Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем

Розробка 2D RPG гри для Android

Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення»

Керівник курсової роботи
ст. викладач Борозенний С. О.

_____

(підпис)
“____” _____ 2023 р.
Виконав студент 3 р. н.
Дубовик А. В.
“____” _____ 2023 р.

Київ – 2023

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем

ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем

доцент, к.ф.-м.н.

О. П. Жежерун

_____2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Дубовику Андрію Вікторовичу 3-го курсу факультету інформатики

ТЕМА Розробка 2D RPG гри для Android

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Календарний план виконання роботи

Анотація

Вступ

1 Огляд ігрових рушіїв

2 Аналіз задачі

3 Реалізація гри

Висновки

Список використаних джерел

Дата видачі „___” _____ 2023 р. Борозенний О.С. _____

(підпис)

Завдання отримав _____

(підпис)

Тема: Розробка 2D RPG гри для Android

Календарний план виконання роботи:

№ п/п	Назва етапу	Термін виконання	Примітка
1.	Отримання завдання на курсову роботу	01.11.2022	
2.	Огляд технічної літератури за темою роботи	05.12.2022	
3.	Ознайомлення з ігровими рушіями	20.12.2022	
4.	Детальне ознайомлення з рушієм Unity	10.01.2023	
5.	Проєктування гри	01.02.2023	
6.	Розробка гри	15.02.2023	
7.	Написання текстової частини роботи	01.04.2023	
8.	Створення слайдів для доповіді та написання доповіді	11.05.2023	
9.	Захист курсової роботи	24.05.2023	

Студент Дубовик А. В.

Керівник Борозенний О. С.

“_____”_____

Зміст

Анотація

Курсова робота присвячена розробці 2D гри в жанрі RPG для операційної системи Android. Детально було розглянуто основі принципи розробки з використанням Unity та продемонстровано процес створення гри на даному рушії. Результатом роботи є готова гра, яка може бути використана як основа для подальшого розширення та вдосконалення.

Ключові слова: ігрова розробка, RPG гра, Android гра, ігровий рушій Unity.

Вступ

Відеоігри є одними з найбільш популярних розваг у сучасному світі та займають важливе місце в культурі, а ігри на мобільних платформах з кожним роком стають все більш популярними. Розробка гри є доволі складним і творчим процесом, що вимагає від розробників вмінь і знань з різних областей.

У цій курсовій роботі розглянуто процес розробки 2D RPG гри з використанням рушія Unity. Описано процес створення основних компонентів гри: головного героя, ворогів, ігрового світу, інших об’єктів. Показано взаємодію всіх ігрових елементів.

Ця робота дозволяє отримати практичні знання та навички у створенні відеоігор, а також ознайомитись з процесом проєктування та розробки гри жанру RPG. Робота може бути корисною для тих, хто вивчає дану галузь, а також для тих то хоче отримати загальне уявлення про процес розробки гри.

Основна частина

Розділ 1 : Огляд ігрових рушіїв

1.1 Загальний огляд ігрових рушіїв

Ігровий рушій (англ. game engine) – це програмне забезпечення, що використовується для розробки відеоігор та інтерактивних застосунків. Він надає розробникам доступ до різноманітних функцій та інструментів, які можна використовувати для створення графіки, фізики, звуку, штучного інтелекту, мережевої гри та іншого.

Ігровий рушій зазвичай містить в собі готові бібліотеки, які надають функції для роботи з графікою, звуком, введенням даних та іншими різноманітними аспектами гри. Це дозволяє розробникам зосередитися на створенні геймплею, ігрової логіки та іншого контенту, замість того, щоб займатись розробкою власних інструментів

Великі компанії часто мають власні ігрові рушії, які розроблені для своїх потреб. Інші розробники часто використовують уже готові рішення. Серед найбільш популярних рішень - Unity та Unreal Engine. Кожен з цих рушіїв має свої особливості, переваги та недоліки, які варто враховувати при виборі платформи для розробки.

Unity - це ігровий рушій, який був випущений у 2005 році. Він здатний працювати з різноманітними платформами, включаючи Windows, macOS, Linux, Android та iOS. Основною мовою програмування, що використовується при розробці на Unity є C#. Однією з головних переваг Unity є простота використання та доступність для новачків у галузі розробки відеоігор. Unity також має велику спільноту розробників, яка надає підтримку та відповіді на запитання. Недоліком Unity є те, що він має меншу широку функціональність порівняно з Unreal Engine. Крім того, Unity не має настільки потужної графічної системи, що може призвести до обмежень у розробці графічно важких ігор.

Unreal Engine - це ігровий рушій, який був розроблений компанією Epic Games. Він випущений у 1998 році та зараз використовується для розробки великої кількості відеоігор на різних платформах, включаючи PC, консолі та мобільні пристрої. Для розробки в Unreal Engine використовується мова програмування C++. Однією з найбільших переваг Unreal Engine є його потужна графічна система, яка дозволяє розробникам створювати графічно привабливі ігри з високою деталізацією. Unreal Engine також має велику функціональність, яка дозволяє розробникам створювати складні системи

поведінки персонажів та фізики. Проте Unreal Engine є складнішим для вивчення та використання порівняно з Unity, тому для новачків у галузі розробки відеоігор може виявитись складнішим у використанні. Крім того, Unreal Engine має відносно високі вимоги до комп'ютера, що може зменшити продуктивність на старих або менш потужних системах.

Обидва ігрові рушії мають велику кількість ресурсів, документації та підтримки від спільноти розробників. Вибір між Unity та Unreal Engine залежить від потреб та вимог проєкту, а також від рівня досвіду розробника. Обидва ігрові рушії можуть бути використані для розробки різноманітних ігор та додатків з високими стандартами якості та професійним дизайном.

