ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Комп'ютерна гра "Lost Element" 2D платформер із сайтом вікіпедії та використанням алгоритмів генерації залів, інтелектуальної поведінки ботів (комплексна робота)»

Ступінь вищої освіти – бакалавр
Спеціальність – 122 «Комп’ютерні науки»
Освітня програма «Комп’ютерні науки»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: здобувач 4 курсу
групи КН-21
Дмитро ПРОЦЕНКО
Керівник: зав. кафедри комп'ютерних наук
к.е.н., с.н.с, доцент
Сергій МІЧКІВСЬКИЙ

Засвідчую, що кваліфікаційна робота оформлена відповідно до ДСТУ 3008:2015 та не містить запозичень з праць інших авторів без відповідних посилань.
Здобувач: _____
(підпис)

м. Київ – 2025 рік

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

ЗАТВЕРДЖУЮ:
завідувач кафедри
комп’ютерних наук
_____Сергій МІЧКІВСЬКИЙ
«___»____20___р

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Проценко Дмитро Олександрович

Тема роботи	Комп'ютерна гра "Lost Element" 2D платформер із сайтом вікіпедії та використанням алгоритмів генерації залів, інтелектуальної поведінки ботів (комплексна робота)
Номер та дата наказу про затвердження теми	№121-7 від 24 грудня 2024 року
Коротка постановка завдання	Розробити 2D комп’ютерну гру, платформер з використанням алгоритмів генерації залів, інтелектуальної поведінки ботів, а також UI/UX для гри; Розробити сайт вікіпедію з внутрішнім магазином, а також UI/UX для сайту.
Посилання на джерела інформації (не більше п’яти найменувань, які рекомендує науковий керівник)	1. Моделювання поведінки неігрових персонажів у комп'ютерних іграх : Н.О. СОКОЛОВА, В.В.ГНАТУШЕНКО, М.С. МІЩЕНКО, О.А. АТАМАНЧУК, Національний технічний університет «Дніпровська політехніка» 2. Чеботарьова І. Б., Черкашина Г. І. Основні тренди UI/UX дизайну 2024. Поліграфічні, мультимедійні та web-технології : матеріали Молодіжної школи-семінару IX Міжнар. наук.-техн. конф., 14-28 травня 2024 р. – Харків : ТОВ «Друкарня Мадрид», 2024. – Т. 2. – С. 40-47
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має передбачити теоретичне, системотехнічне або експериментальне дослідження складного спеціалізованого завдання або практичної проблеми в галузі комп’ютерних наук, яке характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів інформаційних технологій.

Дата видачі завдання 27 грудня 2024 р.

Керівник Сергій МІЧКІВСЬКИЙ

Здобувач освітнього ступеня бакалавра Дмитро ПРОЦЕНКО

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
Підготовчий етап			
1	Вибір напряму дослідження	02.12.2024 р.	*виконано*
2	Формування теми та призначення керівника	16.12.2024 р.	*виконано*
3	Затвердження теми кваліфікаційної роботи	23.12.2024 р.	*виконано*
4	Затвердження завдання на кваліфікаційну роботу	27.12.2024 р.	*виконано*
Основний етап			
5	Розробка концепції кваліфікаційної роботи	13.01.2025 р.	*виконано*
6	Підбір та вивчення джерел інформації з напряму дослідження. Огляд існуючих аналогів	20.01.2025 р.	*виконано*
7	Затвердження розширеної постановки завдання. Підготовка та подання керівникові розділу 1 кваліфікаційної роботи	10.03.2025 р.	*виконано*
8	Проєктування. Підготовка та подання керівникові розділу 2 кваліфікаційної роботи	24.03.2025 р.	*виконано*
9	Підготовка доповіді для експертизи стану виконання кваліфікаційної роботи (проміжний контроль)	31.03-04.04.2025 р.	*виконано*
10	Реалізація. Підготовка та подання керівникові розділу 3 кваліфікаційної роботи	07.04.2025 р.	*виконано*
11	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	14.04.2025 р.	*виконано*
12	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівникові доопрацьованого варіанту кваліфікаційної роботи	21.04.2025 р.	*виконано*
Завершальний етап			
13	Представлення рукопису для перевірки на плагіат	28.04-04.05.2025 р.	*виконано*
14	Підготовка презентації та доповіді на передзахист	05.05-11.05.2025 р.	*виконано*
15	Передзахист кваліфікаційної роботи	12.05-16.05.2025 р.	*виконано*
16	Доопрацювання роботи за результатами передзахисту	19.05-06.06.2025 р.	*виконано*
17	Експертиза роботи керівником та зовнішнім експертом	09.06-15.06.2025 р.	*виконано*
18	Доопрацювання доповіді та презентації для захисту	09.06-15.06.2025 р.	*виконано*
19	Захист кваліфікаційної роботи	16.06-22.06.2025 р.	*виконано*

Керівник Сергій МІЧКІВСЬКИЙ

Здобувач освітнього ступеня бакалавра Дмитро ПРОЦЕНКО

Проценко Д.О. Комп'ютерна гра "Lost Element" 2D платформер із сайтом вікіпедії та використанням алгоритмів генерації залів, інтелектуальної поведінки ботів (комплексна робота).

Пояснювальна записка кваліфікаційної роботи за спеціальністю 122 – Комп’ютерні науки (освітня програма – Комп’ютерні науки) ОС Бакалавр – ВНЗ «Університет економіки та права «КРОК», Навчально-науковий інститут інформаційних та комунікаційних технологій, кафедра комп’ютерних науки, Київ, 2025.

Розглянуто проблему індустрії комп’ютерних ігор на прикладі розробки платформера «Lost Element». Реалізовано комп’ютерна гра «Lost Element» на рушії Unity з застосуванням алгоритмів генерації залів і інтелектуальної поведінки ботів. Реалізовано сайт вікіпедію гри з внутрішнім магазином.

Ключові слова: скрипт, комп’ютерна гра, Unity.

Рис. 27 Бібліограф.: 16 найм.

Protsenko D.O. Computer game "Lost Element" 2D platformer with a Wikipedia site and the use of algorithms for generating halls, intelligent behavior of bots (complex work)

Explanatory note of qualification work in specialty 122 - Computer Science (educational programme - Computer Science), Bachelor's degree - University of Economics and Law "KROK", Educational and Research Institute of Information and Communication Technologies, Department of Computer Science, Kyiv, 2025.

The problem of the computer game industry is considered using the example of the development of the platformer "Lost Element". The computer game "Lost Element" is implemented on the Unity engine using room generation algorithms and intelligent bot behavior. A Wikipedia site for the game with an internal store is implemented.

Keywords: script, computer game, Unity.

Fig. 27. Bibliography: 16 Items.

ЗМІСТ

ВСТУП

Актуальність теми. Жанр платформер вже давно користується великою популярністю серед гравців завдяки своїй простоті та відчуттю ностальгії. Класичні ігри, наприклад, Super Mario та Sonic заклали основу жанру, який залишається популярним і до сьогодення.

2D платформери залишаються актуальним і до тепер, це можна побачити на прикладі популярної гри Hollow Knight, яка в свій час отримала оцінку 10/10 у відомих ігрових видань Game Informer, Destructoid та PC Gamer [3]. Також на сьогоднішній день можна очікувати ще багато нових релізів у даному жанрі, що вказує на життєздатність та високий інтерес гравців.

Розробка 2D платформерів є першим кроком для початкових розробників, завдяки відносній простоті у створені та можливості експериментувати з ігровими механіками. А такі ігрові рушії, як Unity, спрощують процес розробки та дозволяють зосередитися на інших аспектах гри.

Мета дослідження. Розробити комп'ютерну 2D гру «Lost Element» в жанрі платформер зі сайтом вікіпедія, з використанням алгоритмів генерації залів та інтелектуальної поведінки ботів.

Завдання дослідження:

- дослідити існуючі комп'ютерні ігри жанру платформер та визначити їх переваги та недоліки;
- створити концепт гри;
- спроектувати основну архітектуру гри та алгоритмів;
- розробити програмне забезпечення комп'ютерної гри «Lost Element» з використанням алгоритмів генерації залів та інтелектуальна поведінка ботів.

Об'єктом дослідження є ігровий процес комп'ютерної 2D гри в жанрі платформер.

Предметом дослідження комп’ютерні ігри та їх механіки, візуальні стилі 2D ігор в жанрі платформер.

Методи дослідження. Підходи та алгоритми генерації залів та інтелектуальної поведінки ботів.

Практичне значення. Розроблена комп’ютерна 2D гра «Lost Element» в жанрі платформер зі сайтом вікіпедія, з використанням алгоритмів генерації залів та інтелектуальної поведінки ботів.

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків та списку посилань (16 найменувань). Пояснювальна записка містить 27 рисунків. Загальний обсяг пояснювальної записки становить 47 сторінок, основний зміст викладено на 45 сторінках.

РОЗДІЛ 1
ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ КОМПЮТЕРНОЇ ГРИ «LOST ELEMENT»

1.1 Опис предметної області

Відеогра це як книга зі своєю історією і персонажами, де ви не просто спостерігаєте за подіями, а й приймаєте в ній участь. У деяких випадках вам навіть доводиться приймати рішення, які кардинально змінюють хід сюжету.

Комп'ютерні ігри вже давно стали невід'ємною частиною сучасної культури та індустрії розваг. Їхнє головне призначення приносити задоволення та дарувати гравцям незабутні емоції. Це може відбуватися як через глибокий сюжет, так і завдяки добре прописаним персонажам.

Ігрова індустрія налічую велику кількість жанрів та під жанрів, основними можна виділити пригоди, рольові, стратегії, симулятори та екшен.

Пригоди (Adventure) – жанр комп'ютерних ігор, який зосереджується на досліджені світу, поданні сюжету та вирішенні головоломок. Основний акцент поставлено на сюжетній складовій та логічних завданнях, тому зазвичай у таких іграх бойові механіки або повністю відсутні, або відіграють мінімальну роль. До жанру входять інтерактивна література, графічні пригоди, пригоди-головоломки, візуальні новели, інтерактивні фільми.

Рольові (RPG Role-Playing Games) – жанр, який зосереджується на прокачуванні свого персонажу шляхом дослідження ігрового світу, проходження сюжету та другорядних завдань. Важливими аспектами жанру також є сюжет, який часто залежить від рішень гравця та створення персонажу, що дає можливість не тільки налаштовувати зовнішній вигляд персонажу, а й обирати клас та інші характеристики. Різновидами жанру є японські RPG, західні RPG, тактичні RPG та MMORPG.

Стратегії (Strategy) – жанр, головною метою якого є розробка плану розвитку та прийняття тактичних та стратегічних рішень для досягнення поставлених цілей. Як правило, на початкових етапах надається обмежена кількість ресурсів та територій, тому для подальшого розвитку гравцю доведеться обережно розпоряджатися ресурсами. Жанр поділяється на підвиди покрокова стратегія та стратегія в реальному часі.

