

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук (або центр післядипломної освіти, або навчально-науковий центр заочної форми навчання)
(повна назва)

Кафедра _____ програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Ігровий програмний застосунок у жанрі Metroidvania з елементами карткової гри.
User Interface. Анімація. Графічне відображення
(тема)

Виконав:
студент 4 курсу, групи ПЗП-20-1

_____ Лухменко Е.К.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна

Освітня програма Програмна інженерія
(повна назва освітньої програми)

Керівник ст.викл. кафедри ПІ Саманцов О.О.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри

_____ З.В.Дудар
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
 Кафедра _____ програмної інженерії _____
 Рівень вищої освіти _____ перший (бакалаврський) _____
 Спеціальність _____ 121 – Інженерія програмного забезпечення _____
 Тип програми _____ Освітньо-професійна _____
 Освітня програма _____ Програмна Інженерія _____
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«___» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Лухменко Ельвірі Костянтинівні _____
 (прізвище, ім'я, по батькові)

1. Тема роботи _____ Ігровий програмний застосунок в жанрі Metroidvania з елементами карткової гри. User Interface. Анімація. Графічне відображення
 Затверджена наказом по університету від _____ 20.05. 2024р. № 471 Ст _____
2. Термін подання студентом роботи до екзаменаційної комісії _____ 12.06.2024 _____
3. Вихідні дані до роботи Розробити ігровий застосунок в жанрі Metroidvania з елементами карткової гри, який може взаємодіяти з рівнями (локаціями), ворожими персонажами, NPC, міні-грою та колекційними картками, мовою програмування JavaScript використовуючи ігровий двигун St.js, програмою для створення ілюстрацій Krita та програмою для створення анімацій Spine. _____
4. Перелік питань, що потрібно опрацювати в роботі
Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, впровадження розробленого програмного забезпечення, висновки, додатки. _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	20.04.2024	<i>виконано</i>
2	Створення специфікації ПЗ	28.05.2024	<i>виконано</i>
3	Проектування ПЗ	05.05.2024	<i>виконано</i>
4	Розробка ПЗ	12.05.2024	<i>виконано</i>
5	Тестування ПЗ	30.05.2024	<i>виконано</i>
6	Оформлення пояснювальної записки	02.06.2024	<i>виконано</i>
7	Підготовка презентації та доповіді	07.06.2024	<i>виконано</i>
8	Попередній захист	07.06.2024	<i>виконано</i>
9	Нормоконтроль, рецензування	07.06.2024	<i>виконано</i>
10	Здача роботи у електронний архів	07.06.2024	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	12.06.2024	<i>виконано</i>

Дата видачі завдання 18 грудня 2023р.

Студент (ка) _____
(підпис)

Лухменко Е.К.

Керівник роботи _____
(підпис)

ст.викл. кафедри ПІ Саманцов О.О.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра, 87 стор., 41 рис., 4 табл., 21 джерел.

ГЕЙМПЛЕЙ, 2Д ГРАФІКА, ДІАГРАМА ДІЯЛЬНОСТІ, ДІАГРАМА ПАКЕТІВ, ДІАГРАМА ПРЕЦЕДЕНТІВ, ДІАГРАМА СТАНУ, ІГРОВИЙ ПРОГРАМНИЙ ЗАСТОСУНОК, КОЛЕКЦІЙНА КАРТКОВА ГРА, СКЕЛЕТНА АНІМАЦІЯ, СТ JS, JAVASCRIPT, KRITA, METROIDVANIA, SPINE, UML

Об'єкт розробки – ігровий програмний застосунок у жанрі Metroidvania з елементами карткової гри.

Мета розробки – створення ігрового програмного застосунку, який буде містити 6 рівнів та міні-гру у жанрі Metroidvania з елементами карткової колекційної гри.

Метод рішення – середовище розробки St.js, програма для створення ілюстрацій та спрайтів Krita, програма для створення анімацій Spine, мова програмування JavaScript.