1.2 Огляд ігрового рушія Unity

Хоча Unreal Engine також є потужним рушієм, Unity має деякі переваги, які роблять його більш привабливим для розробників Android ігор:

- Доступність:
  Unity має дещо меншу криву вивчення порівняно з Unreal Engine. Unity є більш доступним для новачків у даній сфері, оскільки має меншу кількість складних функцій, що швидше приступи до самого процесу розробки.

- Низькі вимоги до апаратного забезпечення:
  Unity має менші вимоги до апаратного забезпечення, що дозволяє розробляти ігри для менш потужних пристроїв, що є особливо важливим у випадку розробки під Android.

Саме через ці переваги даного проєкту в якості ігрового рушія було обрано Unity.

Інтерфейс Unity має багато функцій, включаючи сцену, де можна розміщувати об'єкти, ієрархію, де можна керувати структурою об'єктів, вікно інспектора, де можна змінювати властивості об'єктів, та вікно проєкту, де можна керувати файлами проєкту. Інтерфейс дуже інтуїтивно зрозумілий та дозволяє розробникам швидко розпочати роботу з Unity.

Unity також має підтримку шейдерів, які дозволяють розробникам налаштовувати візуальні ефекти в грі, такі як світло, тіні, рельєф, прозорість та інші. Крім того, Unity має вбудовану фізичну систему, яка дозволяє розробникам створювати реалістичну фізику для об'єктів в грі, таких як колізії, сили та інерція.

Далі буде розглянуто основні поняття з якими зустрічається розробник, що працює з Unity.

1. Сцена - це основний простір, де відбувається ігровий процес. Вона містить об'єкти, які утворюють ігровий світ, такі як персонажі, об'єкти навколишнього середовища, світло, звук та інші елементи гри. В Unity можна створити декілька сцен, які можуть бути завантажені під час гри в залежності від потреби.

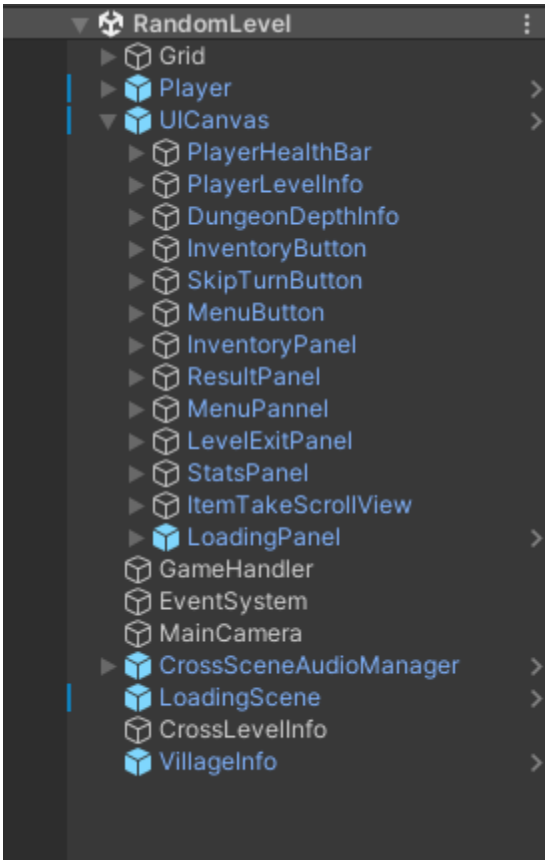

Рисунок 1. Структура сцени

2. Ігрові об'єкти - це основні елементи гри, які можуть бути розміщені на сцені. Ігрові об'єкти можуть бути різних типів, таких як персонажі, об'єкти навколишнього середовища, зброя, ефекти, інтерфейс та інші. Кожен ігровий об'єкт має певні властивості, такі як розмір, положення, орієнтація, текстури та інші.