Симулятори (Simulation) – жанр, який зосереджується на відтворені процесів реального світу у грі, що дозволяє гравцю керувати ними або взаємодіяти. В більшості випадковість це симуляції різних професій, управління транспортними засобами, природними явищами, тваринами, тощо.

Екшен (Action) – жанр ігор, основним аспектом якого є бойові сцена та динамічний геймплей, що вимагає у гравців координації та достатньої швидкості реакції. Екшн поділяється на піджанри: файтинг, платформер, шутер, тощо.

Платформер – піджанр екшн ігор, який зосереджується на проходженні рівнів, які складаються з різних платформ. Основною механікою піджанру є стрибки, які дають можливість персонажу долати перешкоди та переміщатися між платформами.

1.2 Аналіз потенційних конкурентних переваг програмного продукту

Celeste (рис. 1.1) — комп’ютерна гра в жанрі платформер. Сюжет повідає нам історію дівчинки на ім’я Медлін, яка страждає від тривоги і депресії. Її головна мета піднятися на вершину гори Селеста. Гра поділена на дев’ять розділів, які складаються з декількох кімнат. Кожен розділ має свої механіки та сюжет, і більшість з них містять додаткові колекційні предмети [1].

Переваги:

- чарівний художній стиль;
- прості механіки;
- глибокий сюжет;

– дизайн рівнів.

Недоліки:

– висока складність для новачків.

Рисунок 1.1 — Celeste

Джерело [2]

Hollow knight (рис. 1.2) — комп’ютерна гра в жанрі метроїдванія. Події гри відбуваються у давно покинутому королівстві Халлоунест. Сюжет розповість історію безіменного жука, якого персонажі називають просто Лицар. Йому доведеться подолати безліч пасток та здолати велику кількість босів, щоб виконати своє призначення.

Переваги:

– атмосфера;

– звуковий супровід;

– різноманітність ворогів і босів;

– бойова система;

– візуальний стиль.

Недоліки:

– орієнтація на місцевості виконана досить погано.

Рисунок 1.2 — Hollow knight

Джерело [3]

Ori and the Blind Forest (рис. 1.3) — комп'ютерна гра в жанрі платформер, метроїдванія. Сюжет представляє собою розповідь історії події, якої відбуваються в Нібельському лісі, яке населяють тварини й духи. Подальше лихо змушує ліси в'янути. Задля порятунок свого дому, маленький дух на ім'я Орі вирушає через весь ліс, щоб запобігти подальшому катаклізму.

Переваги

– візуальний стиль;

– дизайн локацій;

– битва з босами;

– глибокий сюжет.

Недоліки

– малий час проходження;

– гра не стимулює гравця досліджувати світ.

Рисунок 1.3 — Ori and the Blind Forest

Джерело [4]

Cuphead (рис. 1.4) — комп’ютерна гра в жанрі платформер. Сюжет розповідає історію про двох персонажів Капхеда та Магмена, які пішли грати в казино та програли свої душі Дияволу. Аби врятувати душі він дає завдання зібрати до опівночі наступної доби контракти на душі, які були викрадені іншими боржниками. Після отримання здібності стріляти з пальців вони вирушають на пошуки.

Переваги

- графіка виконана у комічному стилі старих анімацій Disney 30-х років;
- кооператив;
- різноманітність бойових стилів;
- унікальні боси;
- цікава побудова боїв.

Недоліки

- висока складність гри.

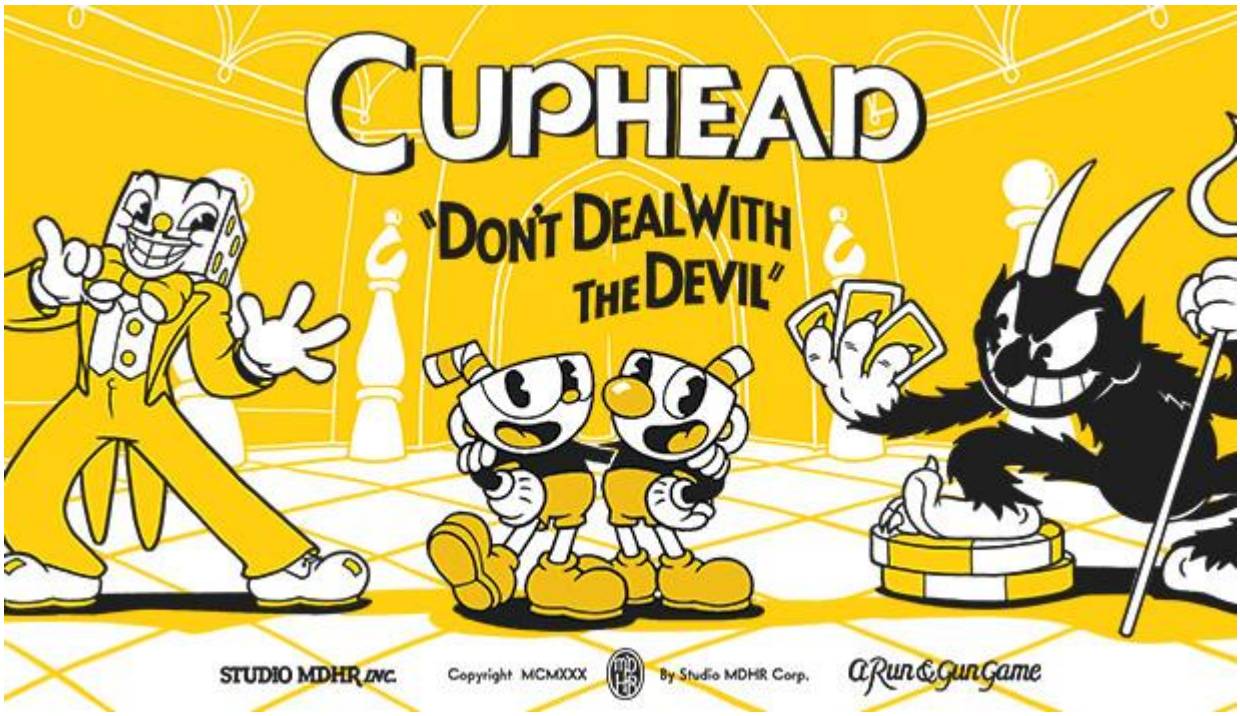


Рисунок 1.4 — Cuphead

Джерело [5]

1.3 Постановка завдання

Правила гри:

- базові механіки: персонаж матиме базовий набір механік: ходьба, стрибки, біг та атаки ближнього бою.
- дослідження локацій: гравець зможе самостійно обирати маршрути для проходження. Локації будуть мати кімнати з загадками, які персонаж повинен вирішити щоб отримати нагороду. Також будуть присутні секретні кімнати, знайти їх можна досліджуючи локацію.
- взаємодія: гравець зможе взаємодіяти з деякими предметами, будівлями, персонажами, тощо.
- додаткові механіки: система збереження буде реалізована у вигляді точок з якими гравець має взаємодіяти. При загибелі персонажу, його буде перенесено до останньої точки збереження.

Учасниками гри люди віком від 15-35 років, до їхнього числа можуть входити як фанати ретро та платформерів, так і люди, які полюбляють дослідження та відкритий світ.

При розробці планується використання автоматизованих функцій:

- зміна стану анімацій персонажу;
- оновлення карти при дослідженні;
- генерація локацій при новій грі;
- спавн ворогів у відповідних зонах;
- поведінка ворогів: патрулювання, атака, переслідування;
- збереження статусу ворогів – вбиті вороги не відроджуються, поки гравець не взаємодіє з об'єктом збереження;
- спеціальні тригери подій для зустрічі з босом та запуску кат-сцени;
- система діалогів.

Висновки до розділу 1

Проведено дослідження предметної області продукту. Аналіз жанрів комп’ютерних ігор та обрання жанру для майбутнього продукту.

Проведено аналіз потенційних аналогів продукту, опис їхніх переваг та недоліків.

Визначені завдання такі, як розробка базових механік, взаємодія гравця з об’єктами локації та персонажами, система збережень, моделювання поведінки ворогів та босів, створення візуального стилю світу та звукового супроводження.

РОЗДІЛ 2
ПРОЄКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

2.1 Модулювання поведінки функцій продукту

Функціональні вимоги:

- гравець повинен мати можливість керувати персонажем;
- персонаж повинен взаємодіяти з об'єктами локації;
- коли ворог атакує персонажа, його здоров'я зменшується; гра має містити систему збереження;
- після втрати всього здоров'я персонаж відроджується на останній точці збереження;
- вороги повинні мати задану траєкторію патрулювання та переслідування персонажу;
- гра має містити головне меню та HUD персонажу.

Нефункціональні вимоги:

- кожна дія має супроводжуватися звуковим ефектом;
- звуки та музика повинні змінюватися залежно від локації;
- гра повинна працювати на операційній системі Windows;
- зрозумілий інтерфейс
- гра повинна працювати стабільно та не мати критичних помилок, які б зупиняли проходження гри;
- підтримка клавіатури.

На рис. 2.1 представлено діаграму прецедентів «Функціональність гри».