У результаті розробки було створено ігровий програмний застосунок з головним героєм, який може взаємодіяти з рівнями (локаціями), ворожими персонажами, NPC, міні-грою та колекційними картками.

GAMEPLAY, 2D GRAPHICS, ACTIVITY DIAGRAM, PACKAGE DIAGRAM, USE CASE DIAGRAM, STATE DIAGRAM, GAMING SOFTWARE APPLICATION, COLLECTIBLE CARD GAME, SKELETON ANIMATION, СТ JS, JAVASCRIPT, KRITA, METROIDVANIA, SPINE, UML

The object of development is a game software application in the Metroidvania genre with elements of a card game.

The purpose of development is to create a gaming software application that will contain 6 levels and a mini-game in the Metroidvania genre with elements of a card collecting game.

The solution method used was the Ct.js development environment, Krita, a programme for creating illustrations and sprites, Spine, a programme for creating animations, and the JavaScript programming language.

As a result of the development, a gaming software application was created with a main character that can interact with levels (locations), enemy characters, NPC, a mini-game and collectible cards.

Я, Лухменко Ельвіра Костянтинівна, студентка гр. ПЗПІ-20-1, здобувачка вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Ігровий програмний застосунок у жанрі Metroidvania з елементами карткової гри», що буде представлена до екзаменаційної комісії для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайоmlена з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови до допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі	10
1.1 Аналіз предметної галузі.....	10
1.2 Виявлення та вирішення проблем	16
1.3 Постановка задачі.....	21
1.3.1 Цільова аудиторія	21
1.3.2 Визначення завдання	22
2 Формування вимог до програмної системи.....	24
2.1 Формування вимог	24
2.2 Виявлення та формування вимог.....	26
3 Архітектура та проектування програмного забезпечення	29
3.1 UML проектування ПЗ.....	29
3.2 Проектування архітектури ПЗ	35
3.2.1 Референси	35
4 Опис прийнятих програмних рішень	45
4.1 Інструментарій Krita	45
4.1.1 Шари.....	46
4.1.2 Використання фільтрів.....	46
4.1.3 Створення власного пензля.....	48
4.1.4 Інструментарій для побудови перспективи.....	50
4.1.5 Функція Wrap Around Mode.....	51
4.1.6 Інструментарій для анімацій.....	52
4.2 Інструментарій Spine	53
4.2.1 Використання скелетної анімації	54
4.2.2 Використання Skins	59
4.2.3 Робота з анімаціями	60
5 Тестування програмного забезпечення.....	63
5.1 Галузь застосування.....	63
5.2 Типи тестування	64
5.2.1 Тестування інсталяції	64

5.2.2 Функціональне тестування.....	65
5.2.3 Тестування інтерфейсу користувача.....	66
5.2.4 Тестування відмовостійкості та відновлення.....	66
5.2.5 Тестування продуктивності	67
5.2.6 Тестування конфігурації	68
5.3 Етапи тестування.....	69
5.4 Баг-репорт	70
6 Впровадження програмного забезпечення	76
6.1 Просування у соціальних мережах.....	76
6.2 Вихід у продакшн.....	81
Висновки	83
Перелік джерел посилання	85
Додаток А.....	87
Додаток Б.....	88
Додаток В	100
Додаток Г	119

ВСТУП

Ігровий програмний застосунок у жанрі Metroidvania [1] з елементами карткової гри є темою кваліфікаційної роботи. Основною метою даного проекту є створення ігрового досвіду, який занурить гравця у світ Київської Русі та ознайомить його зі слов'янською міфологією та культурою.

Завдання даної роботи передбачає розробку комплексної програмної системи, що включає в себе ігровий додаток доступний для використання на різних платформах. Її можна використовувати на персональних комп'ютерах з операційними системами Linux, Mac та Windows у вигляді програм, а також на мобільних пристроях з операційними системами Android та iOS, що дозволяє користувачам насолоджуватися грою у будь-який час та в будь-якому місці. Крім того, гравці можуть користуватися ігровим додатком через веб-версію, що збільшує його доступність.

