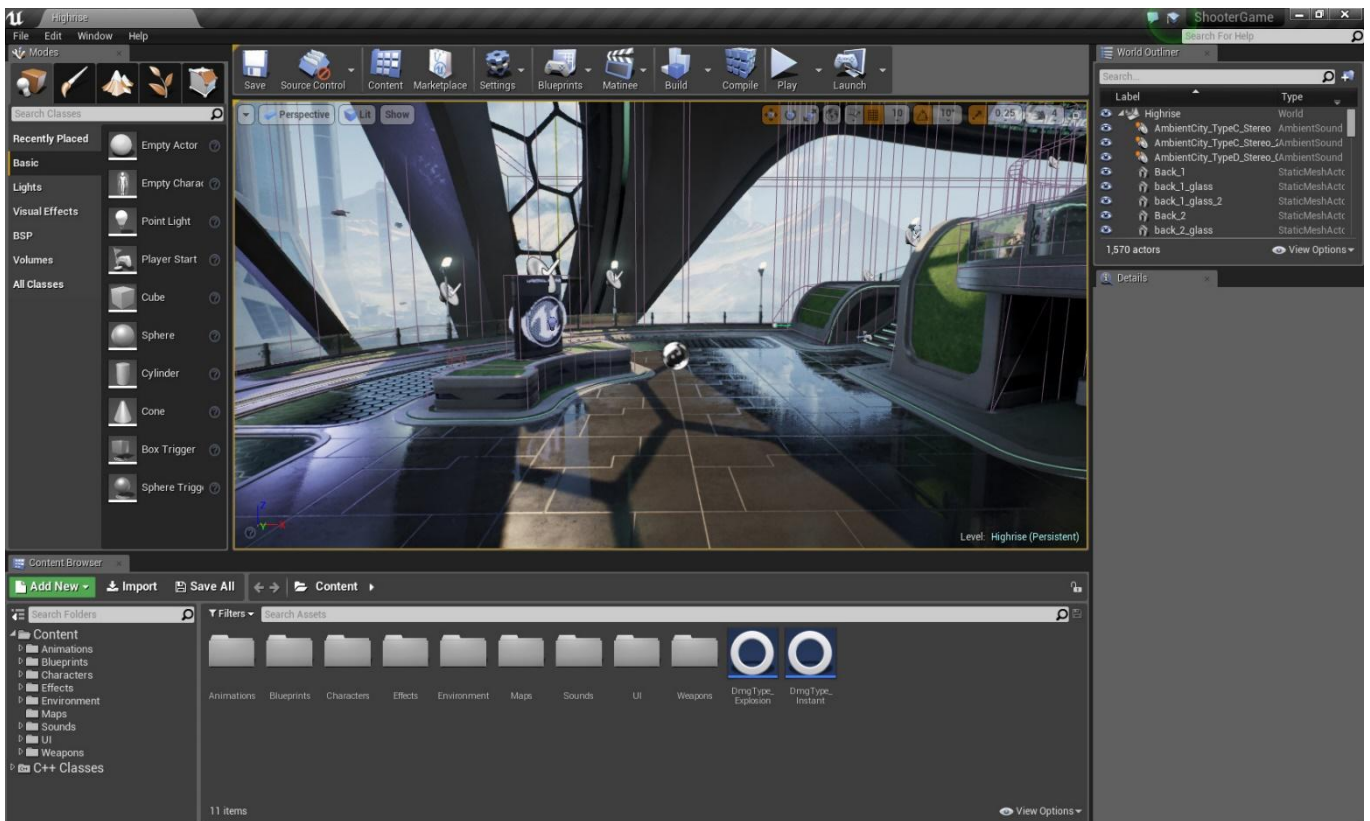


Рис. 1.2 – Інтерфейс Unreal Engine

CryEngine – це ігровий рушій створений компанією Crytek, сьогодні він належить компанії Ubisoft. Він є кросплатформним, дозволяє працювати з графікою на високому рівні, чудово себе зарекомендував у створенні мережових ігор. CryEngine є ряд недоліків які роблять його зовсім не придатним для освоєння початківців в сфері розробки ігор, а саме не зручна і не повна документація, часті проблеми при компіляції проекту, відсутність підтримки яка могла б виправити помилки які виникають в коді самого ігрового рушія а не коді гри, висока вартість ліцензії для користувача, складний інтерфейс. Даний ігровий рушій вимагає знання технології C++ та Flow Graph які є зовсім не привітними до новачків через свою

складність.

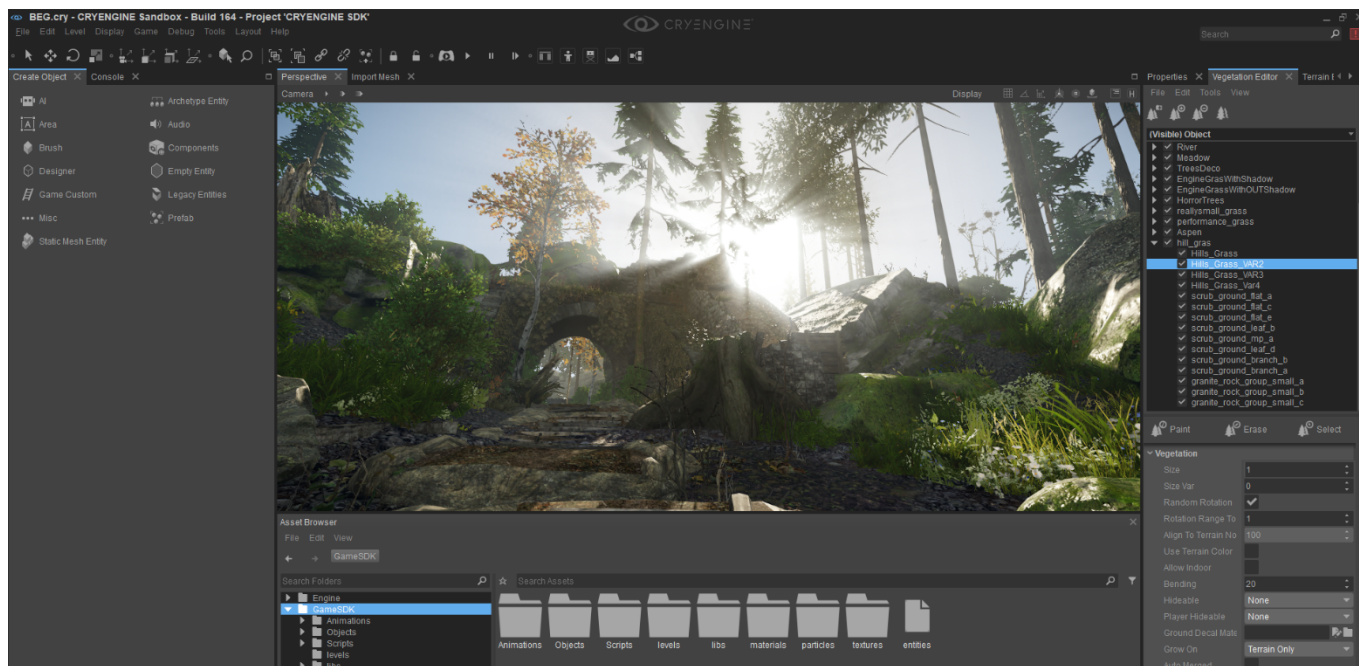


Рис. 1.3 – Інтерфейс CryEngine

Unity – це ігровий рушій створений компанією Unity Technologies в 2005 році, з тих пір компанія постійно підтримує та розвиває свій продукт. Unity є кросплатформним, він підтримується на 25 різноманітних платформах, включаючи ігрові консолі, мобільні пристрої, інтернет-застосунки, персональні комп'ютери та інші.

До переваг Unity слід віднести зручний та інтуїтивно зрозумілий інтерфейс, обширну актуальну документацію та її переклад декількома мовами, зручний магазин асетів, різноманітність плагінів, кросплатформенна підтримка, можливість тестувати ігровий застосунок на кожному етапі створення, велика кількість інструментів для роботи з 2D та 3D, безкоштовність ліцензії для початківців з не високим прибутком та разова оплата ліцензії при умові що прибуток досягає 200 000\$ на рік. Unity підтримує скрипти написані на мові C#, це знижує поріг входження, адже C# є високорівневою та дуже популярною мовою програмування, з великою кількістю відео уроків та зручною документацією, саме це робить Unity

найкращим вибором для новачків в галузі розробки ігрових додатків.