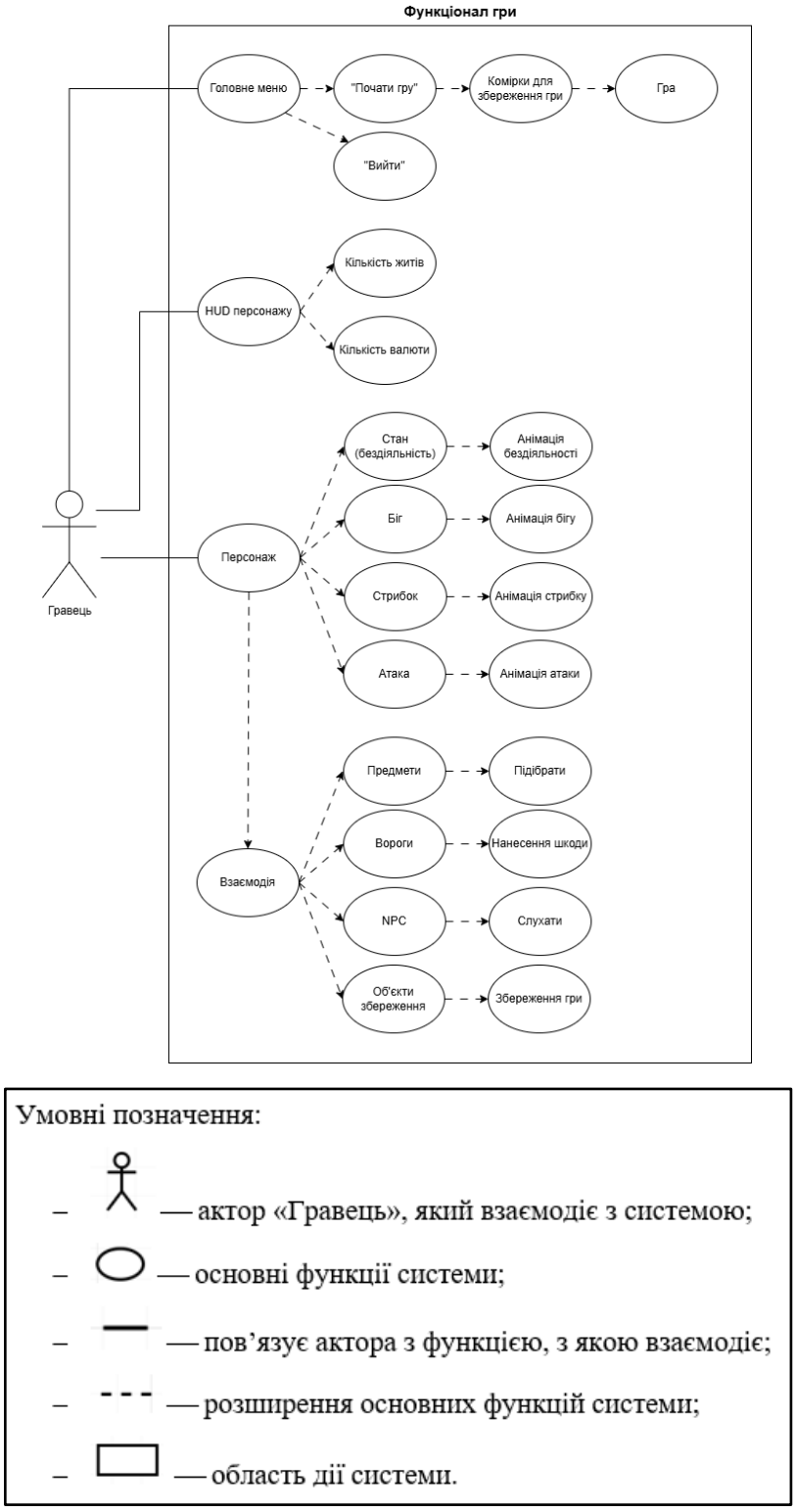


Рисунок 2.1 —Діаграма прецедентів «Функціональність гри»

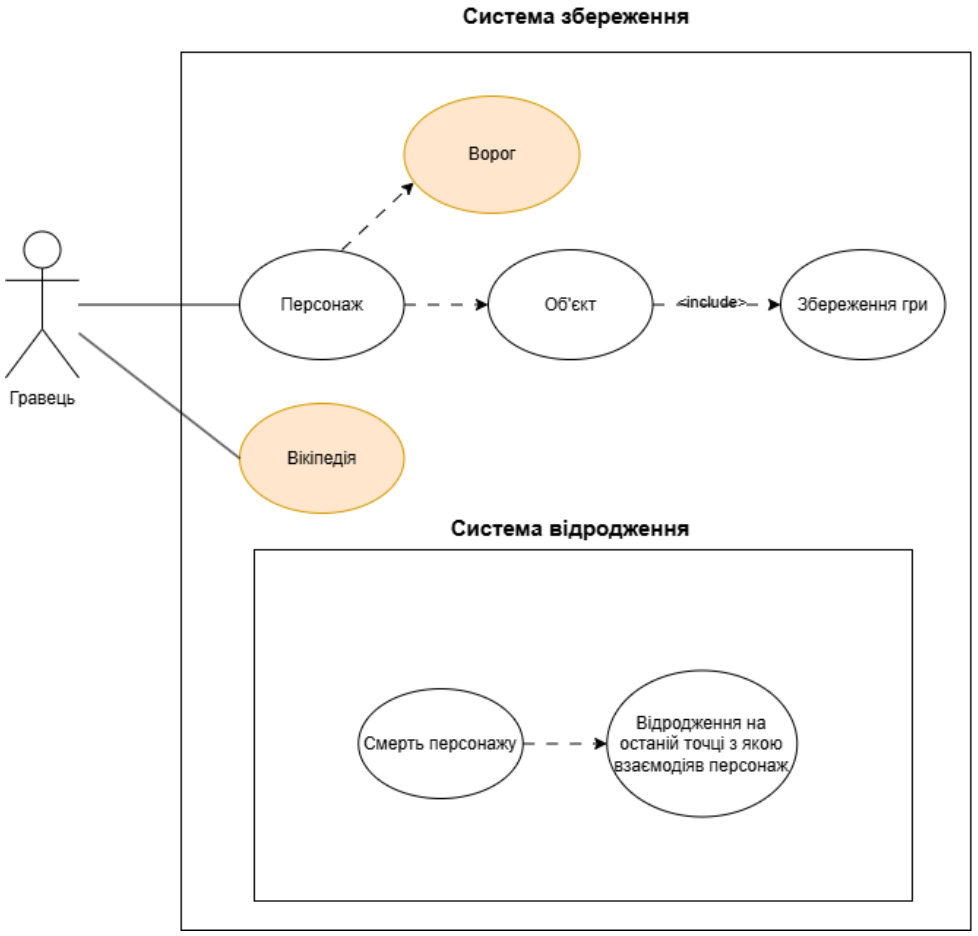
Джерело: розроблене автором

Пояснення діаграми. Гравець має можливість взаємодіяти з функціями гри:

- головне меню: гра починається з головного меню. Воно складається з кнопки «Почати гру» та «Вийти». При натисканні кнопки «Вийти» гра автоматично закривається і перекидає гравця на робочий стіл. При натисканні кнопки «Почати гру» відкривається меню в якому представлено комірки для створення збереження, . Після обрання комірки починається гра, і всі подальші збереження будуть зберігатися в обраній комірці;
- HUD персонажу: під час гри гравець зможе відслідковувати кількість здоров’я, валюти персонажу;
- персонаж: гравець може контролювати персонажа, який володіє основними механіками: біг, стрибок, атака. Усі дії та стан бездіяльності супроводжуються анімаціями. Також він може взаємодіяти з: другорядними персонажами для активації діалогу та об’єктами локації, які потрібні для вирішення загадок або для збереження гри. Атакуючи ворога герой наносить йому шкоду.

На рис. 2.2 представлено діаграму прецедентів «Система збереження та відродження». За допомогою персонажу, гравець може взаємодія з об’єктом збереження, при його активації починається процес збереження гри. Після смерті персонаж відроджується на останній точці збереження з якою взаємодіяв.

На рис. 2.3 представлено діаграму активності «Зміна анімацій персонажу».



Умовні позначення:

– — актор «Гравець», який взаємодіє з системою;

– — основні функції системи;

– — пов'язує актора з функцією, з якою взаємодіє;

– — розширення основних функцій системи;

– — область дії системи.

Рисунок 2.2 Діаграма прецедентів «Система збереження та відродження»

Джерело: розроблено автором

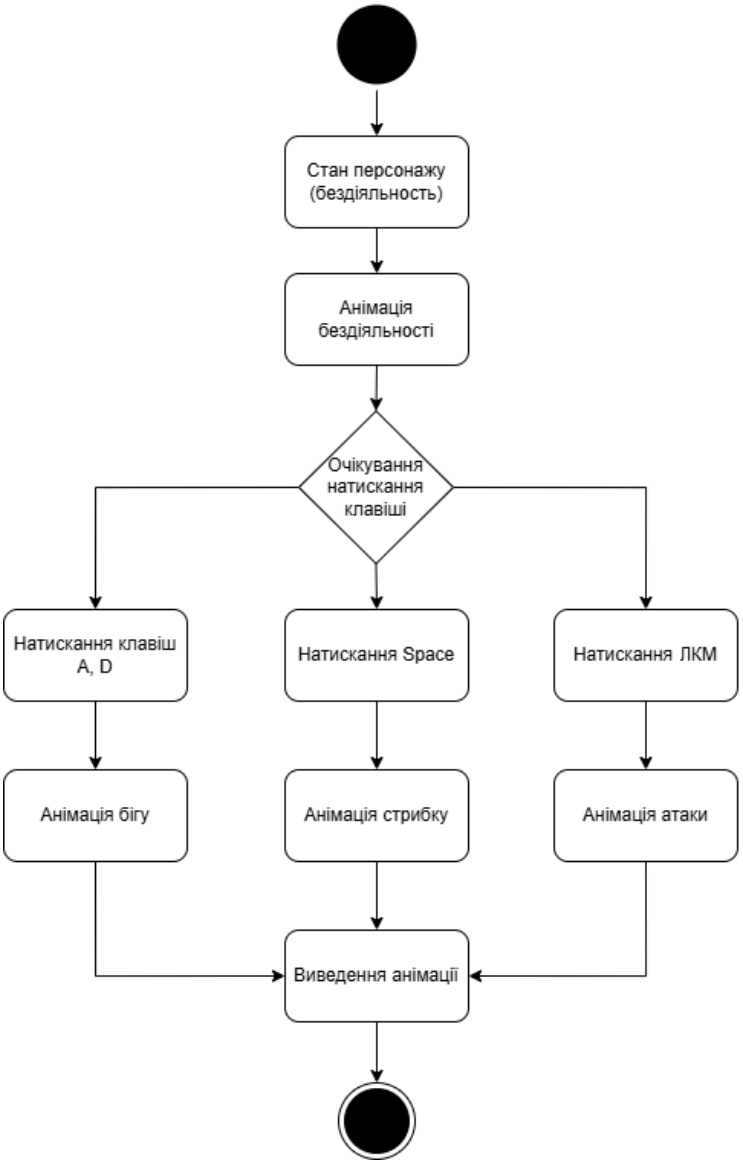


Рисунок 2.3 —Діаграма активності «Зміна анімацій персонажу»

Джерело: розроблено автором

Умовні позначення (рис. 2.3):

– ● — початок набору дій або операцій;

– ▭ — позначення набору дій або операцій;

– ◇ — позначення умови перевірки;

–  — вказує послідовність виконання дій або операцій;

–  — завершення усіх потоків керування та потоків об’єктів в дії або операції.

Персонаж починає зі стану бездіяльності, дія супроводжується анімацією. Гравець взаємодіє з клавішами A, D, кнопкою Space та правою кнопку миші, після чого починається процес переходу з анімації бездіяльності в анімацію бігу, стрибку та атаки відповідно. Потім анімація відтворюється на екрані гравця.

На рис. 2.4 представлено діаграму активності «Система збереження гри».