Цей ігровий додаток є інноваційним в своєму роді, оскільки поєднує в собі елементи жанрів Metroidvania та карткової колекційної гри. Така суміш створює унікальний і захоплюючий геймплей, який відкриває перед гравцем широкі можливості для відчуття нових вражень та випробування власних навичок у віртуальному світі. Оскільки створений програмний застосунок є 2Д грою, для його реалізації було використано середовище розробки C#.js для програмування на мові JavaScript, програму для створення ілюстрацій та спрайтів Krita для графічної частини, а також програму для створення анімацій Spine для анімаційних ефектів. Перераховані інструменти були відмінно підібрані під встановлені цілі через свою виключну орієнтацію на 2Д графіку, що дозволило створити багатофункціональний, зручний та доступний ігровий додаток, що може зацікавити широке коло користувачів.

Ця робота має велике значення через поширення ігрових застосунків серед широкого спектру користувачів. В сучасному світі ігри не лише розважають, але і використовуються для навчання, розвитку навичок та спілкування. Їх застосування у педагогіці та освіті, а також у розважальній галузі, стає все більш популярним

явищем. Створення нових та цікавих ігор є ключовим аспектом розвитку програмного забезпечення. Ігрові застосунки не лише пропонують розвагу, але й можуть бути ефективним інструментом для вивчення культури, історії та міфології різних народів. Вони можуть стимулювати цікавість користувачів до вивчення нових тем і розширення їх знань. Ця робота зацікавить як любителів комп'ютерних ігор, так і тих, хто має бажання поглибити свої знання про історію України епохи Київської Русі, вивчити релігійність того періоду та освоїти народний фольклор. Таким чином, результати цієї роботи можуть мати велике значення для розвитку як освітніх, так і розважальних ігрових додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Сучасний цифровий світ наповнений різноманітними комп'ютерними іграми, які відрізняються за жанрами, цільовою аудиторією, технологіями, платформами, для яких вони призначені. Ця індустрія постійно розвивається, пропонуючи гравцям поринати у нові всесвіти. Завдяки широкому асортименту ігор, кожен зможе знайти застосунок, який відповідає його вподобанням та меті застосування. Іноді жанри можуть поєднуватися, створюючи нові гібридні версії, або навіть з'являтися цілком нові жанри, засновані на існуючих іграх, подібно до *Metroidvania*. Ця постійна еволюція індустрії комп'ютерних ігор створює безмежні можливості для творчості та інновацій, що сприяє нескінченному розширенню ігрового світу та задоволенню потреб кожного гравця.

Metroidvania - це особливий жанр ігор, що поєднує в собі елементи пригодницького бойовика та платформера. Він відзначається унікальними механіками геймплею та атмосферою, яка залежить від таємничості та дослідження. Цей жанр зберігає вірність класичним іграм серій *Metroid* і *Castlevania*, від яких і виникла назва "*Metroidvania*".

Головний акцент робиться на нестандартних рішеннях у розв'язанні головоломок та дослідженні взаємопов'язаних світів. У *Metroidvania* часто зустрічається велика карта, яку гравець може вивчати і досліджувати від краю до краю. Проте, деякі ділянки цього світу можуть залишатися недосяжними, поки гравець не здобуде спеціальні предмети, інструменти, або здібності, які необхідні для подолання перешкод. Кожен такий об'єкт відкриває нові можливості, допомагаючи гравцеві перемагати більш складних ворогів і розкривати секретні місця та проходи, які можуть залишатися прихованими в глибинах цього захоплюючого світу.

У ігор жанру *Metroidvania*, сюжет і геймплей сплітаються в одну взаємну нитку, створюючи захоплюючий імерсійний досвід для гравця. Кожен крок, кожне рішення, що робить гравець, впливає на подальший розвиток подій і веде його до

нових відкриттів та викликів. Такий підхід до геймплею не лише заохочує гравця досліджувати кожний куточок світу, але і запрошує його зануритися в неповторний досвід, який завойовує серця шанувальників ігор.