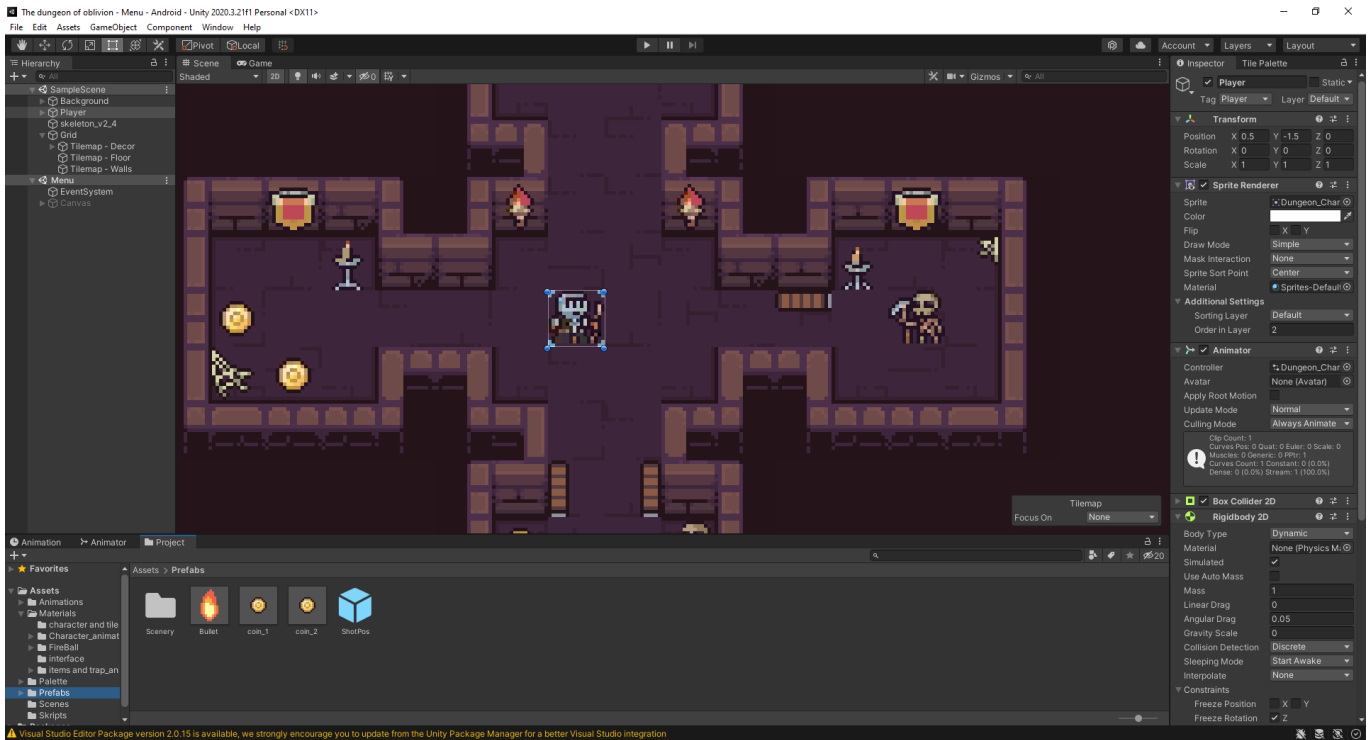


Рис. 1.4 – Інтерфейс Unity

На етапі вибору середовища розробки, ми зупиняємось поміж Unity та Godot, оскільки вони поширюються за безкоштовною ліцензією та є значно зручнішими та простішими для новачків ніж CryEngine та Unreal Engine які в свою чергу вимагають початкових вкладень та не малого досвіду розробки.

До переваг Unity слід віднести те що він є кросплатформенний, зручний, простий в освоєні, підходить для створення 2D та 3D застосунків, багатофункціональний, зі зручною документацією, великою кількістю відео уроків, хорошою підтримкою, великим магазином асетів та плагінів. Даний ігровий рушій ми вивчали під час проходження навчального курсу дисципліни «Розробки ігрових додатків на Unity», цей курс дав загальне поняття та практичні навички у використанні Unity.

Перевагами Godot над Unity є більш зручне перенесення між платформами, та можливість використовувати GDScript замість C#, проте він програє в функціональності та оптимізації, має меншу кількість інструментів.

Взявши до уваги всі переваги та недоліки обох ігрових рушіїв було прийнято рішення використовувати Unity, головною перевагою виступає наявність досвіду.

1.1.2 Асети для ігрового застосунку

Через повну відсутність здібностей до малювання та проєктування 3D моделей було прийнято рішення скористатись готовим 2D асетом графіки при створенні ігрового застосунку. Після не довгого пошуку було обрано асет «Dungeon» автора Pixel_Roem який розмістив його на сайті itch.io на правах вільної ліцензії та зазначив, що всі охочі можуть користуватись його роботою у своїх проєктах та на свій розсуд. В даному асеті є майже всі необхідні ресурси для створення гри такі як спрайти персонажей, оточення, декорацій, елементи інтерфейсу.

На сайті itch.io було знайдено асет «Pixel Art Fire Ball» автором якого є Matheus Frantz, він також розмістив свою роботу на правах вільної ліцензії наголосивши що кожен охочий може змінювати та використовувати дану роботу у своїх проєктах, автор заборонив пере розповсюджувати свою роботу на інших ресурсах.

Музику для фону було взято на сайті soundcloud, «Tormented Souls», автор Муун дозволив використати її для розробки ігрового застосунку.

1.1.2 Порівняльний аналіз з обраного програмного забезпечення

Взявши до уваги вище описані програми, їх переваги та недоліки було прийнято рішення використовувати ігровий рушій Unity та мову програмування C#.

Написання коду скриптів буде відбуватися в середовищі Visual Studio 2022.

2D графіку буде взято з сайту itch.io, подальше редагування її буде проведено засобами Unity, деякі плагіни буде взято з Unity Asset Store. Для зручності порівняльний аналіз з обраного програмного забезпечення наведено у таблиці 1.1.

Таблиця 1.1

Категорія	Варіанти програмного забезпечення	Висновок
Ігровий рушій	Godot, Unreal Engine, CryEngine, Unity	Unity зручний інтерфейс, низький порог входження, надає доступ до магазину асетів. Має обширну документацію, багато відео уроків.
Мова програмування	C++, C#, GDScript, UnrealScript	C# як мова програмування була визначена ігровим рушієм
IDE	Visual Studio 2022, VS Code	Visual Studio 2022 – зручна багатофункціональна IDE яка чудово підходить для роботи з мовою C#, дає змогу зручно переміщатись між вкладками, функція Debug допомагає виявити причину проблеми та швидко її виправити.
Додаткові ресурси	Unity Asset Store	Unity Asset Store – це магазин ресурсів який також надає безкоштовні асети та плагіни. Надає доступ до різноманітних ресурсів таких як моделі, анімації, навчальні посібники, текстури.

1.2. Опис предметного середовища

Unity – універсальний кросплатформений ігровий рушій який постійно вдосконалюється та підтримується на 25 різних операційних системах.

Unity надає широкі функціональні можливості, це дозволяє використовувати рушій в розробці як 2D так і 3D ігрових застосунків. Її широкий спектр можливостей також дозволяє здійснювати первинну обробку текстур та певних ресурсів, а велика кількість посібників часто пропонує готове рішення проблеми яка може виникнути в процесі розробки ігрового застосунку.

Можна виділити лише декілька суттєвих недоліків Unity: робота із зовнішніми бібліотеками є складнішою адже їх взаємодію користувачеві необхідно налаштувати

самотужки, також виникають проблеми під час роботи з схемами які містять багато компонентів.

1.3. Огляд існуючих платформ

На ринку мобільних телефонів найпопулярнішою операційною системою є Android з вагомих конкурентів для цієї можна виділити лише iOS.

iOS – це операційна система для мобільних телефонів компанії Apple, вона є другою по популярності після Android, її код є закритим, що ускладнює розробку ігрових застосунків під дану платформу проте робить її більш захищеною.

Android – це мобільна операційна система з відкритим кодом яка є дуже гнучкою, та безкоштовною, зручною у використанні. За оцінками система Android має понад 3 мільярди користувачів це значно перевищує кількість користувачів iOS.

Переваги розробки ігрових додатків під операційну систему Android:

- велика кількість користувачів;
- хороший прибуток за рахунок реклами;
- велика кількість посібників з розробки;
- значно дешевша вартість публікації в магазині додатків;
- мала кількість обмежень.

Недоліки розробки ігрових додатків під операційну систему Android:

- необхідність оптимізації під різні екрани та пристрої з різною комплектацією;

Переваги розробки ігрових додатків під операційну систему iOS:

- вищий дохід з рядового клієнта;
- стабільність системи.

Недоліки розробки ігрових додатків під операційну систему iOS:

- значно менша кількість користувачів;
- необхідність купувати підписку та поновлювати її;

- необхідність в певному обладнанні для кращого тестування;
- велика кількість обмежень щодо контенту програми;
- високі критерії до програми яку бажаєте завантажити в магазин додатків в App Store.

Google Play – це магазин додатків для платформи Android, аби розмістити гру в даному магазині необхідно здійснити реєстрацію та разову оплату в розмірі 25\$, після подання додатку на публікацію його буде перевірено та якщо він відповідає критеріям, розміщено у магазині.

App Store – це внутрішній магазин для програмних продуктів під операційну систему iOS, для публікації необхідно зареєструватись як фізична особа, оформити підписку вартістю 99\$, після перевірки яка зазвичай вона займає від 3 тижнів, можна буде перейти до наступного етапу після якого відкриється можливість здійснювати запити на публікацію. Вимоги до програмних продуктів які публікуються в App Store значно вищі ніж у Google Play.

Варто зазначити що інтеграція реклами у свої продукти значно простіше здійснити в додатки орієнтовані під платформу Android, за рахунок великої кількості користувачів, реклама на даній платформі приносить значно більше доходів ніж реклама з пристроїв під керуванням iOS.

У кваліфікаційній роботі для створення ігрового застосунку було обрано платформу Android, оскільки вона має ряд переваг над iOS. Вирішальними пунктами на користь Android є «велика кількість користувачів», «мала кількість обмежень», «значно дешевша вартість публікації в магазині додатків».

1.4. Огляд наявних аналогів

Аналіз схожих за ідеєю ігрових додатків, їх механік, дизайну допомагає покращити власний продукт удосконаливши його в тих чи інших аспектах. На аналіз було обрано наступні ігрові додатки:

- Remixed Dungeon: Pixel Rogue
- Pocket Rogues

- Caves (Roguelike)
- Rogues` Tales: Action RPG

Remixed Dungeon: Pixel Rogue – це рольова 2D гра в піксельному стилі яка наслідує Rogue (це була перша гра жанру roguelike). Особливістю гри є те, що кожна ігрова сесія є певною мірою унікальною, кожен рівень генерується випадково, предмети змінюють свої властивості в кожній новій ігровій сесії, гра не дає змоги зберегти прогрес аби потім відновити його, після смерті персонажа, він втрачає всі артефакти, предмети та рівні, все слід розпочинати спочатку. В грі просте покрокове керування, зручний інтерфейс, керується персонаж дотиком, переміщається в ту точку яку вказав гравець.



Рис. 1.5 – Ігровий процес Remixed Dungeon: Pixel Rogue

Pocket Rogues – це динамічна Action-RPG гра з видом зверху в піксельному стилі та жанрі roguelike. В цій грі гравцеві доведеться пробиватися через орди монстрів, досліджувати унікальні, випадково згенеровані локації аби розвинути власну фортецю та героїв. Керування здійснюється завдяки джойстику та екранним кнопка які позначають «напад» та «захист».



Рис. 1.6 – Ігровий процес Pocket Rogues

Caves (Roguelike) – це класична покрокова гра в жанрі roguelike з піксельною графікою, головною особливістю гри є механіка копання печер, та їх дослідження. В цій грі поєднається магія і високі технології. Гра пропонує створити свого унікального персонажа та обіцяє новий досвід при кожній ігровій сесії, це є можливим завдяки випадковій генерації печер що і зацікавлює кожен раз досліджувати їх.