Рисунок 2.4 —Діаграма активності «Система збереження гри»

Джерело: розроблено автором

Умовні позначення (рис. 2.4):

–  — початок набору дій або операцій;

–  — позначення набору дій або операцій;

–  — позначення умови перевірки;

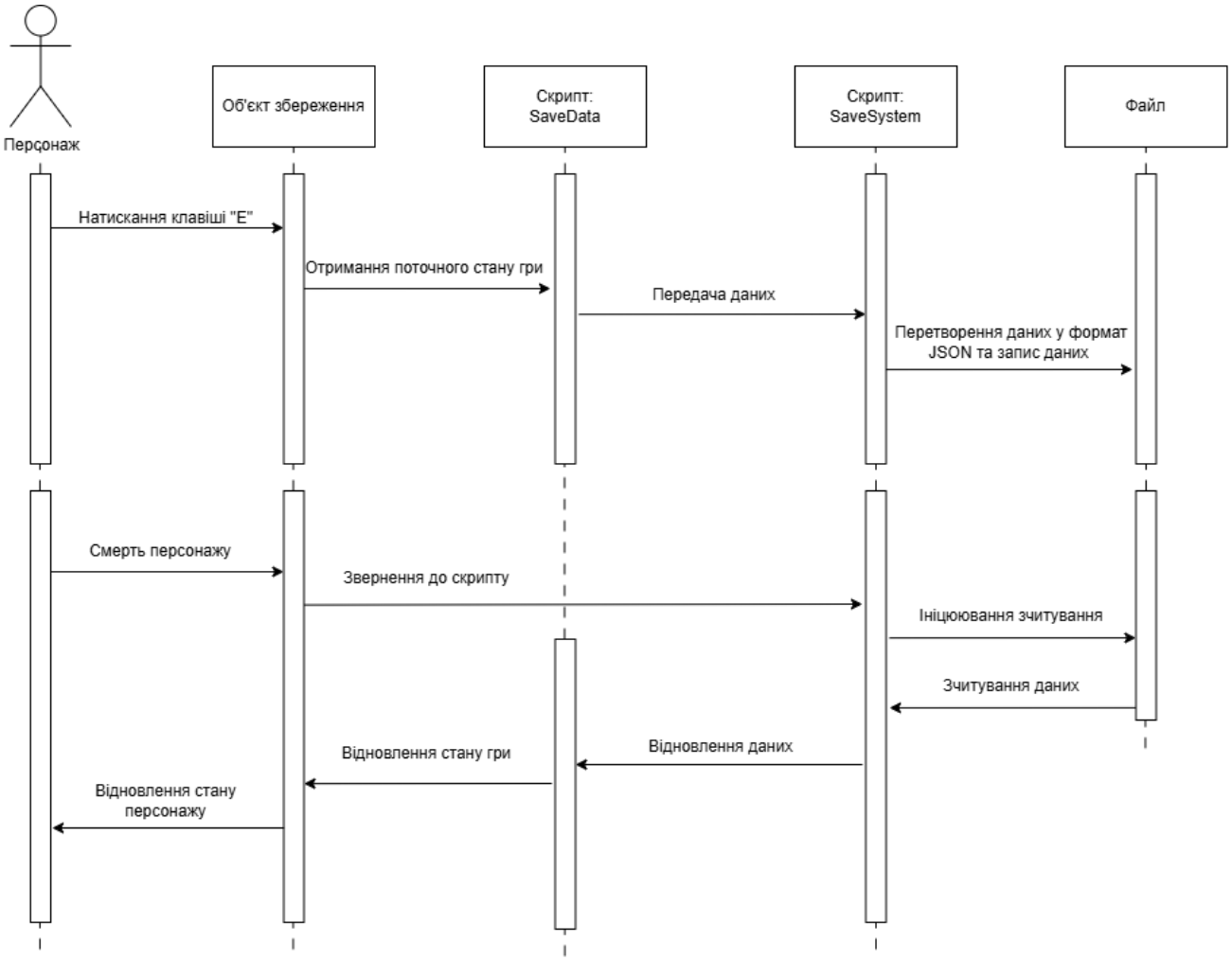
–  — вказує послідовність виконання дій або операцій;

–  — завершення усіх потоків керування та потоків об’єктів в дії або операції.

Пояснення діаграми. Гравець, керуючи персонажем, заходить у зону об’єкту, він обирає, чи пройти повз, чи взаємодіяти з ним. При взаємодії відбувається збереження стану персонажу, об’єкту та світу. Далі отриманні дані перетворюються у формат JSON, після чого дані записуються у файл.

На рис. 2.5 представлено діаграму послідовності.

На рис. 2.6 представлено діаграму комунікації. Персонаж взаємодіє з об’єктом збереження, через натискання клавіші «E», після чого починається передача стану гри у скрип SaveData, у якому записані усі необхідні елементи, які потрібно зберегти. Інформація передається у скрипт SaveSystem де відбувається перетворення даних у формат JSON. Далі перетворені дані записуються у файл. Після смерті персонажу йде звернення до скрипту SaveSystem, далі ініціювання зчитування з файлу, процес зчитування даних та подальша передача до скрипту SaveData, відновлення стану гри, відродження персонажу на останньому об’єкті збереження з яким він взаємодіяв.



Умовні позначення:

– — актор, який використовує систему;

– — елемент послідовності, що надсилає та/або отримує повідомлення;

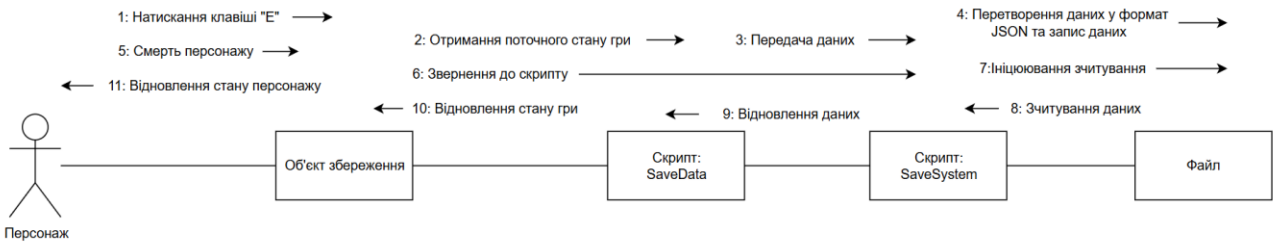
– — позначає коли об’єкт надсилає або отримує повідомлення;

– — позначає тривалість життя об’єкту в послідовності;

– — передає інформацію від одного об’єкту до іншого.

Рисунок 2.5 —Діаграма послідовності «Система збереження та відродження»

Джерело: розроблено автором



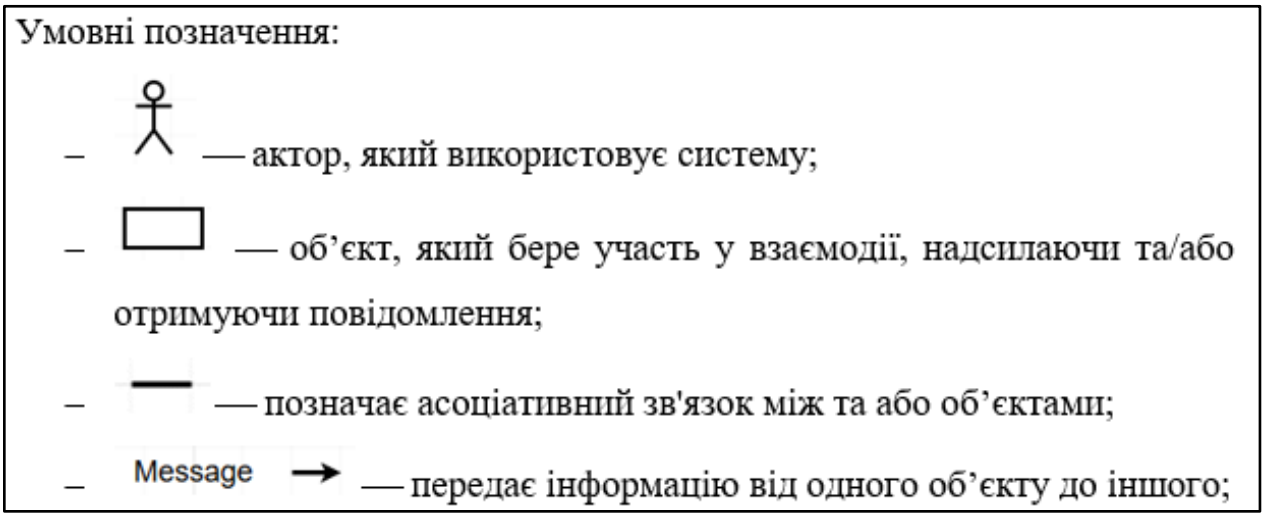


Рисунок 2.6 —Діаграма комунікації «Система збереження та відродження»

Джерело: розроблено автором

2.2 Моделювання структури продукту

На рис. 2.7 представлено діаграму класів, які використовуються для системи зберігання.