При розробці цього ігрового застосунку було не лише вирішено віддати перевагу новому жанру, такому як *Metroidvania*, а й поєднати його з колекційною картковою грою, створивши щось неповторне. Цей синтез відкрив нові можливості для гравців і побудував унікальний геймплей, який відрізняється від традиційних ігор обох жанрів.

Об'єднання *Metroidvania* з колекційною картковою грою створило широкий простір для творчості та експериментів. Гравці можуть насолоджуватися як розвідуванням великого і відкритого світу, так і стратегічними аспектами збирання та управління колекцією карток. Ця унікальна комбінація надає захоплюючий геймплей, який привертає до себе як любителів *Metroidvania*, так і шанувальників колекційних карткових ігор.

Ігрові застосунки відіграють різноманітні ролі у житті користувачів. Деякі із них спрямовані на простий відпочинок та розваги, допомагаючи відштовхнутися від повсякденного стресу та підвищити настрій. Інші ж мають на меті розвивати різні навички та якості особистості, такі як концентрація, швидкість реакції, або навіть розвивати комунікативні здібності.

Зокрема, існують ігри, що спрямовані на здобуття нових знань. Наприклад, вони можуть допомагати користувачеві освоїти нову мову, покращити математичні навички, або вивчити історію через захоплюючий сюжет та взаємодію з головним героєм, розташованим у конкретний історичний період.

Для виявлення проблем та коректної постановки завдання, перед створення програмного ігрового застосунку у жанрі *Metroidvania* з елементами колекційної карткової гри, який буде мати історичний сюжет та містити слов'янський фольклор, було проаналізовано певні його аналоги. А саме, серед цих систем було виділено чотири, що є провідними в ніші 2Д ігор зі схожими характеристиками:

- гра «Pentiment» [2];
- гра «Apotheon» [3];

- гра «Blasphemous» [4];
- гра «Castlevania Aria Of Sorrow» [5].

Однією з таких ігор є «Pentiment», яка є пригодницькою рольовою грою, що розгортається в баварському абатстві 1550-х років та відтворює життя художника рукописів. Гравець має розслідувати несподіване вбивство, щоб виправдати друга, якого звинуватили у цьому злочині. У грі відсутні звичні для рольових ігор бойові системи та інвентар. Гравець може взаємодіяти зі світом насамперед через діалоги з різними персонажами.

Гра представлена у вигляді рукопису епохи Відродження (див. рис. 1.1).



Рисунок 1.1 – Геймплей гри «Pentiment» (за даними [2])

Вона була створена для того, щоб поглибити знання гравців про історію, культуру та побут того періоду. Вона не лише розповідає захоплюючий сюжет, але й відображає реалії та атмосферу того часу через деталізоване відтворення абатства та його оточення. Гра дозволяє гравцям поринути у світ минулого, де вони можуть відчутися частиною історії та відкрити для себе нові факти і деталі життя та культуру того періоду. Вона також викликає рефлексію та задуми про те, які виклики та радощі стояли перед людьми того часу, допомагаючи гравцям отримати глибше розуміння історичного контексту та значення подій. Такі ігри, як «Pentiment», є не лише джерелом розваг, але й важливим засобом освіти та

відкриття нових горизонтів знань. Вони збагачують наше розуміння минулого та допомагають зберегти його спадок для майбутніх поколінь.

Наступною грою-аналогом є «Apotheon». Це екшн-платформер, який має унікальний стиль мистецтва, який нагадує вазопис із стародавньогрецьких міфів. Гравець відправляється на Олімп, де він взаємодіє з богами і монстрами, використовуючи різноманітні зброю та навички. Гра отримала позитивні відгуки за свій стиль мистецтва, геймплей і атмосферу, яка вдало відтворює дух давньогрецьких міфів. Варто підкреслити, що «Apotheon» надихає гравців на вивчення давньогрецьких міфів та легенд через захоплюючий геймплей і деталізованість світу гри. Вона не лише піднімає рівень дозвілля, але й стимулює інтерес до історичної та культурної спадщини, роблячи гру не лише розважальним інструментом, але й освітнім засобом.