Рис. 1.7 – Ігровий процес Caves (Roguelike)

Rogues` Tales: Action RPG – це 2D гра в піксельному стилі, жанру Action RPG з елементами roguelike. Ціль гри перемогти всіх ворогів та виграти бій проти боса гри. Гравець досліджує підземелля в пошуках скарбів та артефактів, розвиває свої навички, після програшу весь прогрес не втрачається, зібраний рівень зберігається а артефакти зникають, таким чином гравець просувається все далі по ігровому процесу.

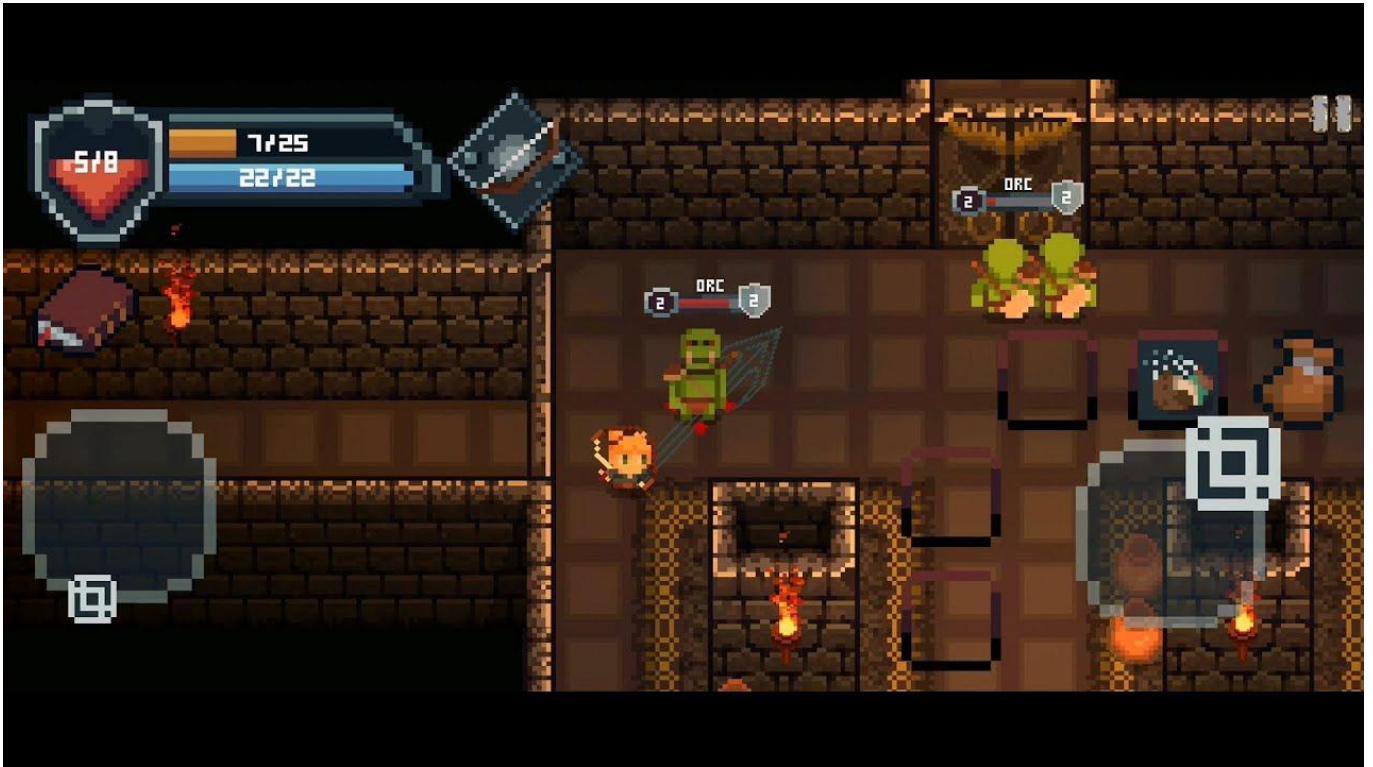


Рис. 1.8 – Ігровий процес Rogues` Tales: Action RPG

Жанр roguelike є доволі популярним серед гравців на мобільних платформах, адже дає змогу грати в туж саму гру по новому. Основні механіки можна реалізувати на кожній платформі, залежно від платформи, спосіб їх реалізації буде відрізнятися.

Провівши аналіз популярних ігор жанру roguelike було підібрано основні механіки бою, пересування та дослідження локацій. Обрано вид в камери збоку у 2D та стиль оформлення рівнів.

Висновок до розділу 1

В даному розділі було виконано вибір ігрового рушія, середовища для розробки, текстури, моделі, звуки для гри, також описано предметне середовище та досліджено платформи для розробки, проаналізовано наявні аналоги ігрових застосунків. Отже, Unity обрано ігровим рушієм, C# мовою програмування всіх скриптів, Visual studio 2022 середовище розробки, текстури для створення гри вирішено було взяти на сайті itch.io, «Dungeon» для створення рівнів, автор Pixel_Poem, «Pixel Art Fire Ball» для створення ефектів, автор Matheus Frantz, музика для фону «Tormented Souls» автора Муuu.

РОЗДІЛ 2

ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз предметної області

Roguelike – жанр комп'ютерних ігор головними критеріями якого є випадкова генерація рівнів, безповоротна смерть персонажа та необхідність в дослідженні локацій. Гра належить до жанру roguelike якщо налічує хоча б один з трьох головних критеріїв. Першою грою цього жанру рахується Rogue створена в 1980 році на університетських Unix системах.

В основі рольових ігор жанру roguelike лежать наступні правила, кожна нова ігрова сесія унікальна, гравець повинен підвищувати рівень аби пройти локацію або вигадати стратегію яка дозволить йому впоратись з противником. Цікавість до ігор такого жанру виникає через свободу у виборі дій, інтерес до вивчення нового, сам гравець вирішує як йому пройти ту чи іншу перешкоду і розв'язання його проблеми є безліч. Все що гравець зберігає після смерті його персонажа це власний досвід.

Ігровий додаток повинен містити наступні механіки:

- пересування персонажа по локації;
- можливість взаємодіяти з об'єктами на локації, скринями, ящиками, дверима;
- шкала здоров'я;
- система рівнів яка дозволяє підсилити персонажа;
- механіка покрокового бою;

2.2 Постановка задачі

В результаті аналізу предметної області сформовано мету кваліфікаційної роботи: розробка мобільної гри у 2D, жанрі roguelike «The Dungeon of Oblivion».

Перелік основних завдань для досягнення мети кваліфікаційної роботи:

- провести аналіз предметної області та огляд аналогів;
- визначити та знайти засоби які необхідні для розробки;
- спроектувати функціонал додатка;
- реалізувати основні механіки;
- провести тестування ігрового додатку.

Ігровий додаток повинен надавати новий досвід в кожній ігровій сесії, розважати гравця.

2.3 Розробка архітектури ігрового додатку

Створення гри завжди було складним, місцями клопітливим процесом. В процесі її створення необхідно об'єднати підготувати текстури, музику, написати скрипти та об'єднати це все в одну програму метою якої є розважити гравця.

Для коректної роботи ігрового застосунку необхідно налагодити всі модулі меню, інтерфейсу з якими може взаємодіяти гравець, та модулі які відповідають за переходи між сценами, взаємодію з об'єктами, ворогами.

Для деталізації всіх вимог слід врахувати всі процеси які можуть відбуватися в предметній області. Для цього їх зображають у вигляді прецедентів на UML діаграмі (рис. 2.1). Прецедент – це сценарій, один з можливих варіантів, який настає при взаємодії гравця з грою. Такий спосіб дозволяє побачити функціонування ігрового застосунку з точки зору гравця.

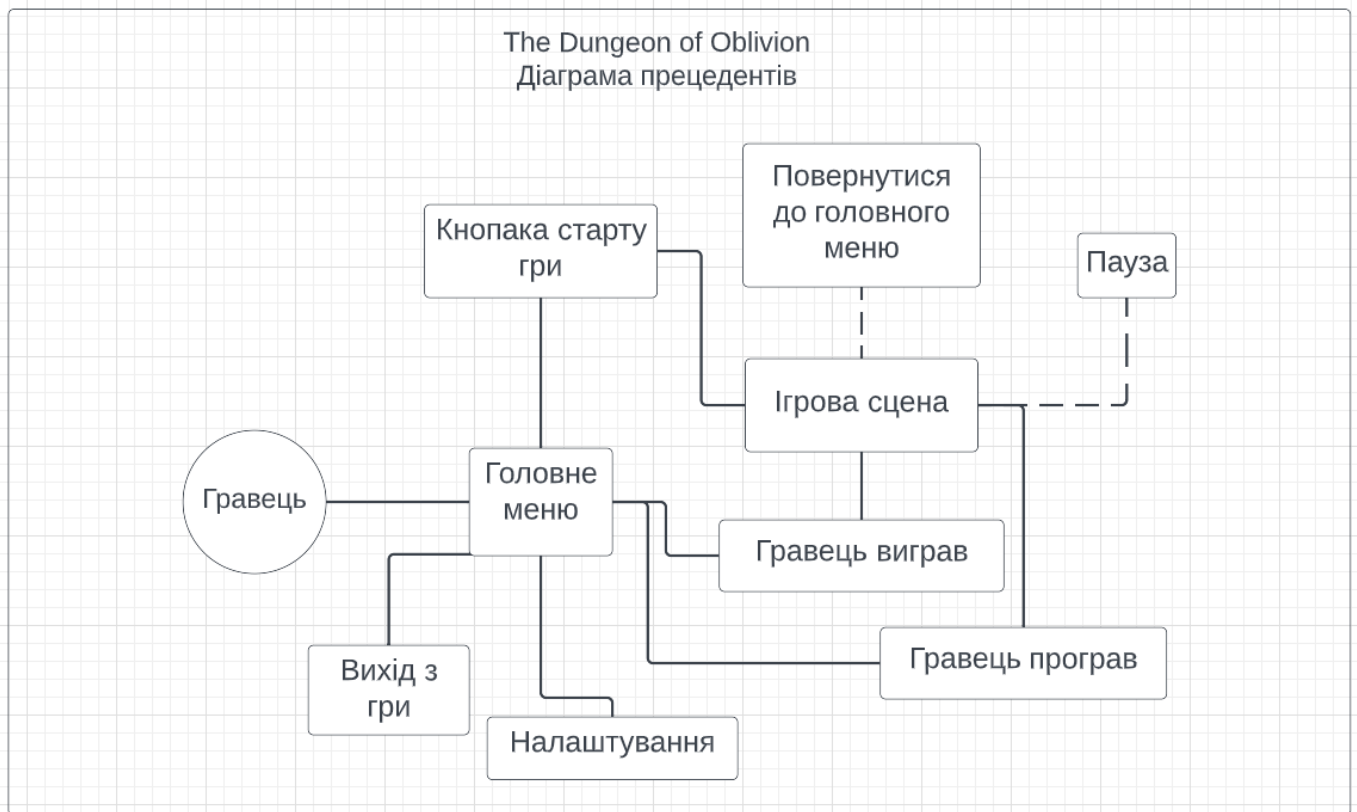


Рис. 2.1 – Діаграма прецедентів

2.4 Проектування інтерфейсу гри

Провівши аналіз вимог, було прийняте рішення щодо макета інтерфейсу ігрового додатка. Кожна сцена гри має свій інтерфейс, який є окремим станом та має свій скрипт для його реалізації.