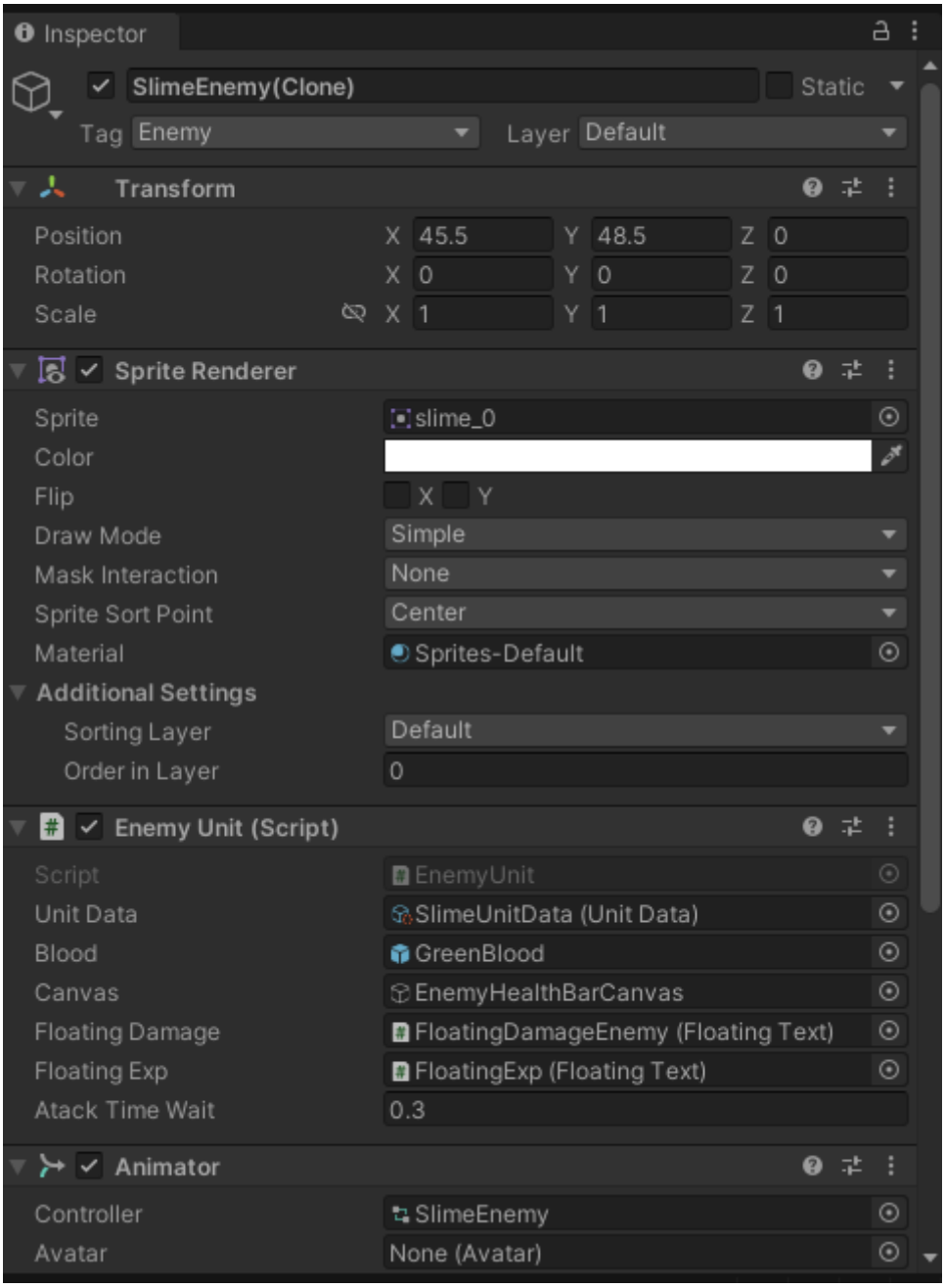

Рисунок 2. Редактор ігрового об'єкта

3. Скрипти - це програми, які можуть бути написані розробником для створення власної логіки та функціональності гри. Вони можуть взаємодіяти з іншими компонентами ігрових об'єктів, зчитувати та змінювати їх властивості, створювати нові об'єкти, видлаяти існуючі об’єкти та багато іншого. Для написання скриптів використовується мова програмування C#.
4. Префаби – “макети” ігрових об’єктів. Префаб дає можливість створювати багато однакових об'єктів за допомогою одного шаблону, що зменшує час, потрібний для створення багатьох

однакових об'єктів та значно спрощує роботу з великою кількістю об'єктів в проєкті.

Для роботи з сценою та ігровими об'єктами в Unity використовуються різні інструменти, такі як редактор сцени, редактор ігрових об'єктів, компоненти, скрипти тощо. Редактор сцени дозволяє розробнику переглядати, розміщувати об'єкти на сцені, переміщати їх, масштабувати, повертати та змінювати інші параметри. Редактор ігрових об'єктів дозволяє налаштовувати властивості кожного окремого об'єкта та його компонентів. Компоненти - це блоки, які можуть бути додані до ігрового об'єкта, з метою зміни його властивостей та надання додаткової функціональності, що дозволяє здійснювати більш глибоку взаємодію. Наприклад, компонент "Рух" може додати можливість переміщення ігрового об'єкта по сцені, а компонент "Колізія" може дозволити об'єкту взаємодіяти з іншими об'єктами на тій же сцені.

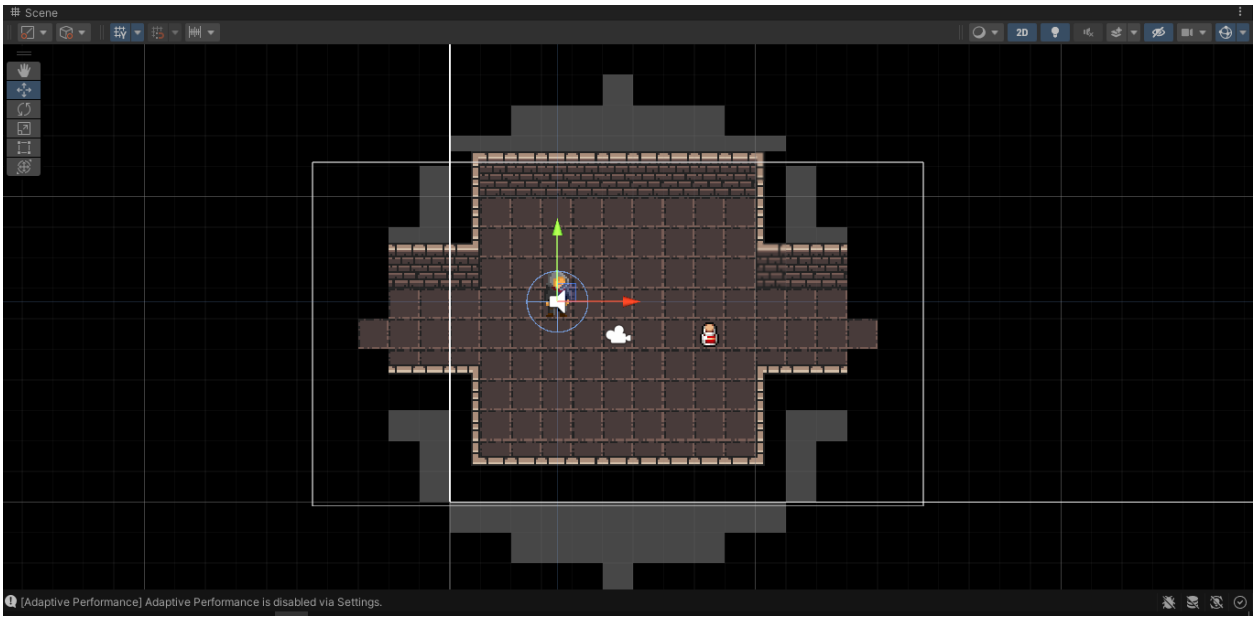

Рисунок 3. Редактор сцени

Усі ці інструменти дозволяють розробникам створювати складні та деталізовані ігри з різноманітними ігровими механіками та інтерактивним світом.