Один з основних плюсів гри - це унікальний візуальний стиль, який надає їй вигляд мистецького шедедру, створеного на основі античного грецького вазопису. Кожен кадр гри вражає деталізацією і майстерністю відтворення стилю мистецтва, який переносить гравця в атмосферний світ античної Греції з її міфічними богами і героями. (див. рис. 1.2).

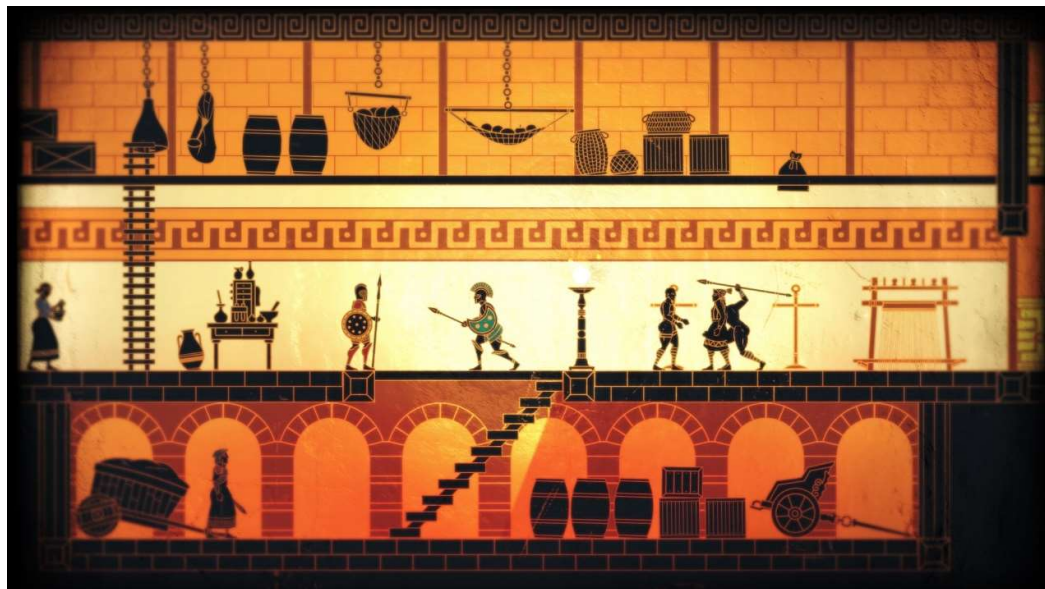


Рисунок 1.2 – Геймплей гри «Apotheon» (за даними [3])

Крім візуального аспекту, «Apotheon» також відзначається захоплюючим геймплеєм, який включає багато елементів екшну та пригод. Гравець має можливість взаємодіяти з різноманітними персонажами та використовувати велику кількість зброї і навичок у битвах. Кожен етап гри наповнений складними викликами та загадками, що додає грі глибини та інтриги.

Далі аналіз був присвячений такій грі, як «Blasphemous», яка є за жанром метроїдванією-екшеном. Гра розгортається у фентезійному світі, де гравець бере на себе роль мученика на ім'я The Penitent One. Головний герой подорожує по сутінковому світу, зустрічаючи різноманітних ворогів та розв'язуючи головоломки. «Blasphemous» - це не лише ігровий досвід, а й глибока подорож у темну атмосферу фентезійного всесвіту, де кожен куток наповнений загадкою і небезпекою. Гравець відчуває на собі вагу вини та спокути, подорожуючи крізь покаєння та боротьбу зі своїми внутрішніми демонами. Кожен бій, кожен крок, кожен вирішальний момент - це випробування, яке вимагає не лише майстерності в бою, але й мудрості та рішучості.