Головний ігровий інтерфейс містить, шкалу здоров'я, шкалу рівня, кнопку виклику меню та паузи, лічильник зібраних монет, два місця для еліксирів здоров'я.

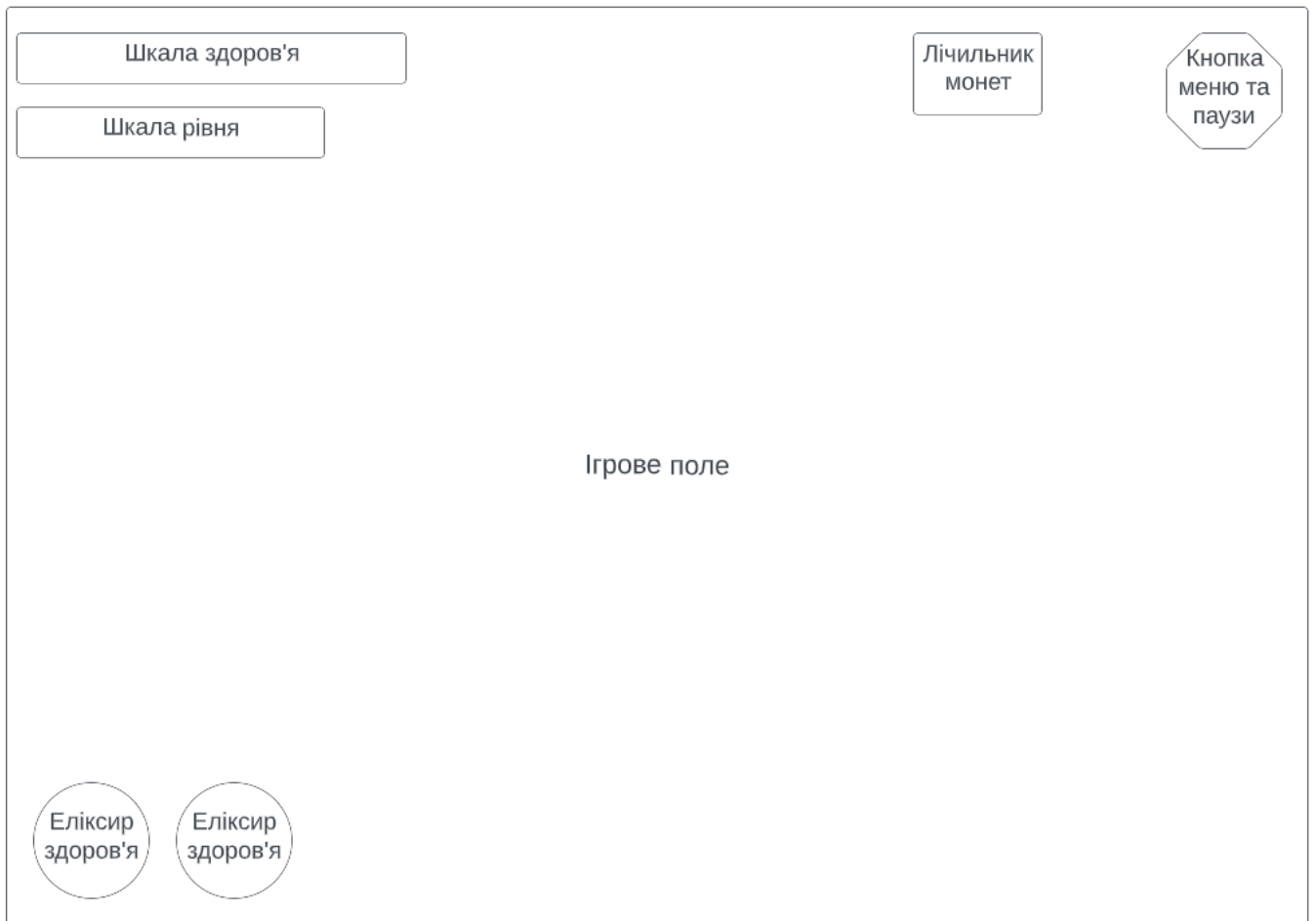


Рис. 2.2 – Прототип інтерфейсу головної ігрової сцени

2.5. Система меню

Перше що зустрічає гравця при запуску ігрового застосунку це головне меню. Завдяки меню можна відкрити налаштування, запустити гру або вийти з неї. Діаграма станів допомагає спроектувати головне меню.

Головне меню складається з:

- Кнопка «Start»
- Кнопка «Settings»
 - Кнопка «Music»
 - Регулятор гучності «Loudness»
- Кнопка «Exit»