Якщо зробити підсумок, то Unity - це потужний та гнучкий інструмент для розробки ігор, який дозволяє розробникам створювати багатофункціональні та цікаві ігри для різних платформ. Завдяки своїй простоті використання, великій кількості ресурсів та великій спільноті

розробників, Unity є одним з найпопулярніших інструментів для розробки ігор у світі.

Розділ 2 : Аналіз задачі

2.1 Огляд схожих ігор

2.1.1 Огляд “Soul Knight”

Soul Knight - це безкоштовна мобільна гра, яка була розроблена ChillyRoom у 2017 році. Гра доступна на Android та iOS і є дуже популярною серед гравців з усього світу(понад 50 млн завантажень в Google Play на 2023 рік).

Головна мета гри – пройти якомога більше рівнів, борючись з ворогами в різних локаціях. Гравець має збирати різноманітну зброю та інші ресурси, які допоможуть йому в його пригодах. Крім того, гра пропонує багато персонажів, кожен з яких має унікальні вміння та характеристики. Геймплей гри відбувається в реальному часі.

Гра має власні унікальні піксельні графічні ефекти, які роблять її дуже привабливо для гравців. Крім того, музика та звукові ефекти також додають грі ще більше атмосферності.

Дану гру було розроблено за допомогою ігрового рушія Unity.

Рисунок 4. Знімок екрана з гри “Soul Knight”

2.1.2 Огляд “OneBit Adventure”

OneBit Adventure – це ретро стильова мобільна гра, яка пропонує гравцям досліджувати випадково згенеровані світи у 8-бітному піксельному

візуалі. Гра має доволі простий, але цікавий покроковий геймплей, що потребує стратегічного мислення.

Гра починається зі створення персонажа, обирається його клас та зовнішній вигляд. Після цього гравець опиняється в світі гри, де йому необхідно боротись з монстрами та збирати різні ресурси.

Як і попередній приклад дана гра є розробленою з використанням Unity.

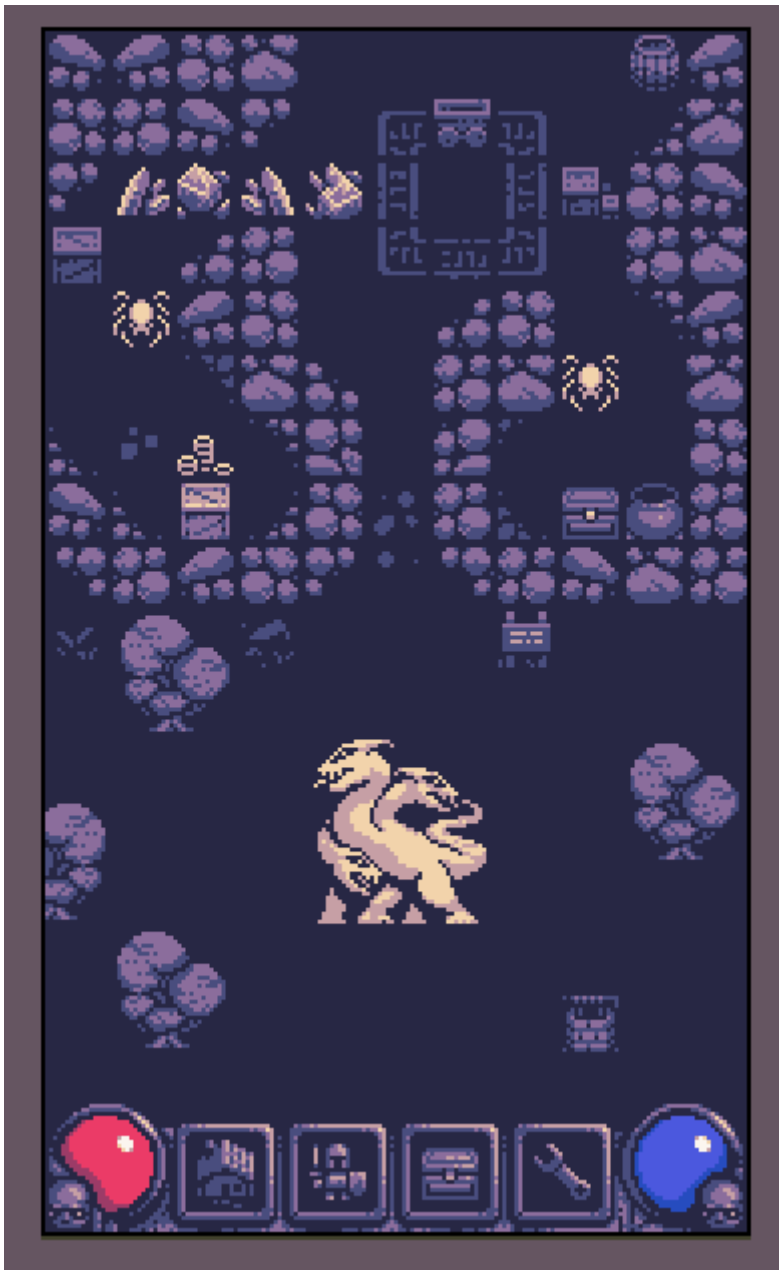

Рисунок 5. Знімок екрана з гри “OneBit Adventure”

2.2 Функціональні вимоги

Проаналізувавши специфіку даного жанру та схожі ігри, а також враховуючи власні інтереси було визначено функціональні вимоги, які необхідно реалізувати для того, щоб отримати гру, в яку вже можливо грати:

- Інтуїтивно зрозумілий інтерфейс.
- Покрокова система дій(рух, атака тощо) головного героя та його ворогів.
- Система інвентаря гравця.
- Генерація ігрових рівнів.
- Рівень гравця та атрибути, які збільшуються зі зростанням рівня.
- Мінімум 2 види ворогів.
- Мінімум 2 види зброї.
- Мінімум 2 види броні.
- Збереження прогресу при виході з гри.