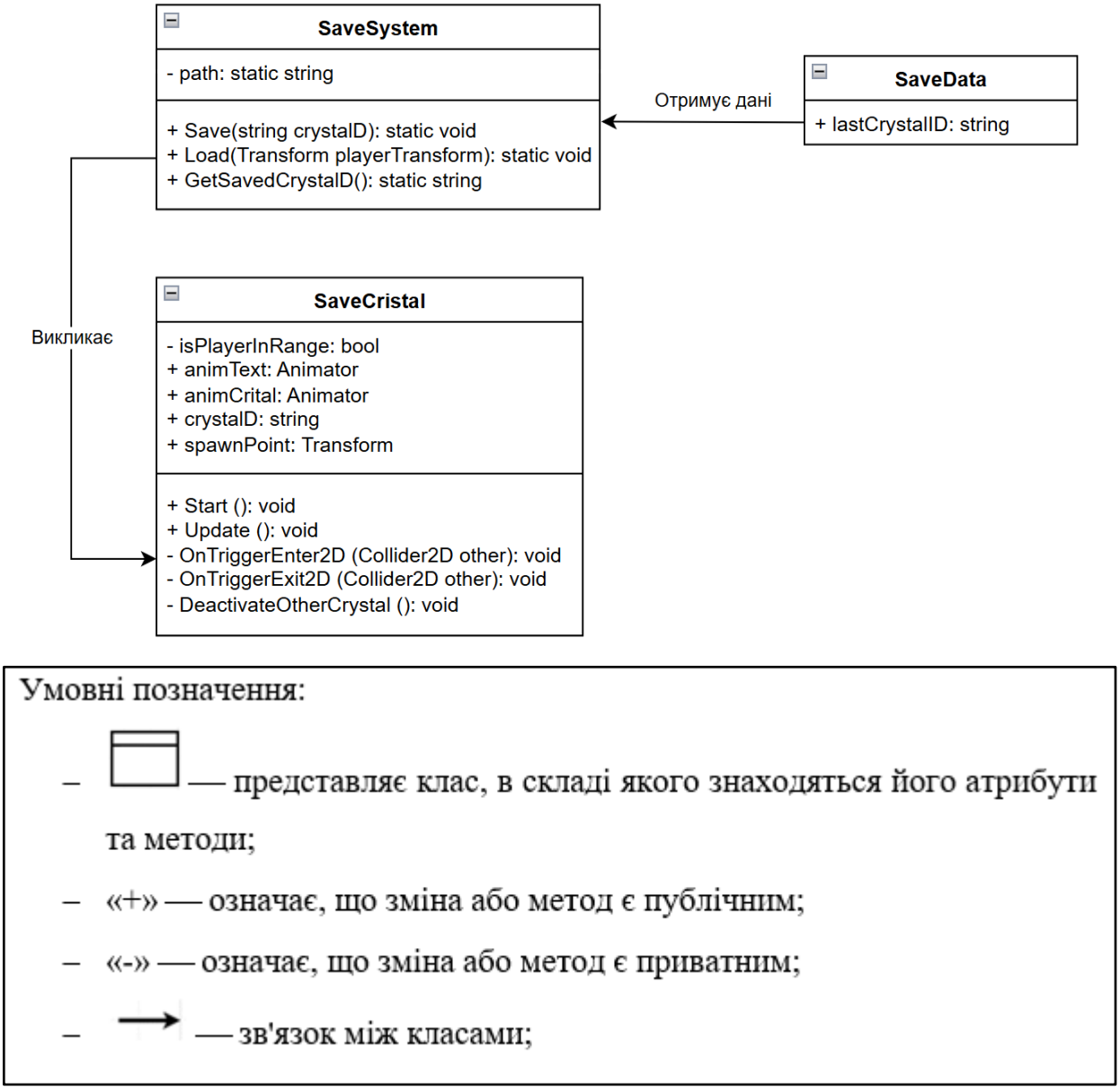


Рисунок 2.7 —Діаграма класів

Джерело: розроблено автором

Пояснення діаграми. Система збереження складається з трьох скриптів SaveData, SaveSystem та SaveCristal. Кожен з них виконує певну роль:

– SaveData відіграє роль сховища, де зберігаються потрібні дані, які необхідно зберегти.

- SaveSystem керує процесом збереження та завантаження гри. Клас відповідає за перетворення даних, отриманих з SaveData, у формат JSON та запису їх у файл, а також за їх зчитування.
- SaveCristal клас, дає можливість гравцю взаємодіяти з кристалами, тим самим викликати метод збереження гри.

2.3 Опис архітектури продукту

2.3.1 Опис просторів імен

При розробці певних механік було використано наступні простори імен.

System — простір імен, який містить базові типи даних, що використовуються у додатках написані мовою C#. В його склад входять класи для роботи з різними типами даних, керування пам’яттю, обробка виключень, тощо.

System.IO — простір імен, що використовується у операціях введення або виведення. В його склад входять класи, що дають можливість працювати з файлами, потоками даних та каталогами. У проекті використовується для збереження та завантаження даних форматі JSON у файлі, що дає можливість реалізувати систему збереження та відродження.

UnityEngine — основний простір імен, який використовується для розробки ігор в Unity. В його склад входять класи, що дозволяють працювати з основними функціями ігрового рушія (управління об’єктами, анімацією, фізикою, компонентами сцени тощо).

UnityEngine.UI — простір імен, який спеціалізується на створені та управлінні інтерфейсу користувача. В його склад входять функції, які допомагають створювати і управляти елементами інтерфейсу (текстові поля, кнопки, полотно, тощо).

2.3.2 Опис класів та компонентів

PlayerController — MonoBehaviour клас, який відповідає за управління персонажем та зміну його анімації.

PlayerAttack — MonoBehaviour клас, який відповідає за управління атакою персонажу та відтворення її анімації.

SaveCristal — MonoBehaviour клас, який відповідає за збереження гри, шляхом взаємодії персонажу з об’єктом. Також у ньому розписані відтворення анімацій об’єкту та тексту.

SaveData — серіалізований клас, який відповідає за збереження даних про персонажа, об’єкту та світу.

SaveSystem — статичний клас, який відповідає за збереження та завантаження гри. Отримані дані, клас перетворює у формат JSON та записує у файл.

MainMenu — MonoBehaviour клас, який відповідає за кнопки головного меню.

Класи та компоненти Unity:

MonoBehaviour — базовий клас, який дає доступ до життєвого циклу.

Rigidbody2D — компонент, який додає до 2D об’єктів фізичні властивості.

SpriteRenderer — компонент, який відображає 2D спрайти на екрані.

Animator — компонент, який керує анімаціями через контролер. Дає можливість запускати, зупиняти та перемикати анімації.

Transform — компонент, який визначає координати, поворот, масштаб об’єкту на сцені.

Input — клас, який обробляє введення з клавіатури та миші.

SceneManager — клас, що дозволяє керувати сценами (завантаження, перезавантаження, перемикання сцен в грі, тощо).

Application — клас, який надає доступ до даних виконання та дозволяє виконувати дії.

Debug — клас, який використовується для виведення повідомлень у консоль Unity, що допомагає відслідковувати дії.

JsonUtility — клас, який дозволяє перетворювати та читати дані формату JSON.

2.3.3 Опис методів

Start (), Update () — базові методи Unity життєвого циклу скрипта.

OnTriggerEnter2D(Collider2D other) — метод, який викликається коли персонаж покидає зону об’єкту збереження. Активує анімацію появи тексту об’єкту.

OnTriggerExit2D(Collider2D other) — метод, який викликається коли персонаж входить в зону об’єкту збереження. Активує анімацію затухання тексту об’єкту.

DeactivateOtherCrystal () — метод, який скидає стан активації кристалу та встановлює анімацію у стан «idle».

Save(string crystalID) — метод, який перетворює отримані дані у формат JSON та записує їх у файл.

Load(Transform playerTransfrom) — метод, який зчитує збережені дані з файлу та переміщує персонажа.

GetSavedCrystalID() — метод, що надає інформацію щодо останнього збереженого ідентифікатора кристалу, що дозволяє зрозуміти, який саме кристал є активним після перезапуску гри.

Висновки до розділу 2

Визначено функціональні та нефункціональні вимоги гри. За допомогою діаграм прецедентів описано функціональність гри та систему збереження. Для розуміння роботи зміни анімацій персонажу та системи збереження гри, побудовано діаграми активності, послідовності та комунікації.

Для показу зв’язків між класами, які використовуються у роботі системи збереження, побудовано діаграму класів.

Описано простори імен, класи та методи, які використовувалися при розробці.

РОЗДІЛ 3.
РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Реалізація та конструювання програмного продукту

3.1.1 Опис ігрових рушіїв

Unreal Engine (рис. 3.1) — це один з найпопулярніших ігрових рушіїв, який розроблений та підтримується компанією Epic Games. Двигун створений для розробки ігор різних жанрів, у більшості випадків 3D проектів. Unreal Engine використовує мову C++, а також має власну систему візуального програмування Blueprint.

Переваги [6]:

- підтримка сучасних технологій;
- Blueprint;
- висока якість графіки та візуальних ефектів;
- доступ до вихідного коду;
- підтримка розробників.

Недоліки [6]:

- високі системні вимоги;
- мала кількість функцій у розробці 2D ігор;
- призначення для розробки високо бюджетних ігор;
- погана стабільність роботи.

Рисунок 3.1 – Unreal Engine

Джерело [7]

Godot Engine (рис. 3.2) — безкоштовний та відкритий багатоплатформовий ігровий рушій, який розробляється компанією Godot Engine Community. Двигун дає можливість розробляти як 2D, так і 3D ігри. Для розробки гри рушій використовує різні мови програмування C# C++, а також власну GDSript (схожу на Python).

Переваги:

- відкритість та безкоштовність;
- кросплатформеність;
- простота використання;

Недоліки:

- менша підтримка графіки;
- не достатня кількість функцій у розробці 3D.

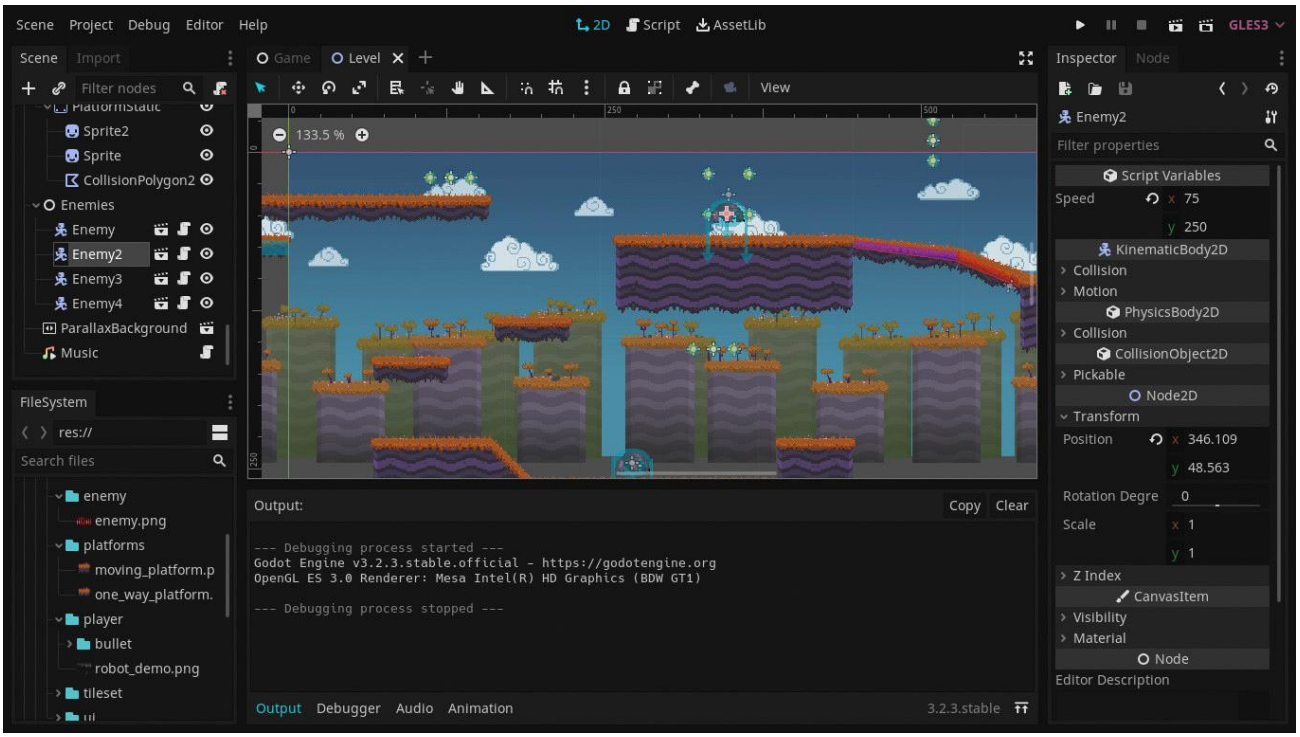


Рисунок 3.2 —Godot Engine

Джерело [8]

Unity (рис. 3.3) – це ігровий рушій, який розробляється компанією Unity Technologies. Двигун дає змогу працювати як у 2D просторі, так і у 3D. Мову, яку використовує рушій це C#.