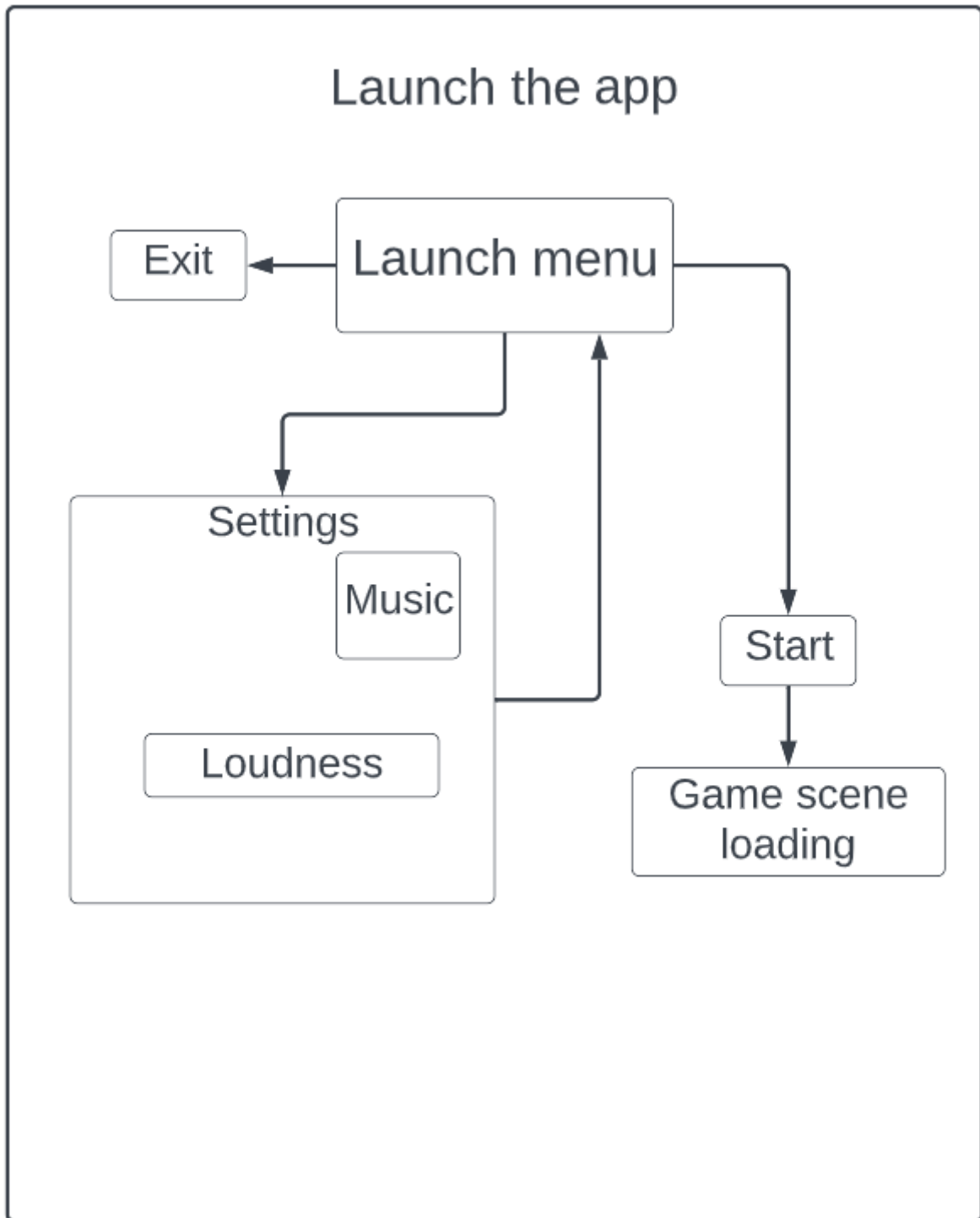


Рис. 2.3 – Діаграма станів – Запуск додатку

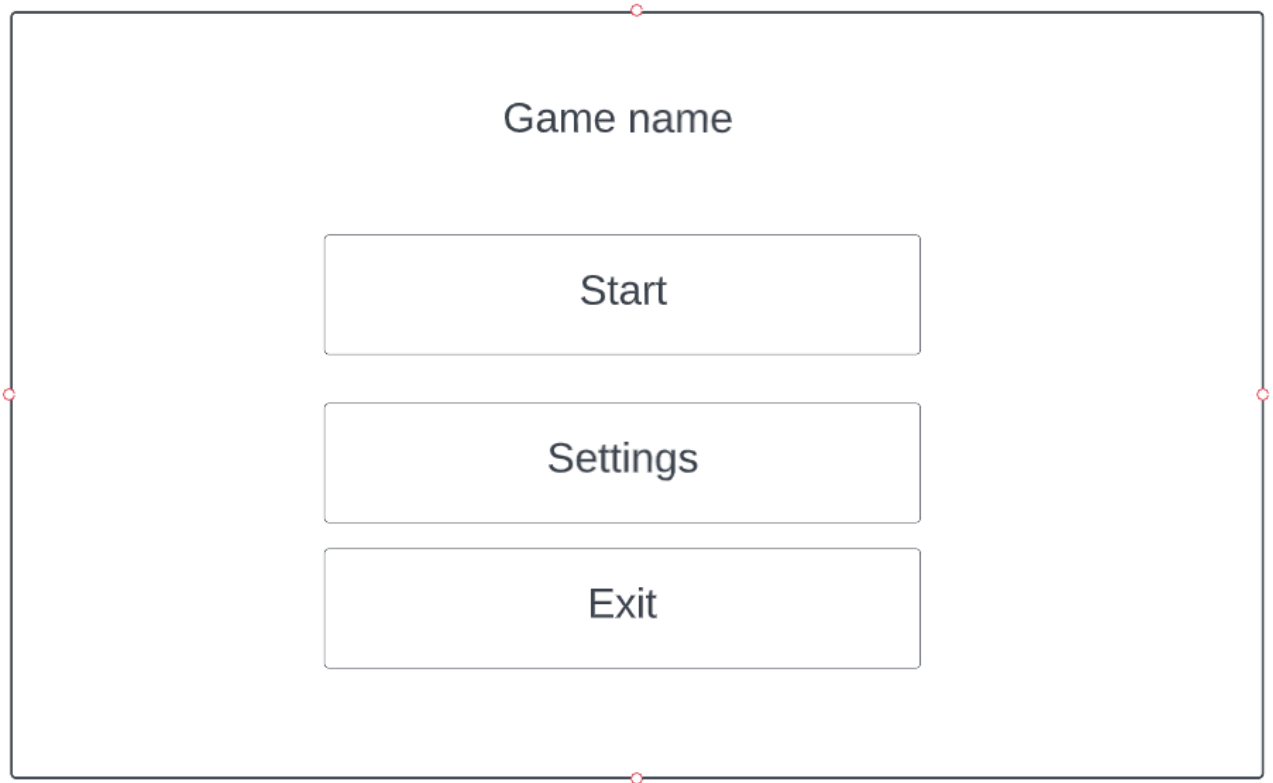


Рис. 2.4 – Макет головного меню

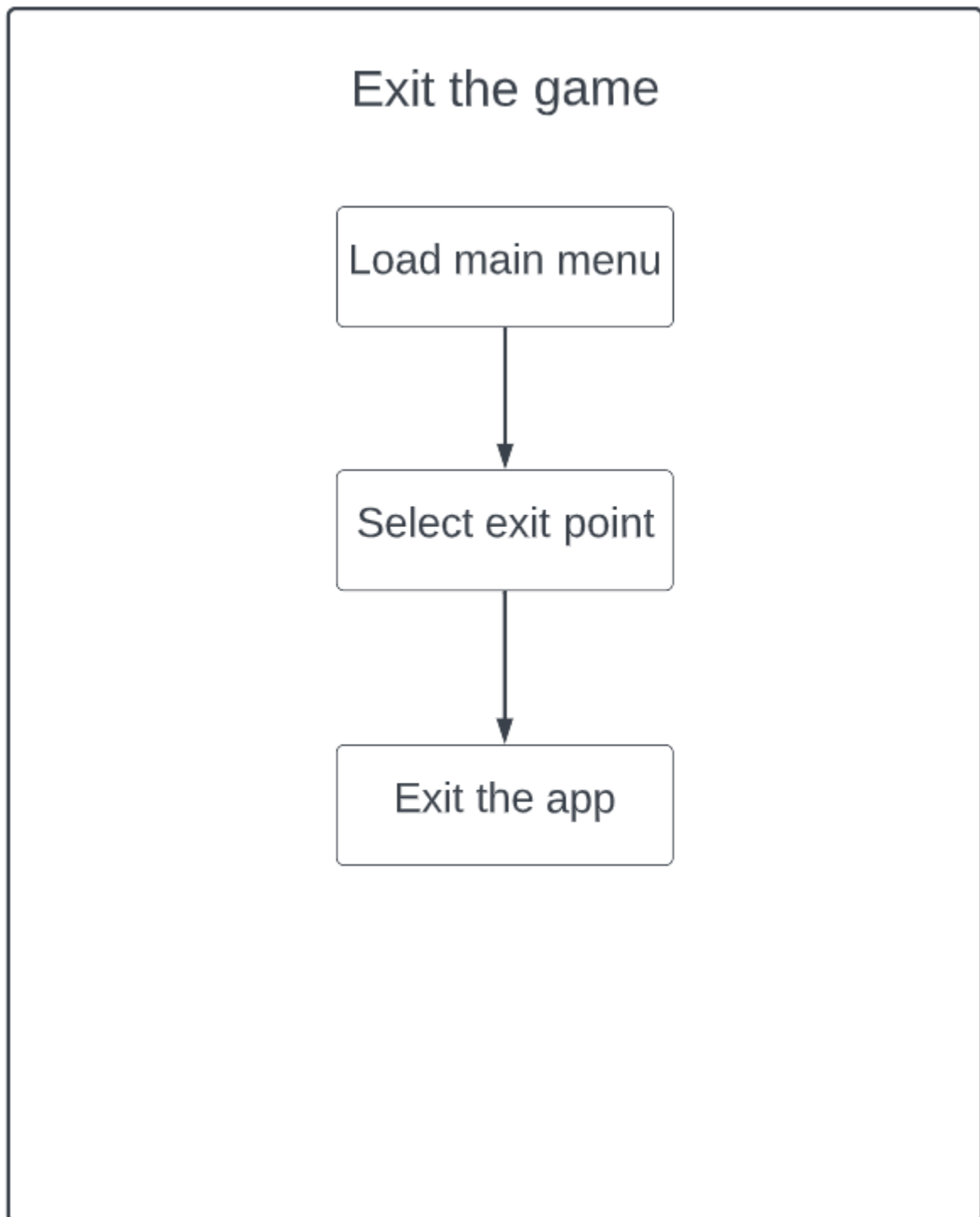


Рис. 2.5 – Діаграма станів – Вихід з гри

В ролі сцени для паузи буде викликатись головне меню з можливістю продовжити гру.

Висновок до розділу 2

В даному розділі було проаналізовано предметну область, виконано розробку архітектури, підготовлено постановку задачі ігрового додатка яка включає проектування макетів для головного меню, головного інтерфейсу гри

РОЗДІЛ 3

ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Засоби розробки

Основними засобами розробки є ігровий рушій Unity, мова програмування C# та середовище написання коду Visual Studio 2022, до допоміжних засобів відносяться документації, статті в мереж Інтернет, обговорення на форумах та відео уроки на хостингу YouTube.

3.2. Вимоги до технічного та програмного забезпечення

Ігровий застосунок розроблявся на комп'ютері:

Acer Nitro 5

- Процесор: Intel Core i5-10300H
- Відеокарта: NVIDIA GeForce RTX 2060 4 ГБ
- Оперативна пам'ять: 16 ГБ
- Операційна система: Windows 10 64-розрядна

Тестувалась на мобільному телефоні:

POCO M3 Pro 5G

- Процесор: MediaTek Dimensity 700 8 x2 - 2.2 GHz
- Оперативна пам'ять: 4 ГБ
- Операційна система: MIUI 12.5.5 (Android 11)

3.3 Опис програмної реалізації

3.3.1. Створення меню

Для того, щоб створити меню в Unity, не знадобиться багато часу, для цього потрібно створити окрему сцену під назвою Menu.

Сцена Menu містить в собі, головне меню яке в свою чергу дає можливість запустити гру, відкрити меню налаштувань, вийти з гри. Меню налаштувань дає змогу увімкнути або вимкнути всі звуки у грі, регулювати їх гучність.



Рис. 3.1 – Головне меню

Скрипт Menu реагує на натискання кнопок «Start», «Settings», «Exit». При натисканні на «Start», відбувається перехід на ігрову сцену, на «Exit» – закривання ігрового застосунку, на «Settings» – відкриється меню налаштувань, в якому можна змінити рівень гучності та вимкнути фонову музику, для повернення слід натиснути на стрілку яка знаходиться у верхньому лівому куті.