Розділ 3 : Реалізація гри

3.1 Концепція

Основний жанр даної гри RPG, тому спершу необхідно визначити які ігри відносять до даного жанру. RPG (Role-Playing Game) – жанр відеоігор, у якому гравець курує діями персонажа (або персонажів), у якомусь чітко визначеному світі, зазвичай передбачаючи певну форму розвитку персонажа шляхом запису статистики. До ігор даного жанру можна віднести серії Fallout, The Elder Scrolls, The Witcher.

Події гри відбуваються в фентезійному світі, гравець бере на себе роль шукача пригод, який хоче захисти поселення знищивши усіх монстрів, для цього він відправляється в підземелля. Смерть кожного персонажа є остаточною, тому якщо гравець зазнає поразки він починає спочатку. Проте чим глибше гравець дістається у підземелля, тим більшу вигоду отримає поселення. Куди інвестувати зароблені монети гравець може обрати сам, це може бути стартова зброя, додаткове зілля, покращення певних параметрів тощо.

Назва гри - Shadow Crawler - посилається на піджанр ігор Dungeon crawl, який зосереджується на подорожах героїв підземеллями у фентезійних RPG. Гра також містить елементи Roguelike(ігри Hades, Dead Cells, Darkest Dungeon тощо), такі як процедурно генеровані рівні, покроковий геймплей, плиткова графіка та перманентна смерть персонажа у випадку поразки.

3.2 Головне меню

Рисунок 6. Головне меню

Після запуску гри гравця зустрічає головне меню з 5-ма кнопками:

“Dungeon” – кнопка яка починає/продовжує основний ігровий процес.

“Village” – кнопка, яка відкриває меню поселення, та дозволяє інвестувати зароблено золото в певного ремісника.

Рисунок 7. Меню "Village"

“Tutorial” – кнопка, яка відкриває вікно з короткою довідкою інформацією по грі.

Рисунок 8. Меню "Tutorial"

“Settings” – відкриває меню налаштувань гри, дозволяє налаштувати гучність різних ігрових елементів.

Рисунок 9. Меню "Settings"

“Exit” – вихід з гри.

3.3 Ігровий інтерфейс

Рисунок 10. Елементи ігрового інтерфейсу

1. Інформація про рівень гравця. Відображає теперішній рівень гравця (1), також по наповненості світло синім кольором даного елемента можна зрозуміти наскільки багато досвіду гравцю ще потрібно, щоб досягти нового рівня (досвід отримується за знищення монстрів). Якщо натиснути на даний елемент, то камера відцентрується на головному персонажі, а також буде виведено екран зі значеннями атрибутів та більше детальна інформація про прогрес рівня. Всього є 3 атрибути, які підвищуються на певне значення, коли гравець досягає нового рівня: здоров’я(скільки шкоди може витримати герой), атака(скільки в середньому шкоди герой наносить монстрам), захист (на яке значення зменшується шкода, яку отримує герой)

Рисунок 11. Вікно зі статистикою персонажа

2. Ігровий рівень/глибина підземелля на якому знаходиться зараз персонаж. Всього рівнів 5, чим далі гравець пробереться, тим більшу нагороду отримує.
3. Шкала здоров’я персонажа
4. Підменю, де гравець може зберегти гру та повернутись до головного меню, або почати гру спочатку

Рисунок 12. Підменю

5. Меню підбору речей, що відображає всі предмети, які лежать на тій самій клітинці, де знаходиться гравець і відповідно він їх може взяти. Якщо предметів більш як два, то з’являється можливість прокручування списку.
6. Плаваючи значення шкоди. Червоні числа показують скільки шкоди гравець наніс, оранжеві – скільки шкоди гравець отримав
7. Шкала здоров’я монстра
8. Кнопка інвентаря. Показує наявні у гравця предмети та дозволяє з ним взаємодіяти. Всього гравець має 6 слотів у рюкзаку, 1 слот для спорядженої зброї та 1 слот для спорядженої броні. Броню чи зброю можна одягнути/зняти або викинути. Зілля можна випити для відновлення частини здоров’я або викинути для того, щоб звільнити слот.

Рисунок 13. Інвентар гравця

9. Кнопка пропуску ходу

Натискання кожної кнопки як і в меню супроводжується стандартним або унікальним(інвентар, спорядження зброї) звуковим ефектом.

3.4 Головний герой та інші персонажі

Для всіх персонажів було створено спільний базовий клас Unit, який містить ScriptableObject(скриптований об’єкт, який в Unity виступає як контейнер для даних) під назвою UnitData, що містить загальні дані кожного

персонажа. Крім того, клас Unit містить функції властиві всім персонажам(рух, атака тощо). Також було створено окремі класи PlayerUnit та EnemyUnit, які наслідуються від класу Unit та визначають поведінку притаманну саме головному герою чи противникам.

```
public class UnitData : ScriptableObject
{
    public new string name;
    public int maxHP;
    public int attack;
    public int maxAttackDistance;
    public int pathfindingXMaxDistance;
    public int pathfindingYMaxDistance;
    public int defense;
    public int expReward;
    public float moveSpeed;
    public Sprite sprite;

    [System.Serializable]
    2 references
    public class Drop
    {
        public ItemData itemData;
        public float dropChance;
    }

    public List<Drop> drops;
}
```

Рисунок 14. Клас UnitData

Для знаходження найкоротшого шляху гравця до обраної клітинки було використано алгоритм пошуку A*(“A зірочка”). Вороги завжди намагаються наблизитись до гравця (якщо він на певній відстані від них) на відстань атаки, використовуючи цей же алгоритм.