Переваги:

- зрозумілий інтерфейс;
- магазин безкоштовних та платних ресурсів;
- мова програмування.

Недоліки

- оптимізація та продуктивність;
- проблеми з масштабністю.

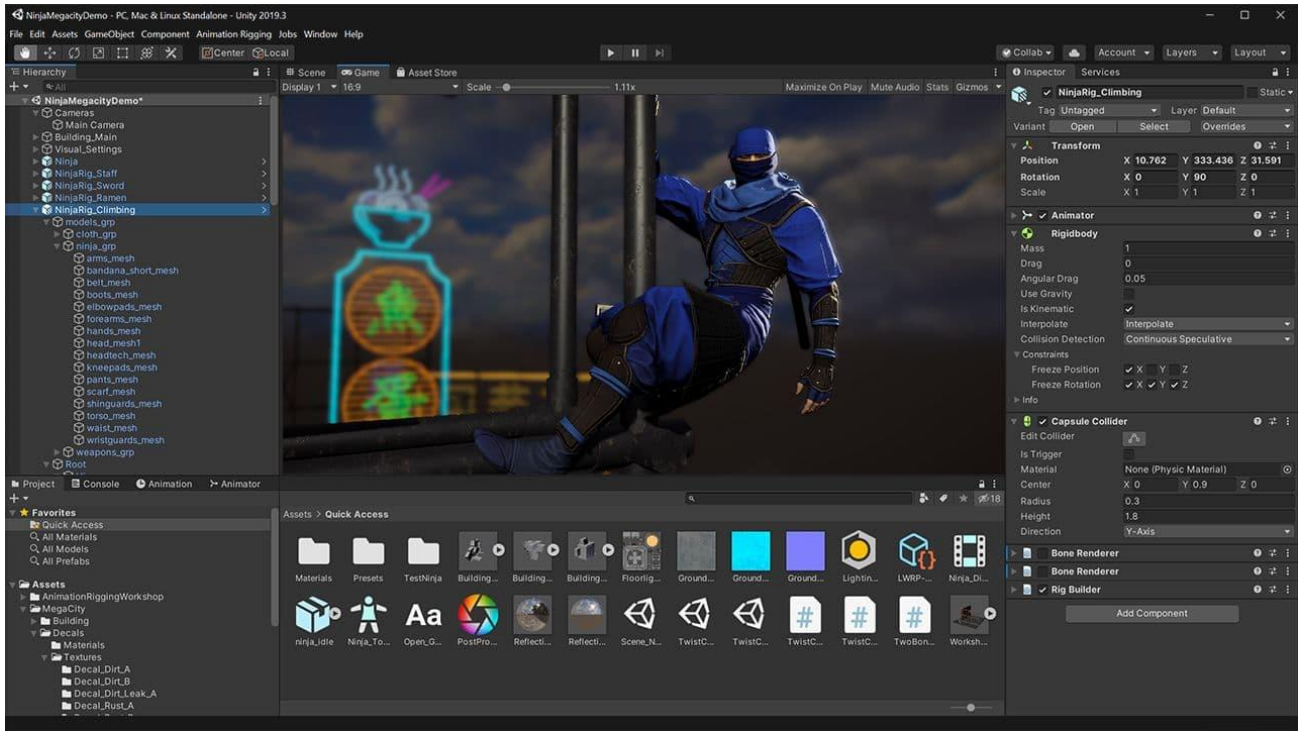


Рисунок 3.3 — Unity

Джерело [9]

3.1.2 Опис графічних програм

GraphicsGale (рис. 3.4) — графічна програма, яка орієнтована на створення піксель артів та анімації.

Переваги:

- безкоштовна версія;
- зручний інтерфейс.

Недоліки:

- застарілість;
- обмежений набір функцій.

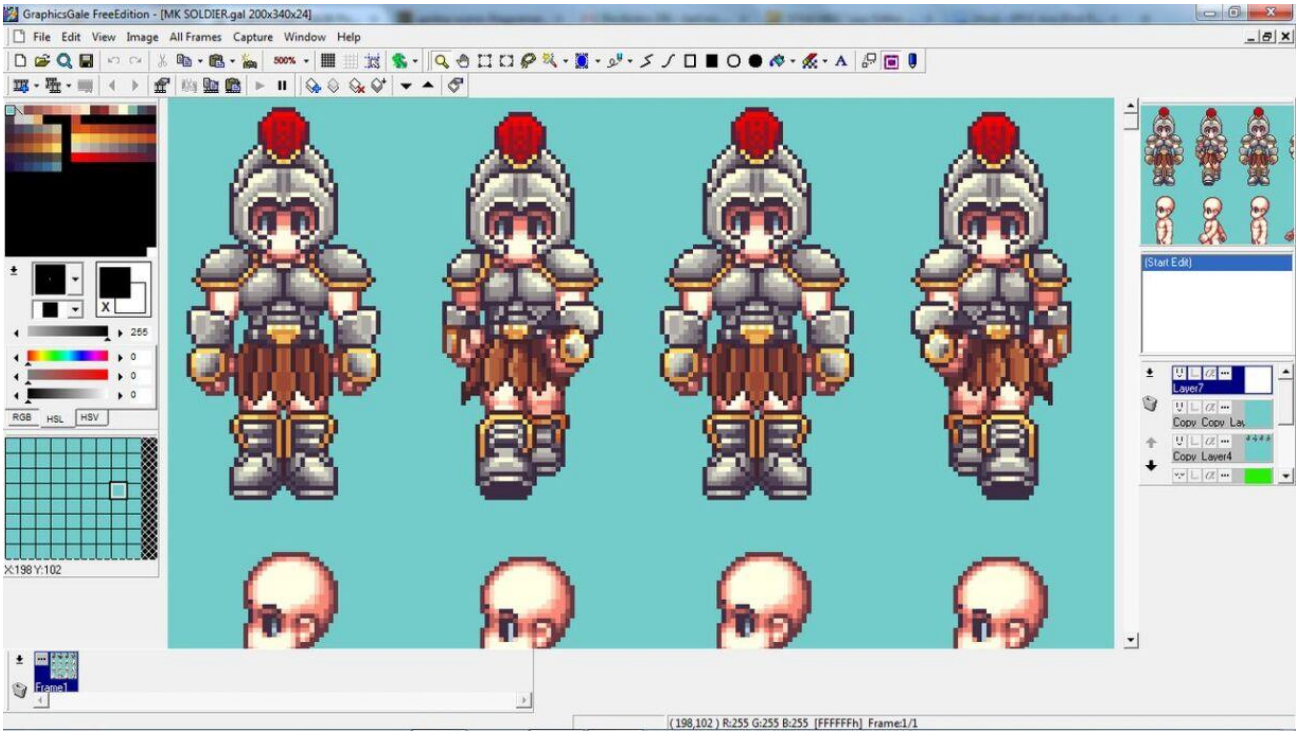


Рисунок 3.4 —GraphicsGale

Джерело [10]

Pyxel Edit (рис. 3.5) — графічна програма, розроблена для створення піксель артів, анімацій, а також тайлсетів.

Переваги:

- робота з сіткою, що спрощує створення тайлсетів;
- простий інтерфейс.

Недоліки:

- обмежені функції для складної анімації.

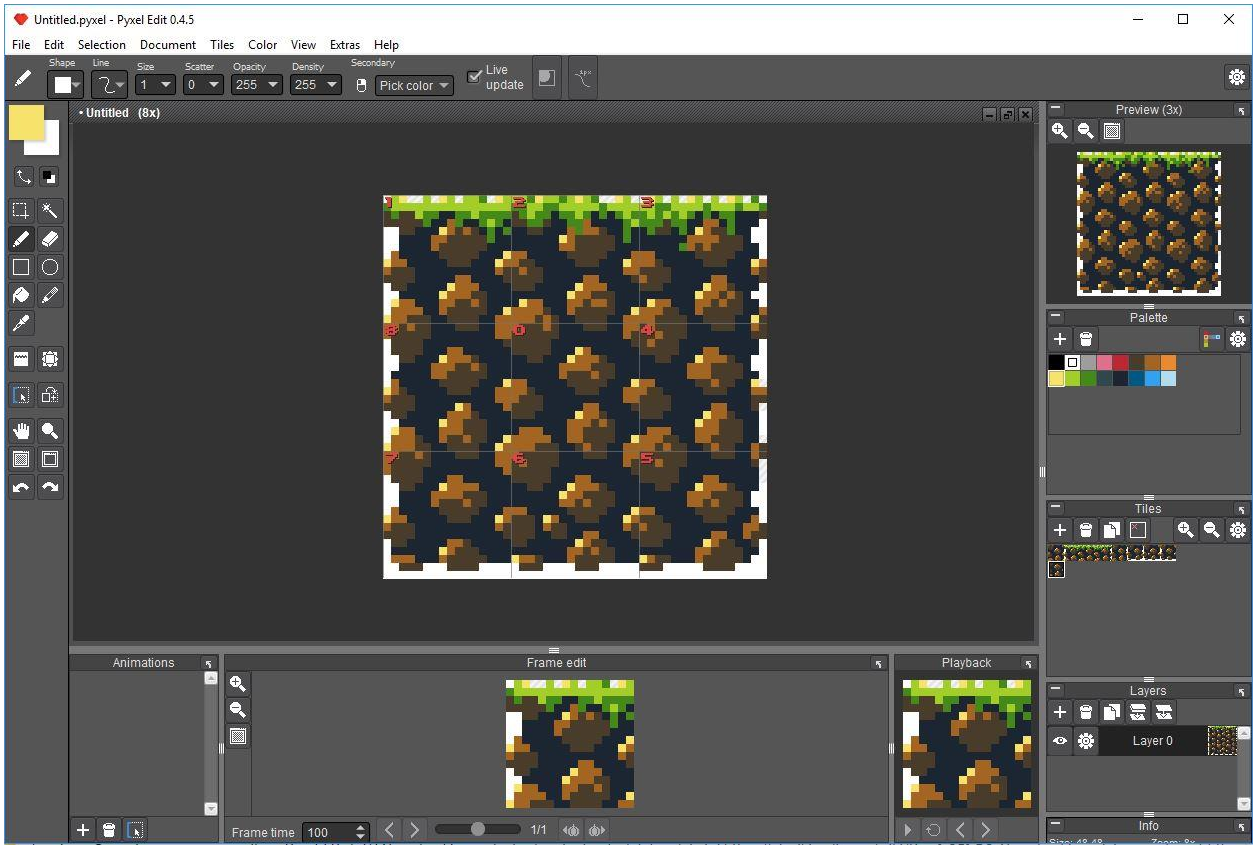


Рисунок 3.5 — Pyxel Edit

Джерело [11]

Aseprite (рис. 3.6) — графічна програма, яка розроблена для створення 2D-анімації, спрайтів і будь-якої графіки для ігор.