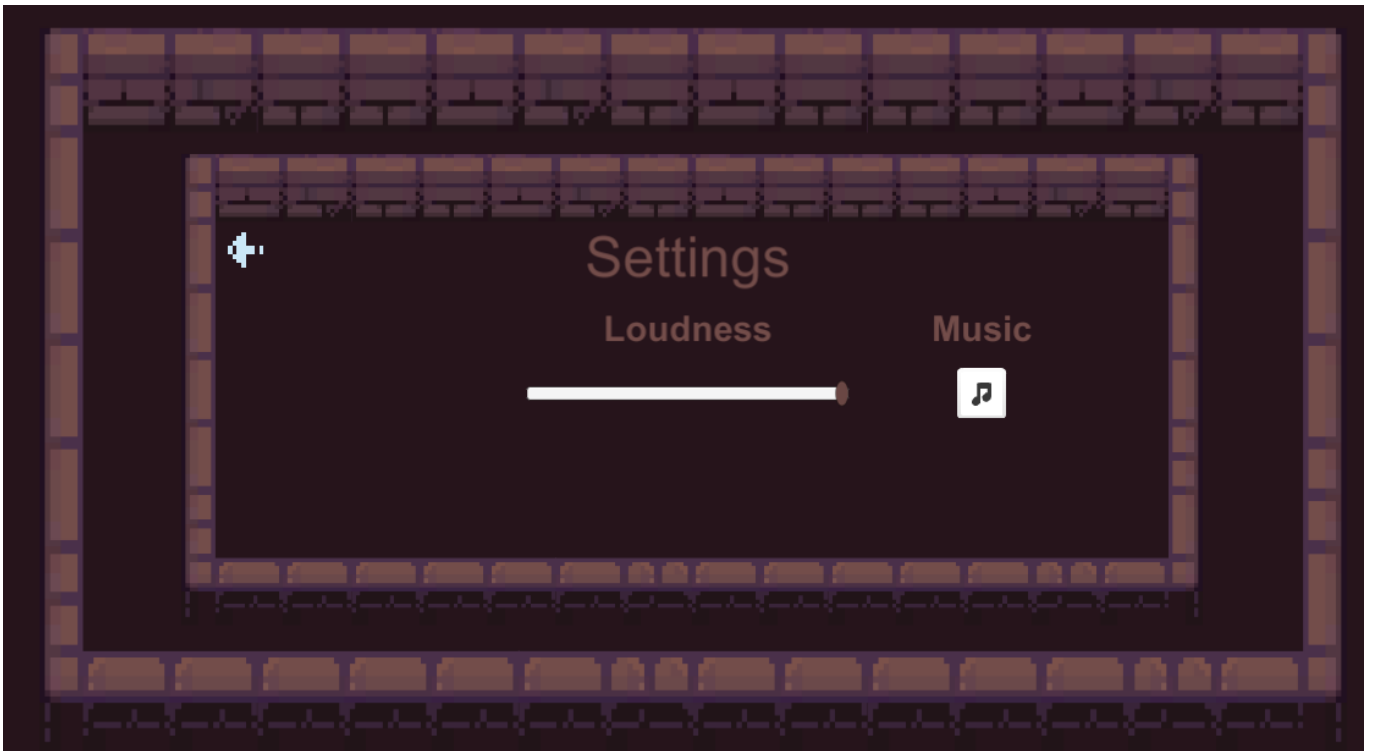


Рис. 3.2 – Меню налаштувань

3.3.2. Створення кімнат ігрового рівня

Для створення рівня використовують сітку, на ній розміщують текстури об'єктів. Підхід з використанням Tile Palette сильно допомагає у створенні рівня тому, що економить багато часу, його суть полягає в формуванні палітри з текстур які потім можна легко розмістити на сітці рівня.

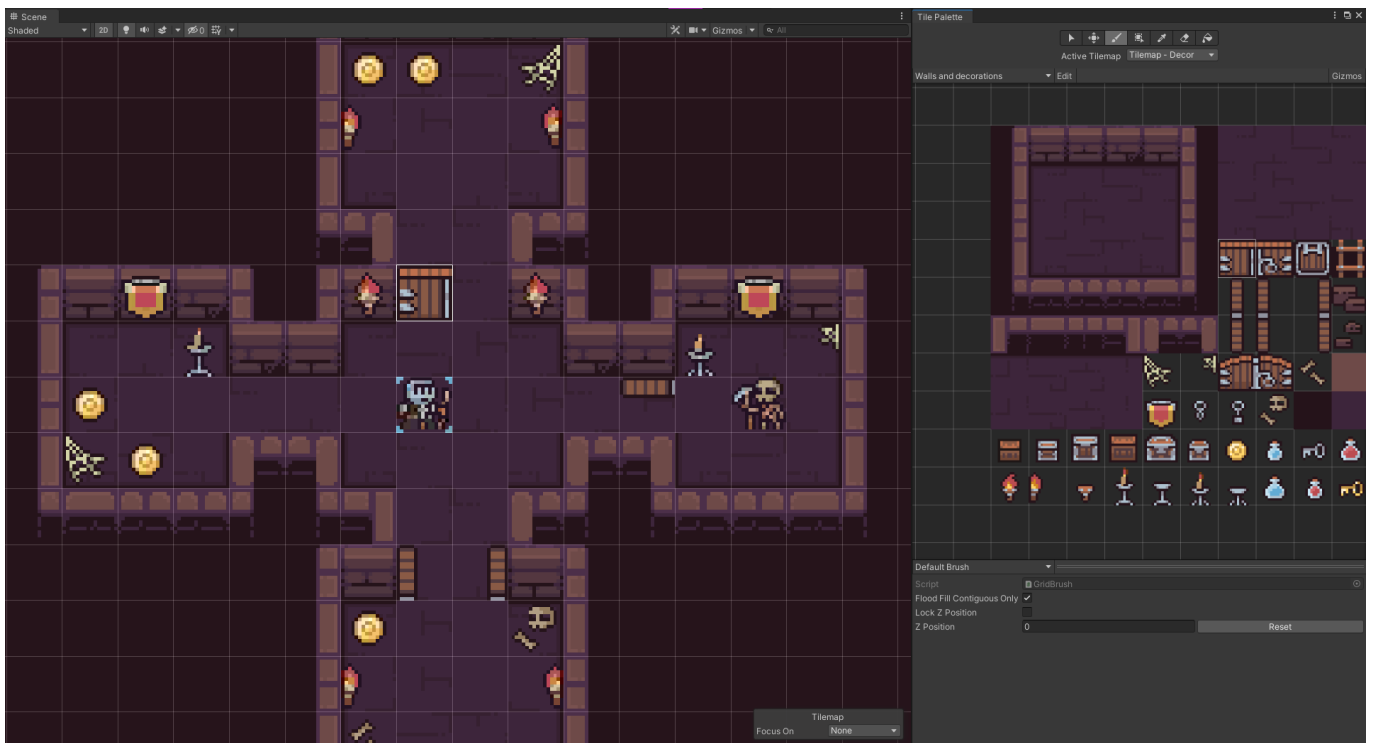


Рис.3.3 – Процес створення ігрового рівня

3.3.3. Інтерфейс

На рисунку 3.4 зображено головний інтерфейс, він відображає шкалу здоров'я, шкалу рівня, рахунок зібраних монет, також дає можливість застосувати зілля, зупинити гру натиснувши на значок паузи. Інтерфейс був реалізований згідно макету.



Рис.3.4 – Головний інтерфейс

У інтерфейсі ігрової сцени присутні наступні елементи:

- HealthBar – відображає рівень здоров'я гравця;
- XpBar – відображає набраний рівень гравця;
- Potion – відображає у гравця наявність зілля для лікування;
- ScoreText - відображення кількість зібраних монет гравцем під час гри;
- PauseButton - відповідальний за виклик меню паузи.

3.3.4. Розробка Player та TouchAndGo

Для створення персонажа було обрано спрайт та створено покадрову анімацію.



Рис.3. 5 – Спрайт персонажа гравця

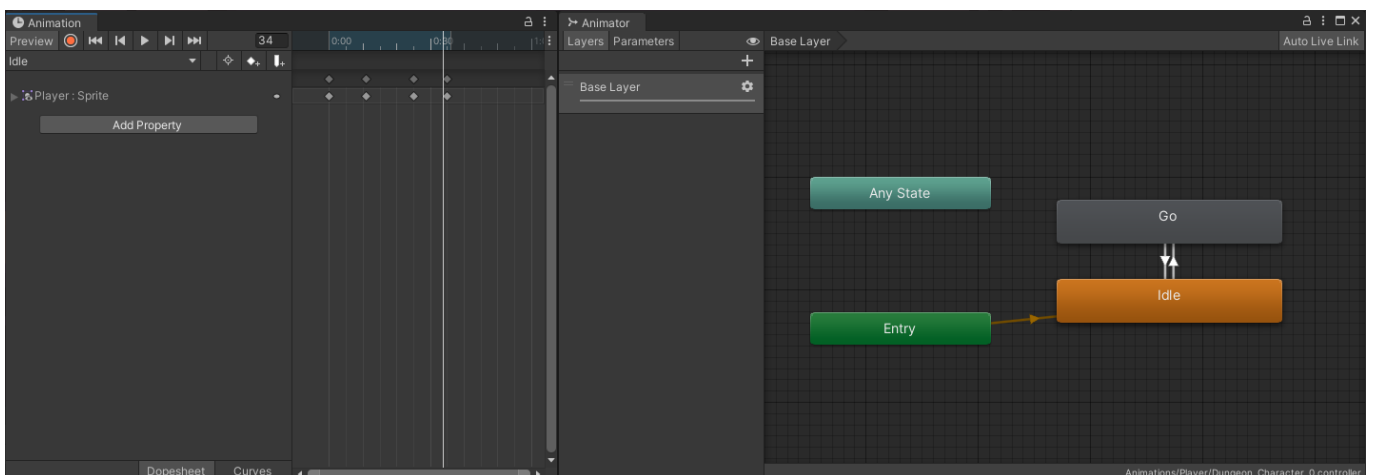


Рис.3. 6 – Створення покадрової анімації

Для того, щоб персонаж не міг проходити через стіни рівня, йому було присвоєно компоненти Box Collider 2D та Rigidbody 2D які наділяють фізичними властивостями. Оскільки гра у 2D, переміщення здійснюється лише в площинах X та Y, слід заблокувати площину Z встановивши галочку в комірці поля Freeze Rotation.