Рисунок 15. Відображення шляху гравця

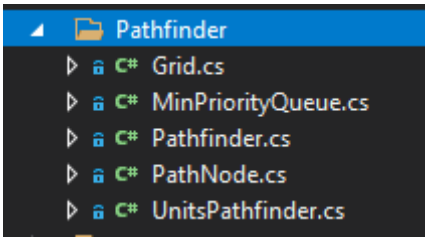

Рисунок 16. Класи для реалізації A та його взаємодії з іншими елементами гри*

3.5 Інвентар головного героя

Гравець може носити з собою до 6 предметів(не враховуючи споряджену зброю та броню). На спорядження, використання, підбирання або викидання предмета витрачається хід героя, що не дозволить гравцю вільно міняти зброю чи користуватись зіллями в бою.

Загалом в грі всього 9 предметів:

- 4 види зброї
- 4 види броні
- 1 вид зілля

Рисунок 17. Всі ігрові предмети

3.6 Генерація рівнів

Для кожного рівня генерується шаблон – двовимірний масив, на основі якого потім створюється відповідна сцена в Unity. За створення даного шаблону відповідає клас TempleteGenerator, який на основі певних параметрів(розмір, кількість кімнат тощо) генерує рівень. Всі об’єкти(стартова кімната, кімната з противниками, коридор тощо) в шаблоні представляються цілими числами (ідентифікаторами). В центрі створюється стартова кімната, у випадкових місцях створюються кімнати з противниками, коридори створюються через уже згаданий алгоритм A*. Для того, щоб зробити коридори більш звивистими перед їх генерацією створюються “невидимі” перешкоди, які пізніше вилучаються з шаблону.

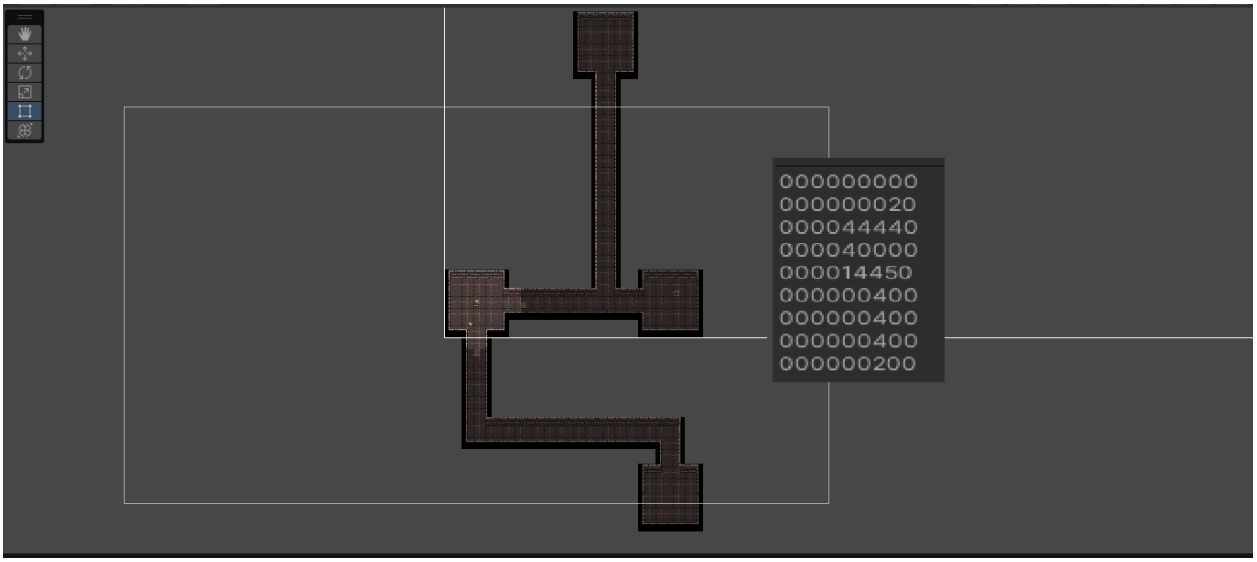

Рисунок 18. Рівень та його шаблон

Для генерації ворогів та предметів на рівні також є своє відповідні класи. Загалом для кожного ворога чи предмета задається кількість яка гарантовано згенерується в кімнаті, шанс генерації додаткових ворогів/предметів, а також параметри, що відповідають за зміну цих значень залежно від рівня.

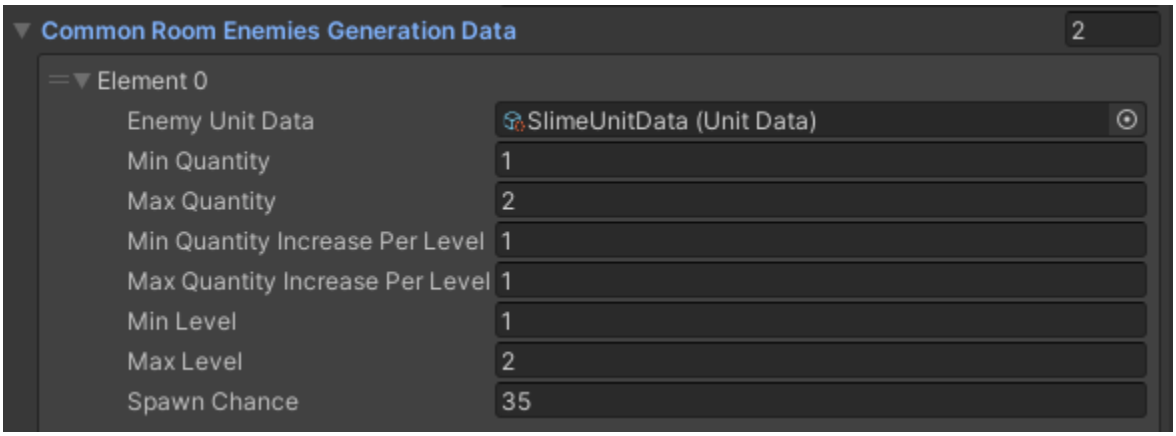

Рисунок 19. Параметри генерації противників типу "Slime"

Наприклад зі знімку екрана вище можна зрозуміти, що на 1-му рівні в кожній звичайній кімнаті є мінімум 1 слайм, також може згенеруватись другий з шансом 35% відсотків. На 2-му рівні відповідно гарантована кількість слаймів зростає до 2-х, а максимальна до 3-х. Для 3-го рівня даний запис не актуальний (“Max Level 2”).