Переваги:

- інтерфейс, який орієнтований на піксель арт;
- функція перегляду попереднього кадру анімації;

Недоліки:

- відсутня підтримка робіт з векторною графікою.

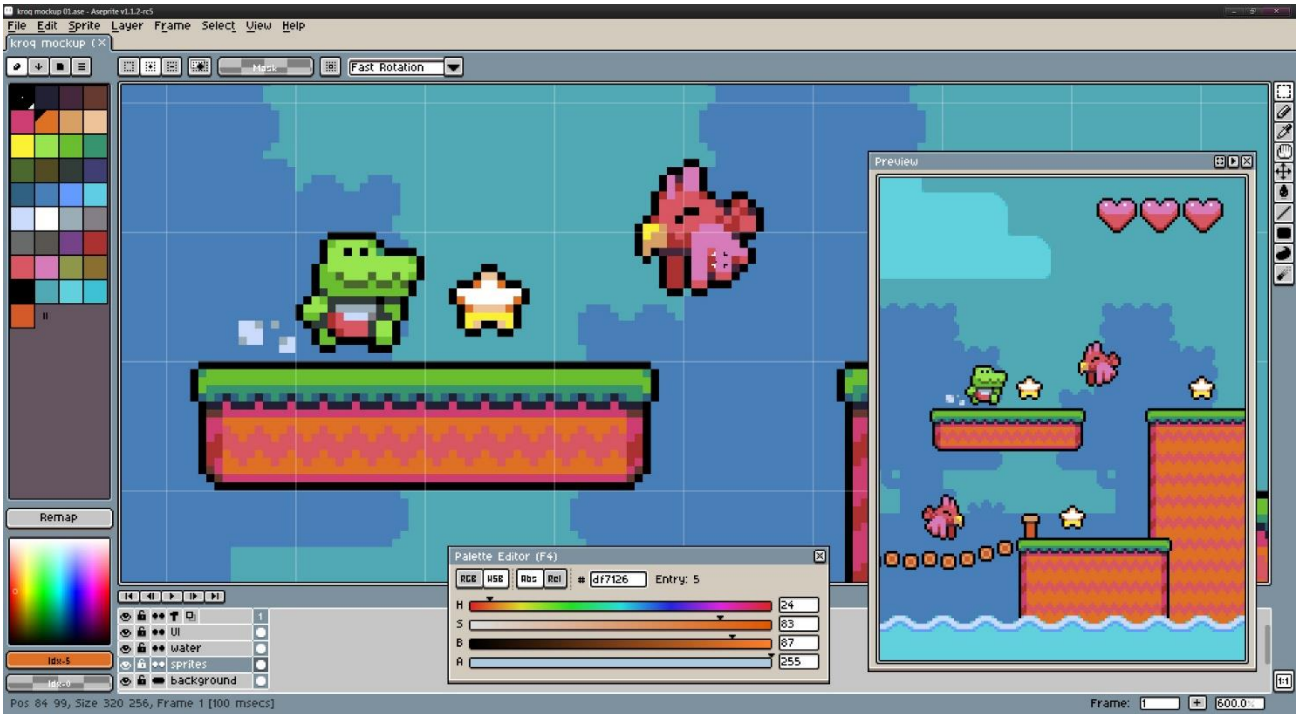


Рисунок 3.6 —Aseprite

Джерело [12]

3.1.3 Unity та Aseprite

Після розгляду варіантів ігрових рушіїв, для розробки програмного продукту було обрано ігровий рушій Unity.

Причини вибору:

- зручний інтерфейс та легкість в освоєні;
- великий інструментарій для розробки 2D ігор;
- більшість 2D платформерів розроблялися саме на двигуні Unity;
- невеликий досвід отриманий у попередніх проектах.

А для створення візуального стилю гри та анімацій було обрано програму Aseprite.

Причини вибору:

- робота з піксельною графікою;

- зручний інтерфейс;
- інструменти для роботи з анімаціями;
- налаштування палітр;
- можливість експорту анімацій.

Microsoft Visual Studio – це середовище розробки, яке використовується для створення комп'ютерних програм, додатків, тощо. Програма підтримує велику кількість мов програмування, наприклад, C#, C++, Python, JavaScript, тощо.

Причини вибору:

- допомога у написані коду (автодоповнення коду, підказки, підсвічування помилок та варіанти вирішення);
- інтеграція Visual Studio for Unity, що дозволяє налагоджувати проект безпосередньо з середовища.

3.2 Тестування програмного продукту

Проведено альфа тестування. Було перевірено правильність роботи основних аспектів:

- генерація локацій;
- нанесення та отримання шкоди;
- інтелектуальна поведінка ботів;
- система збереження та відродження;
- відродження ворогів.

За результатами тестування критичних помилок, які заважали б проходженню гри або некоректної роботи механік не зафіксовано.

3.3 Використання програного продукту

Щоб розпочати гру в головному меню (рис. 3.7) потрібно натиснути кнопку «Почати гру».

Рисунок 3.7 – Головне меню

Джерело: розроблено автором

Управління персонажем складається з: біг (рис. 3.8) – для активації потрібно натиснути клавішу A або D, стрибок (рис. 3.9) – натиснути клавішу Space, атака (рис. 3.10) – натиснути ліву кнопку миші.

Рисунок 3.8 —Анімація бігу

Джерело: зроблено автором

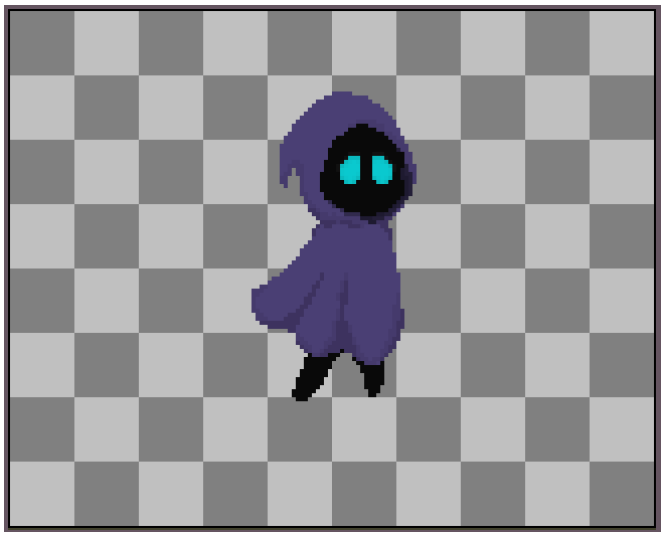


Рисунок 3.9 —Анімація стрибку

Джерело: зроблено автором

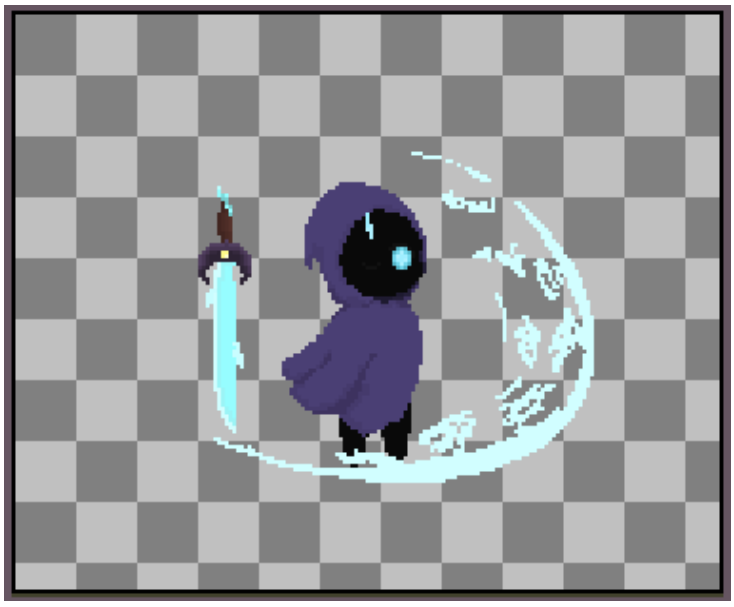


Рисунок 3.10 – Анімація атаки

Джерело: зроблено автором

Замість здоров’я персонаж має шкалу пошкодження (рис. 3.11), вона складається з певної кількості бар’єрів (червоний) та ядра (зелений). При отримані шкоди буде знищуватися один бар’єр. Якщо герой втратить їх усіх у нього залишиться тільки ядро, при його втраті він помирає. Для відновлення втрачених бар’єрів гравцю потрібно взаємодіяти з кристалами відродження (рис. 3.13).

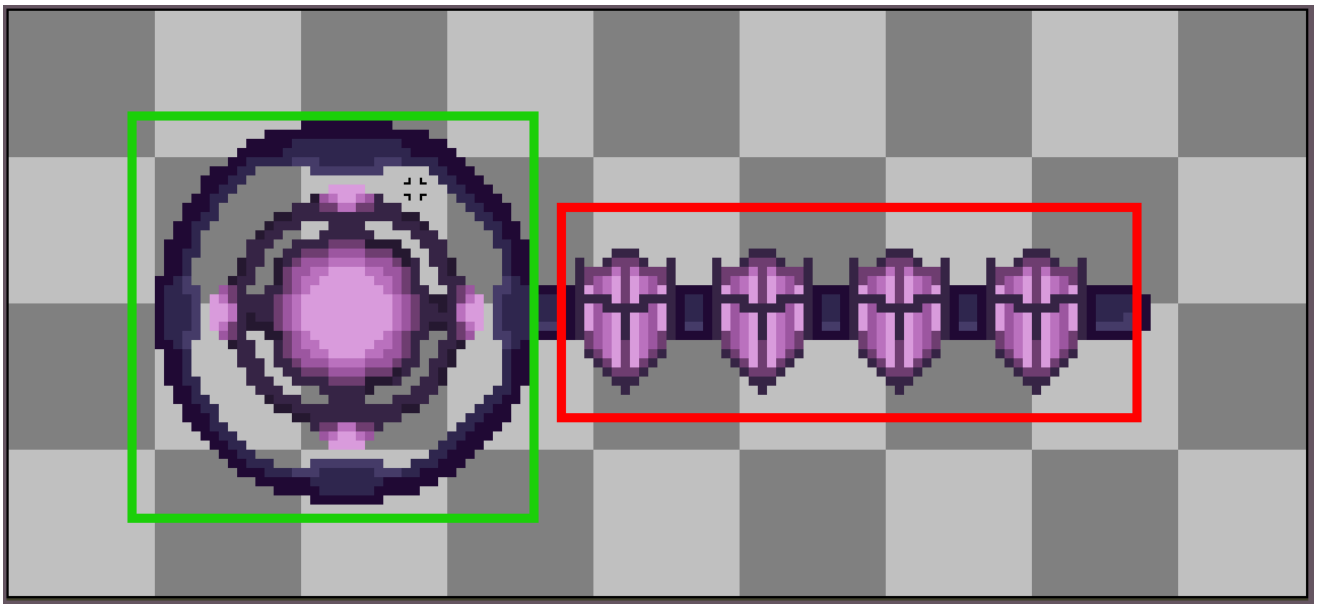


Рисунок 3.11 – Шкала пошкодження персонажу

Джерело: розроблено автором

При дослідженні локації гравець буде зустрічати об’єкти або предмети з якими персонаж може взаємодіяти. Для цього потрібно підійти до об’єкта поки не з’явиться напис, після чого натиснути клавішу E.