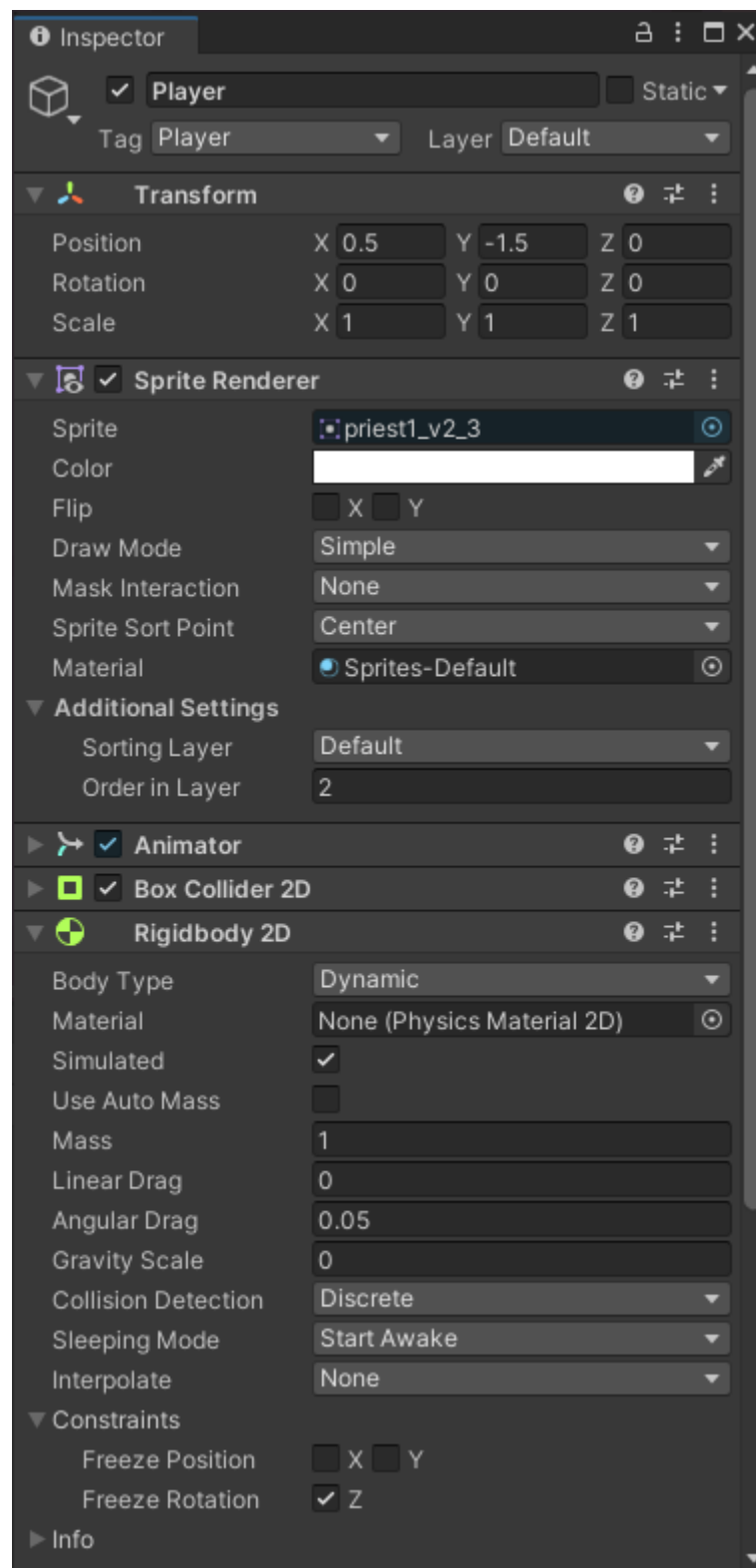


Рис.3. 7 – Компоненти які належать Player та їх налаштування

TouchAndGo скрипт який відповідає за переміщення персонажа по до місця дотику на екрані.

```

1  using ...
4
5  public class TouchAndGo : MonoBehaviour {
6
7      [SerializeField]
8      float moveSpeed = 5f;
9
10     Rigidbody2D rb;
11
12     Touch touch;
13     Vector3 touchPosition, whereToMove;
14     bool isMoving = false;
15
16     float previousDistanceToTouchPos, currentDistanceToTouchPos;
17
18     void Start () {
19         rb = GetComponent<Rigidbody2D> ();
20     }
21
22     void Update () {
23
24         if (isMoving)
25             currentDistanceToTouchPos = (touchPosition - transform.position).magnitude;
26
27         if (Input.touchCount > 0) {
28             touch = Input.GetTouch (0);
29
30             if (touch.phase == TouchPhase.Began) {
31                 previousDistanceToTouchPos = 0;
32                 currentDistanceToTouchPos = 0;
33                 isMoving = true;
34                 touchPosition = Camera.main.ScreenToWorldPoint (touch.position);
35                 touchPosition.z = 0;
36                 whereToMove = (touchPosition - transform.position).normalized;
37                 rb.velocity = new Vector2 (whereToMove.x * moveSpeed, whereToMove.y * moveSpeed);
38             }
39
40             if (currentDistanceToTouchPos > previousDistanceToTouchPos) {
41                 isMoving = false;
42                 rb.velocity = Vector2.zero;
43             }
44
45             if (isMoving)
46                 previousDistanceToTouchPos = (touchPosition - transform.position).magnitude;
47         }
48     }

```

Рис.3. 8 – Вміст TouchAndGo

3.3.5. Камера та звуки

Для того, щоб камера завжди слідувала за гравцем її було переміщено в середину об'єкта Player, таким чином вона наслідує від гравця положення.

Для кращого занурення в гру було додано музику заднього фону. Реалізовано це наступним чином, об'єкту Player присвоєно компонент Audio Source в який свою чергу передано саму музику.

3.3.6. Розробка Енему

Енему – супротивники які зображені як люди, скелети, привиди, вампіри, що б пройти рівень необхідно перемогти їх.

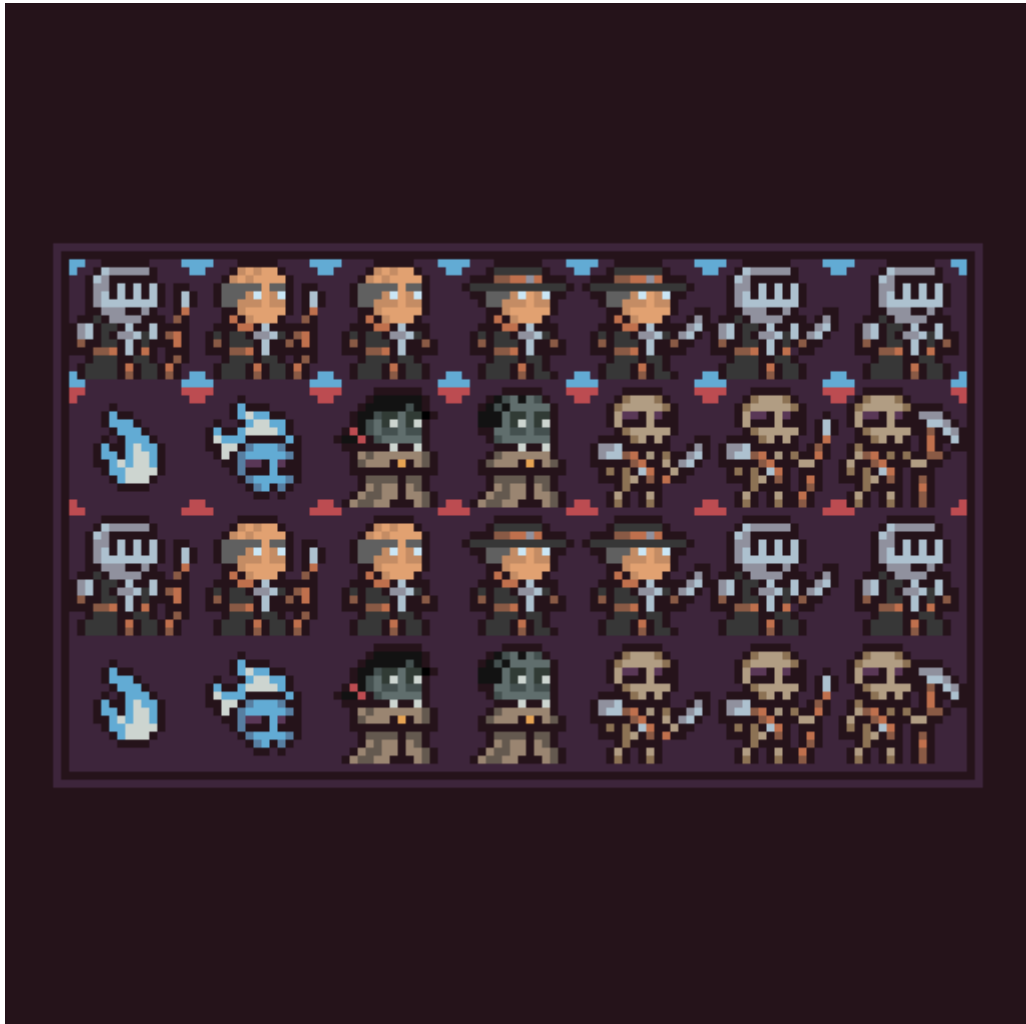


Рис.3.9 – Спрайти всіх енему

Кожен ворог має свою анімацію, відрізняються вони силою, рівнем та кількістю здоров'я. На кожного з них можна напасти перебуваючи на ігровому рівні.

3.3.7. Бойова система

Бойову система за стилем схожа на систему бою японської рольової гри, її суть полягає в покроковому прийнятті рішень та дій, спочатку гравець атакує або ховається потім в свою чергу це робить ворог.



Рис.3. 10 – Демонстрація бойової системи

Інтерфейс такої системи складається з полів які інформують гравця про рівень здоров'я персонажа, його рівня, також і надає інформацію щодо параметрів супротивника, його здоров'я, рівня, імені.



Рис.3. 11 – Інтерфейс бойової системи

У інтерфейсі сцени бою присутні наступні елементи:

- PlayerBattleHud – вказує на місце де під час бою буде розташовано модель ігрового персонажа;
- PlayerLvl – відображає рівень гравця;
- HPPlayerPanel – відображає рівень здоров'я гравця;
- PlayerName – відображає ім'я ігрового персонажа;
- EnemyBattleHud – вказує на місце де під час бою буде розташовано модель противника;
- EnemyLvl – відображає рівень противника;
- EnemyPanel – відображає рівень здоров'я противника;

- EnemyName – відображає ім'я противника;
- DialoguePanel – повідомляє ігрову інформацію;
- AttackButton – кнопка яка відповідає за активацію нападу;
- HealButton – кнопка яка відповідає за активацію лікування.

3.3.8. Робота над логікою гри

Логіку гри прописано в скриптах, кожен скрипт який виконує ігрову логіку має два внутрішні методи, метод start та update. Код написаний в методі start виконується лише раз при першому сцени гри, зазвичай в цей метод прописують логіку генерування ворогів, рівнів чи предметів. На відміну від методу start, метод update викликає виконання коду кожен ігровий кадр, що робить його основним для реалізації логіки гри.

В розробці даного ігрового застосунку для присвоєння фізичних властивостей та можливості взаємодії, об'єктом було надано компоненти Box Collider 2D та Rigidbody 2D. Box Collider 2D робить текстуру об'єкта твердою що дозволяє йому стикатись з іншими об'єктами. Rigidbody 2D наділяє об'єкт масою, ще дозволяє змінювати вплив сили тяжіння на об'єкт.