3.7 “Туман війни”

“Туман війни” є ігровою умовністю - зазвичай темним чи напівпрозорим покривом, що приховує деталі місцевості та розташування ворогів.

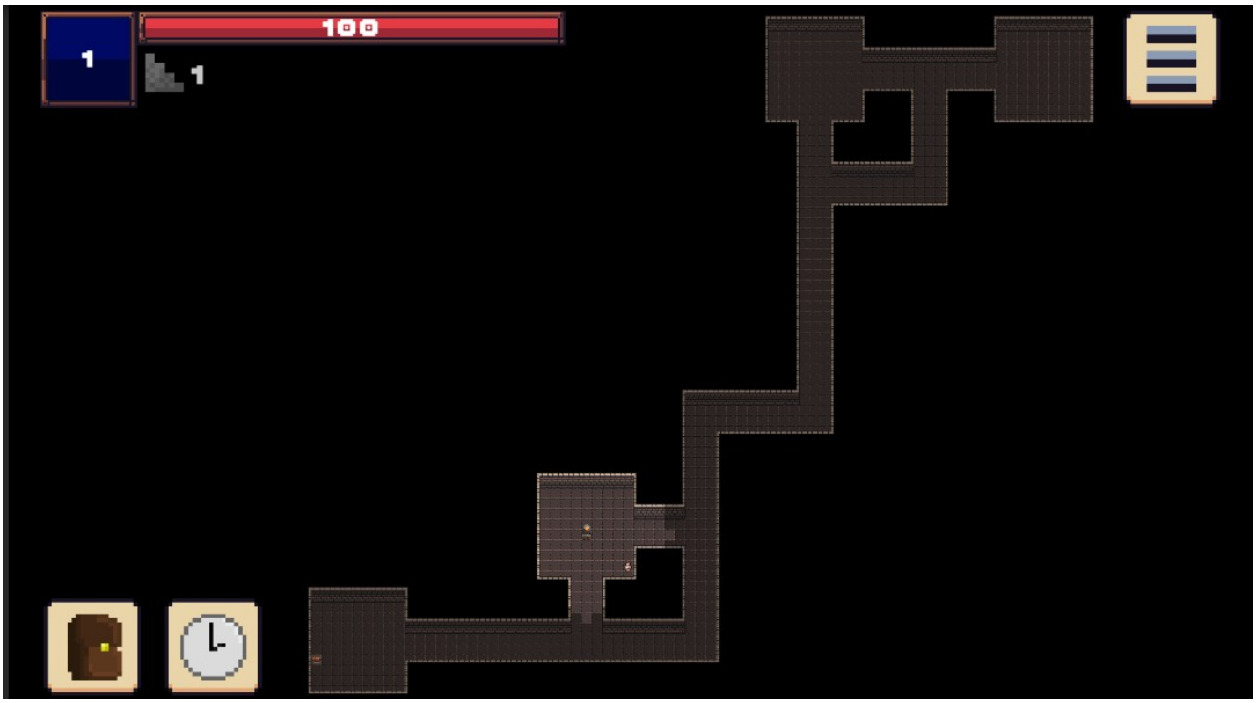

Рисунок 20. Повністю досліджений ігровий рівень

Спершу гравець матиме інформацію лише про ті тайли (ігрові клітинки), які в його полі зору та на доступній відстані. Досить порухатись в якомусь напрямку, що відкрити сусідні тайли. Проте бачити предмети чи ворогів гравець може лише на клітинках в його полі зору (тайли які не полі зору гравця виглядають затемненими).