Також на локаціях можна знайти декілька негральних персонажів (рис. 3.12), з якими гравець зможе поговорити за допомогою клавіші E.

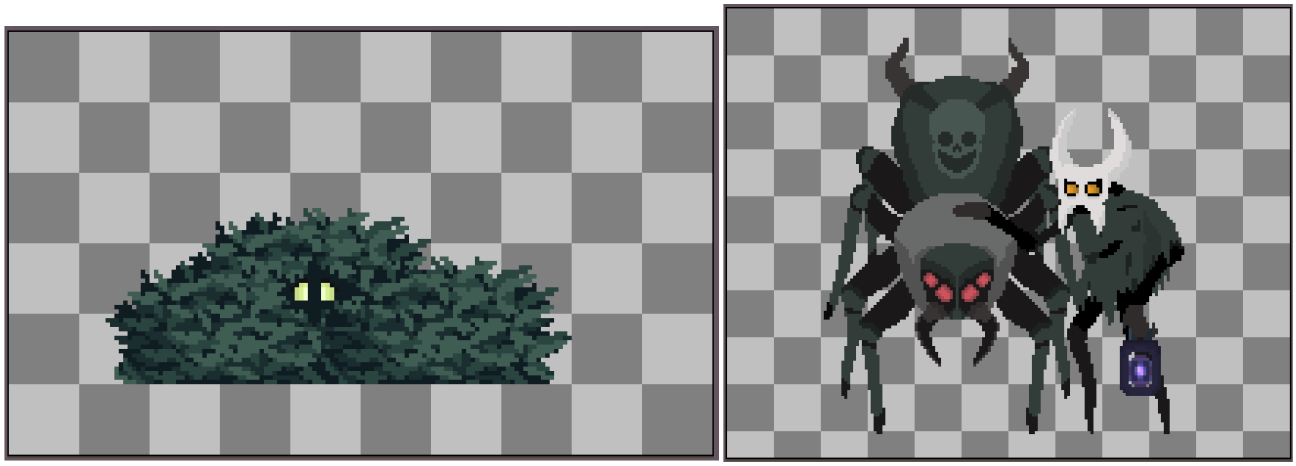


Рисунок 3.12 – Неігральні персонажі

Джерело: розроблено автором

Локації будуть населені противниками (рис. 3.13). При входженні в поле зору ворога він почне атакувати (рис. 3.14), якщо гравець почне тікати, противник буде його переслідувати до певного моменту. Відродження вбитих противників відбувається коли гравець взаємодіяє з кристалом (рис. 3.15).

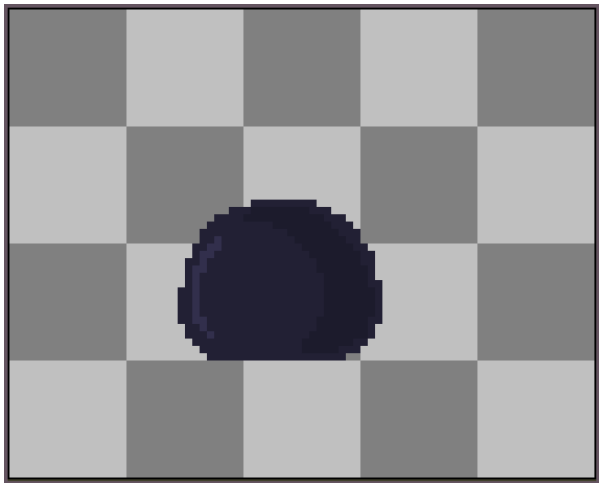


Рисунок 3.13 – Противник

Джерело: зроблено автором

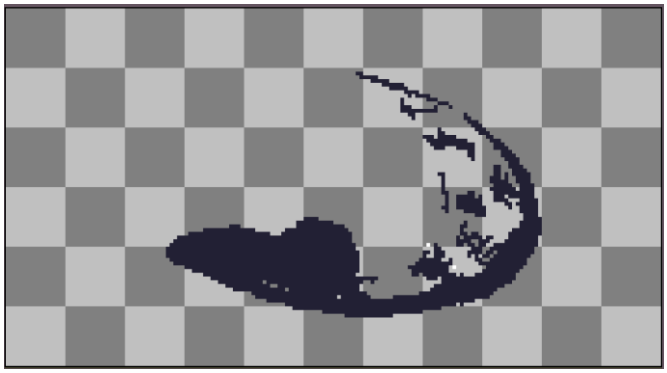


Рисунок 3.14 – Анімація атаки

Джерело: зроблено автором

У грі система збереження і відродження працюють наступним чином: на локаціях розташовані так звані кристали (рис. 3.15), при взаємодії з ними гра буде автоматично зберігатися, вбиті вороги відроджуватися, а персонаж отримає точку відродження. Після смерті персонаж відродиться на кристалі з яким взаємодіям останній раз.

Рисунок 3.15 – Кристали збереження (ліворуч активний, праворуч неактивний)

Джерело: зроблено автором

На останній локації гравець зустрінеться з фінальним босом «Божевільний послідовник» (рис. 3.16). Він має дві фази. В першій атакує простими атаками. Коли персонаж наносить достатньо шкоди босс переходить у другу фазу атаки стають швидше, а також додається декілька нових прийомів. Після вбивства його гра завершується.

Рисунок 3.16 — Божевільний послідовник

Джерело: розроблено автором

Висновки до розділу 3

Було проведено аналіз існуючих ігрових рушіїв та графічних програм, визначено їх переваг та недоліків. Обґрунтовано причини вибору Unity, Aseprite та Microsoft Visual Studio.

Проведено альфа тестування програмного продукту. Перевірка роботи основних механік.

Описано можливості програмного продукту та створено інструкцію користувача.

ВИСНОВКИ

Досліджено призначення комп’ютерних ігор. Аналіз основних жанрів та їх піджанрів. Розглянуто аналоги програмного продукту, визначено їх переваги та недоліки. Розроблено правила гри та описано автоматизовані функції, які використовувалися у розробці.

Визначено функціональні та нефункціональні вимоги продукту. Побудовано діаграми прецедентів для функціоналу гри та системи збереження. Для розуміння механізму зміни анімацій та роботи системи збереження було побудовано діаграми активності, послідовності та комунікації. Описано використані простори імен, компоненти, класи та методи.

Проаналізовано варіанти ігрових рушіїв та графічних програм, які розглядалися як можливі платформи для розробки гри, їх переваги та недоліки. За результатами аналізу приведено причини вибору відповідних програм. Проведено альфа тестування для виявлення критичних помилок продукту. Для розуміння можливостей гри, описано всі її аспекти, механіки тощо. Створено інструкцію для користувачів.

Розроблено 2D комп’ютерну гру жанру платформер з інтерфейсом, HUD персонажу, нанесенням шкоди ворогам, кат сценами, системою збереження гри, діалогами з другорядними персонажами, інтелектуальною поведінкою ботів та генерацією локацій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Celeste // Wikipedia URL: https://uk.wikipedia.org/wiki/Celeste_(%D0%B2%D1%96%D0%B4%D0%B5%D0%BE%D0%B3%D1%80%D0%B0)

2. Celeste // Steam URL: https://store.steampowered.com/app/504230/Celeste/

3. Hollow Knight // Steam URL: https://store.steampowered.com/app/367520/Hollow_Knight/

4. Ori and the Blind Forest // Steam URL: https://store.steampowered.com/app/261570/Ori_and_the_Blind_Forest/

5 Cuphead // Steam URL: https://store.steampowered.com/app/268910/Cuphead/

6. Unreal Engine: Переваги та основні можливості двигуна. // Avada media URL: https://avada-media.ua/services/unreal-engine-preimushchestva-i-osnovnyye-vozmozhnosti-igrovogo-dvizhka/

7. Unreal Engine // Unreal Engine URL: https://www.unrealengine.com/en-US

8. Godot Engine // Godot Engine URL: https://godotengine.org/

9. Unity // Unity URL: https://unity.com/

10. Graphics Gale // Graphics Gale URL: https://graphicsgale.com/us/

11. Pyxel Edit // Pyxel Edit URL: https://pyxeledit.com/

12. Aseprite // Steam URL: https://store.steampowered.com/app/431730/Aseprite/

13. Make Your MAIN MENU Quickly! | Unity UI Tutorial For Beginners // YouTube URL: https://www.youtube.com/watch?v=DX7HyN7oJjE

14. The Fastest Way to Create Trees & Bushes for Beginners (Aseprite Tutorial) // YouTube URL: https://www.youtube.com/watch?v=l0kzBNEOJTU

15. Мічківський С. Microsoft Office (Word, Excel, Outlook …) : навч. посіб. / С. Мічківський, Д. Балдик, В. Головань; Східноукр. нац. ун-т ім. В. Даля, Аграр.

ф-т. – Київ : [Вид-во Східноукр. нац. ун-т ім. В. Даля], 2023. – 128 с. – URL: https://dspace.snu.edu.ua/handle/123456789/1723

16. Vysochyn I., Michkivskyy S. Using machine learning for translation and speech generation in e-book reading applications. Держава, регіони, підприємництво: інформаційні, суспільно-правові, соціально-економічні аспекти розвитку: матеріали VI Міжнародної наукової конференції (5-6 грудня 2024 р., м. Київ). Київ: Університет "КРОК", 2024. С.62-64 – URL: https://dspace.krok.edu.ua/items/6e1488e1-9e21-4282-af10-2e3bca48d2bc