3.3.9. Ліцензії та авторські права на асет текстур, музику

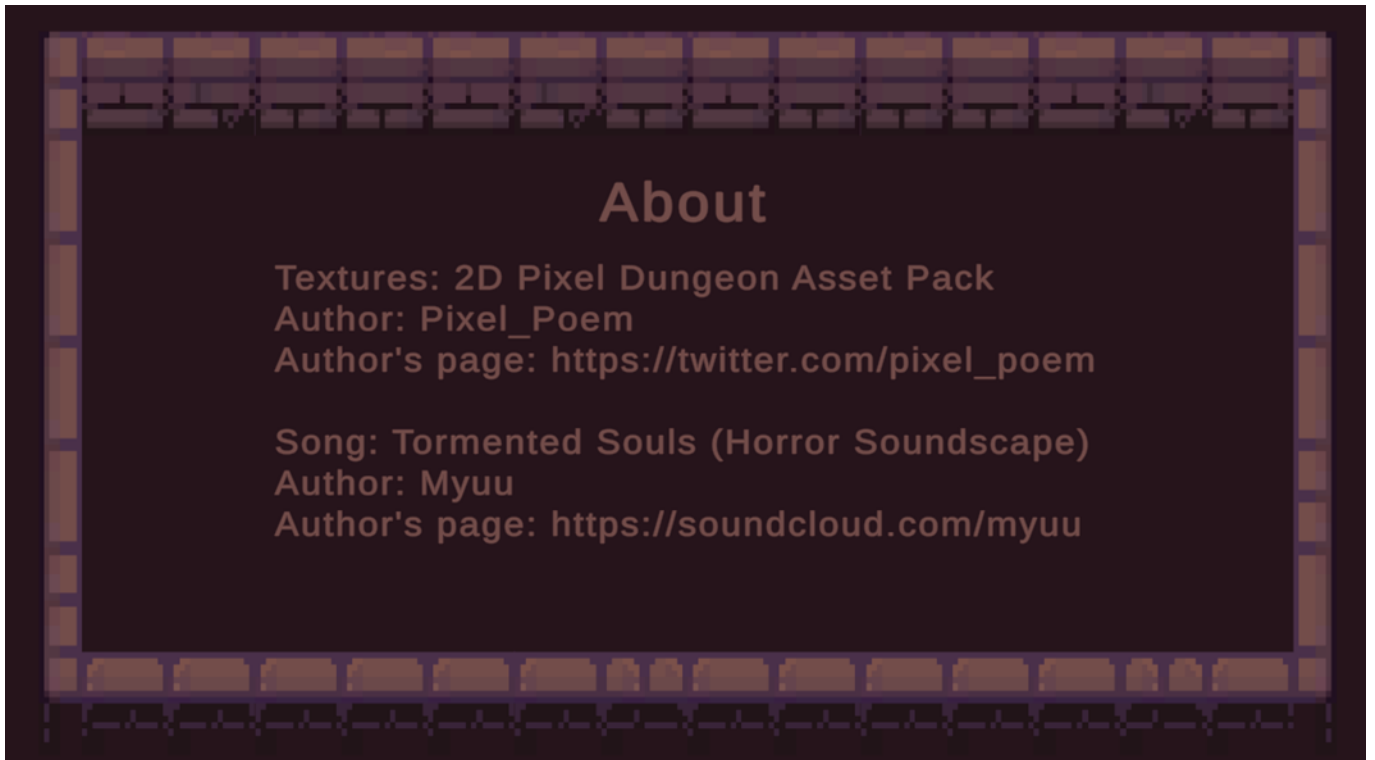


Рис.3. 12 – Вкладка «About»

Згідно правил ліцензії на якій поширюється асети текстур, та музики, кожен хто бажає використати їх у своїх роботах, повинен будь де окремо в проекті зазначити їх назви, псевдонім автора та залишити посилання на сторінку автора.

В кваліфікаційній роботі я використав асети текстур під назвою «2D Pixel Dungeon» автора Pixel_Poem та музику «Tormented Souls (Horror Soundscape)» автора Myuu. Даний асет поширюється за безкоштовною ліцензією, а право на використання музики у кваліфікаційній роботі було запитано напряму у автора, він дав дозвіл на використання, при умові згадки про нього у ігровому додатку.

Висновок до розділу 3

В ході роботи над кваліфікаційним проектом затверджено засоби для розробки, вимоги до програмного та технічного забезпечення також було описано процеси реалізації ігрового застосунку засобами розробки.

Розділ 4

Тестування

4.1 Функціональне тестування

Функціональне тестування – вид тестування який перевіряє відповідність функціональних вимог програмного забезпечення його дійсним характеристикам. Результати функціональне тестування наведено таблиці 4.1

Таблиця 4.1

№	Назва тесту	Очікуваний результат	Фактичний результат	Відмітка про проходження тестування
1	Відображення меню гри	На екрані відображається головне меню з усіма кнопками	На екрані відображено головне меню та кнопки «Start», «Settings», «Exit»	+
2	Робота кнопки «Start»	При натиску на кнопку «Start» відбувається перехід на ігрову сцену	При натиску на кнопку «Start» відбувся перехід на ігрову сцену	+
3	Робота кнопки «Settings»	При натиску на кнопку «Settings» відкривається меню налаштування, зміна позиції повзунка впливає на гучність.	При натиску на кнопку «Settings» було відкрито меню налаштування, зміна позиції повзунка, змінювала рівень гучності	+
4	Робота кнопки «Exit»	При натиску на кнопку «Exit», гра закриється	При натиску на кнопку «Exit» гра закрилась	+
5	Виклик паузи	При натиску на кнопку зі знаком паузи відкривається головне меню	При натиску на кнопку зі знаком паузи відкрилось головне меню	+
6	Перевірка переміщення	Після дотику до дочки на екрані персонаж розпочинає рухатись в задану точку	Після дотику персонаж почав рух в напрямку точки дотику	+

7	Перевірка збору монет	Коли персонаж взаємодіє з монетою, вона зникає а лічильник збільшує число	Після взаємодії монета зникла, лічильник збільшив число	+
8	Перевірка роботи бойової системи	При нападі на противника відкривається інтерфейс бойової системи, бій проходить покровою, шкала здоров'я зменшується при зазнанні шкоди	Після нападу відкрився інтерфейс бойової сцени, бій проходив покровою, шкала здоров'я зменшувалась	+

4.2. Usability Test

Usability Test – тест за результатами якого стає зрозуміло чи продукт є зручним у використанні. Під час тестування учасники намагаються виконати звичайні, типові завдання.

Учасникам тестування необхідно виконати такі задачі:

- почати гру;
- зупинити гру;
- продовжити гру;
- підняти монети;
- вступити в бій з противником;
- закінчити гру.

Висновок до розділу 4

Задачі поставлені перед тестувальником такі як функціональне тестування та Usability test було виконано успішно, під час тестування не виникало жодних проблем, також було зазначено зручність оформлення.

ВИСНОВКИ

В ході написання кваліфікаційної роботи було проаналізовано сучасні технології призначені для розробки ігрових додатків, та їх вплив на ігрову індустрію. Було показано можливості ігрового рушія Unity у розробці гри в стилі 2D, жанрі roguelike.

Unity кросплатформений універсальний ігровий рушій, який використовується у створенні ігор різноманітних стилів, жанрів та платформ. Він є адаптованим під більшість операційних систем та надає всі необхідні інструменти для створення будь якої гри, не залежно від стилю, жанру чи ідеї, є чудовим інструментом як для початківців, які тільки знайомляться із сферою створення відеоігор так і для професіоналів які мають великий досвід у розробці.

Мовою програмування було обрано C#, причиною для її вибору стала її зручність та не високий поріг входження, вона краще підходить для малих та середніх за розмірами проєктів. C# є найкращим вибором для початківця у сфері створення відеоігор тому, що має хорошу документацію, в інтернеті є багато відео уроки з її застосування в розробці ігрових застосунків. Вона була створена з ідеєю об'єднати найкращий функціонал C++, Objective-C, Java, Delphi.

Асети з 2D графікою, музику було взято на відкритих ресурсах, згідно з правами ліцензій та з дозволу авторів контенту.

Таким чином, використовуючи ігровий рушій Unity та асети графіки, музику, можна створити відео гру в жанрі roguelike для платформи Android.

Для досягнення цілі, а саме створення ігрового застосунку в жанрі roguelike під мобільну платформу Android на рушію Unity, було виконано такі завдання:

- Проведено аналіз актуальності теми;
- Виконано аналіз існуючих аналогів;
- Виконано огляд актуальних програмних середовищ для реалізації додатка;
- Спроектовано ігровий додаток;

- Реалізовано ігровий додаток;
- Проведено тестування ігрового застосунку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація по Android [Електронний ресурс]: — Режим доступу: <https://developer.android.com/docs>
2. Документація по Godot [Електронний ресурс]: — Режим доступу: <https://docs.godotengine.org/en/stable/>
3. Загальні відомості по CryEngine [Електронний ресурс] – Режим доступу: <https://godotengine.org/>
4. Операційні Системи [Електронний ресурс]: Операційні системи. – Режим доступу: https://wiki.cuspu.edu.ua/index.php/Операційні_системи.
5. Android operation information [Електронний ресурс]: Android O S. – Режим доступу: <https://www.gadgetgyani.com/wp-content/uploads/2016/03/android-features-pdf.pdf>
6. Загальні відомості по CryEngine [Електронний ресурс] – Режим доступу: <https://www.cryengine.com/>
7. Документація по CryEngine [Електронний ресурс] – Режим доступу: <https://docs.cryengine.com/>
8. Документація по C# [Електронний ресурс]: — Режим доступу: <https://docs.microsoft.com/en-us/dotnet/csharp/>
9. Документація по Visual studio [Електронний ресурс]: — Режим доступу: <https://docs.microsoft.com/uk-ua/visualstudio/windows/?view=vs-2017>
10. Ігровий рушій [Електронний ресурс]: — Режим доступу: https://uk.wikipedia.org/wiki/Ігровий_рушій
11. Unity store [Електронний ресурс]: – https://store.unity.com/ru?_ga=2.200322542.1765686190.1560285797-130111505.1547668569
12. Відео уроки по Unity [Електронний ресурс]: – <https://www.youtube.com/c/Brackeys/videos>
13. Асети текстур для побудови рівнів [Електронний ресурс]: – <https://pixel-poem.itch.io/dungeon-assetpuck>
14. Музика для заднього плану Tormented Souls, автор Myuu [Електронний ресурс]: – <https://soundcloud.com/myuu/tormented-souls-horror-soundscape>