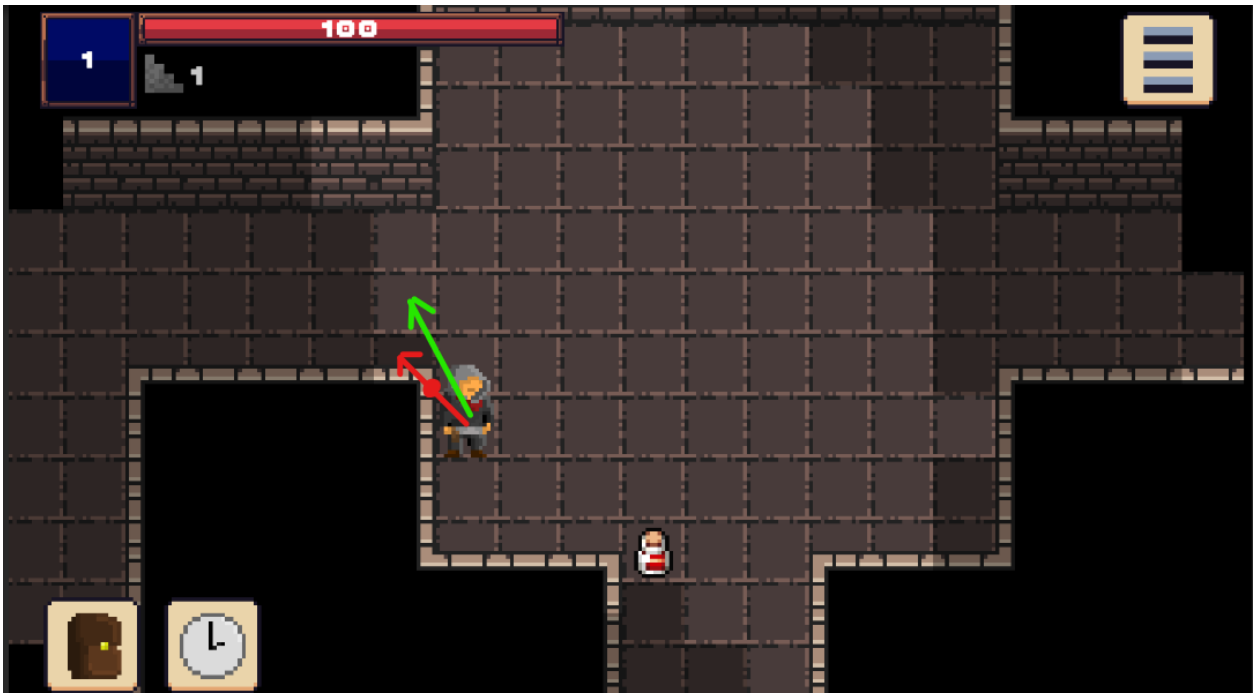

Рисунок 21. Поле зору гравця

3.8 Звуки

На початку розробки була проблем з обриванням звуків. Наприклад при натисканні кнопки “Dungeon” в головному меню (початок гри) звук кнопки різко обривався. Відбувалось це через те, що Unity при переході до нової сцени, знищує всі об’єкти попередньої і відповідно об’єкт(в даному випадку кнопку), за яким було закріплено цей звук, теж знищувався.

У вирішення цього питання допомагає функція DontDestroyOnLoad, яка запобігає знищенню об’єкта при завантаженні нової сцени. Було створено новий об’єкт під назвою CrossSceneAudioManager та прикріплено до нього однойменний скрипт який викликає вищезгадану функцію в методі скрипта Awake(метод що викликається, щойно скрипт завантажено), а також всі необхідні звуки. Це дало змогу адекватно відтворювати всі звуки у будь-яких обставинах.

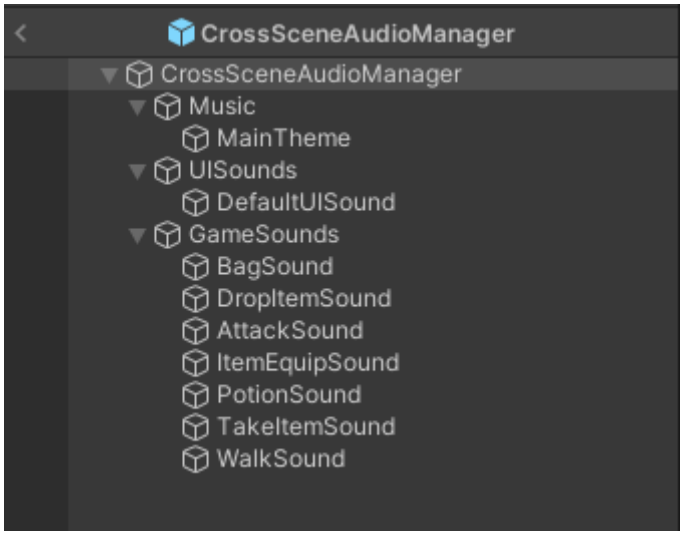

Рисунок 22. Об'єкт CrossSceneAudioManager

3.9 Анімації

Було анімовано гравця та всіх ігрових монстрів. Всі анімації були створені вручну, за основу взяті безкоштовні зображення з інтернет-ресурсів. Для гравця передбачено 3 анімації: звичайна(коли ніяких дій не відбувається), рух, атака. Для ворогів додатково реалізовано анімацію смерті.

Логіку відтворення анімацій було прописано з використання вбудованих в Unity інструментів.

Рисунок 23. Таблиця спрайтів одного з анімованих ворогів

Висновки

Під час даної роботи проведено огляд двох найпопулярніших ігрових рушіїв – Unreal Engine та Unity. Були визначені основні переваги та недоліки кожного з цих інструментів.

Більш детально досліджено рушій Unity, який добре підходить для розробки мобільних ігор. Також зазначено легкість освоєння даного інструменту, який вирізняє його на фоні конкурентів.

В результаті розроблено піксельну покрокову 2D RPG Roguelike гру під назвою Shadow Crawler. Реалізовано основний функціонал: покроковий рух та бій з монстрами, процедурна генерація рівнів, інвентар, збір та використання предметів, ігрові атрибути(здоров’я, атака, захист) та їх покращення з підвищенням рівня, збереження прогресу при виході з гри. Також були додані додаткові можливості у вигляді перманентних покращень, та “системи зору” персонажа, яка показує певні ігрові об’єкти тільки тоді коли між ними й головним героєм нема перешкод.

Список використаних джерел

1. Introduction to Game Engines [Інтернет стаття] – Режим доступу до ресурсу: https://medium.com/techiepedia/introduction-to-game-engines-8dd8cd8af03e
2. Unity 2021 Gaming Report [Електронний ресурс] – Режим доступу до ресурсу: https://create.unity.com/2021-game-report
3. Unity Asset Store [Електронний ресурс] - https://assetstore.unity.com/
4. Pixabay.com [Електронний ресурс] – Режим доступу до ресурсу: https://pixabay.com/
5. Itch.io [Електронний ресурс] – Режим доступу до ресурсу: https://itch.io/
6. User Manual 2021.3 (LTS) [Електронний ресурс] – Режим доступу до ресурсу: https://docs.unity3d.com/Manual/index.html
7. Developing 2D Games with Unity: Independent Game Programming with C# / Jared Halpren., 2018 – (Apress).
8. A* Search Algorithm [Інтернет стаття] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/a-search-algorithm/
9. Soul Knight [Електронний ресурс] – Режим доступу до ресурсу: https://play.google.com/store/apps/details?id=com.ChillyRoom.DungeonShooter
10. OneBit Adventure (Roguelike) [Електронний ресурс] – Режим доступу до ресурсу: https://play.google.com/store/apps/details?id=com.GalacticSlice.OneBitAdventure