Як було вище сказано, гра має унікальну містичну атмосферу, яка демонструється з перших секунд запуску гри (див. рис. 1.4).



Рисунок 1.4 – Геймплей гри «Blasphemous» (за даними [4])

У «Blasphemous» кожна деталь, кожен образ, і кожен звук увібрав у себе крайню темність і трагізм, що поглинає гравця в світ безнадії та відчаю. Ця гра не просто розважає, вона зачаровує, занурюючи в атмосферу, де кожен вибір, кожен вчинок має вагу і наслідки. У «Blasphemous» переміщення та взаємодії передають почуття важкості і сили, що є відображенням глибоких конфліктів і внутрішньої боротьби.

Останньою грою, що була проаналізована є «Castlevania Aria Of Sorrow», яка є дія-пригодницькою та ролевою за жанром. Гравець приймає на себе роль Соми Круза, головного героя, який потрапляє в замок Дракули, розташований в таємничому світі. Головна особливість гри полягає в системі «Tactical Soul», де гравець може отримати та використовувати здібності (Soul) ворогів, яких він подолав. "Castlevania Aria Of Sorrow" - це захоплююча подорож у світ темряви та містики, де кожен куток замку приховує свої таємниці і небезпеки. Гра вдало поєднує в собі елементи екшену, пригод і ролевої гри, дозволяючи гравцеві зануритися в атмосферу готичного жаху. Головна система гри додає глибини інтерактивності, роблячи кожен бій захоплюючим випробуванням та можливістю використовувати унікальні здібності ворогів для власних цілей.

Гра має характерний вигляд для ігор 2000-х років випуску з піксельним інтерфейсом та спрощеним дизайном (див. рис. 1.5).

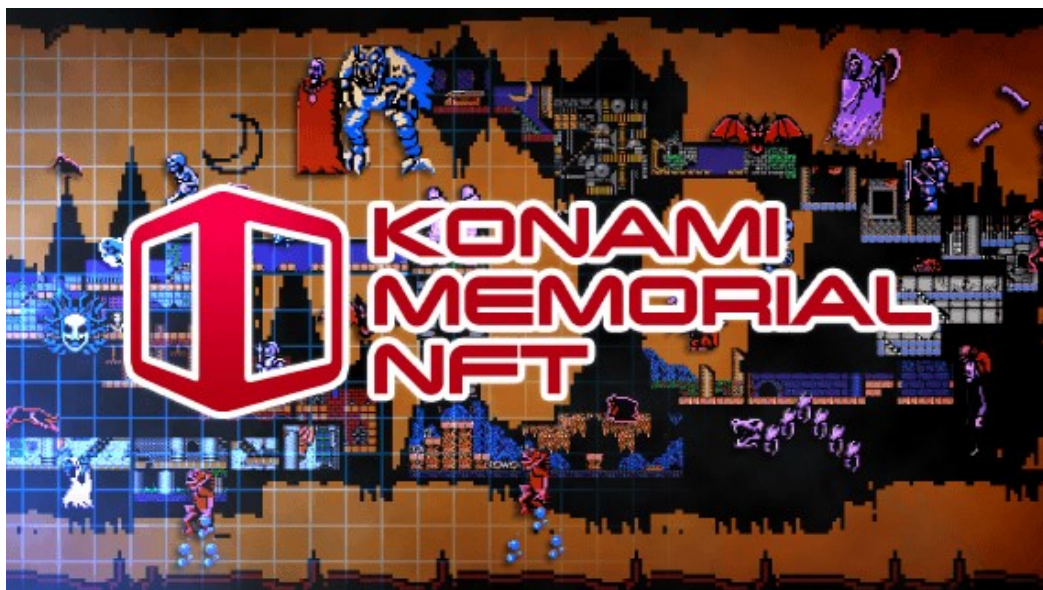


Рисунок 1.5 – Геймплей гри «Castlevania Aria Of Sorrow» (за даними [5])

У «Castlevania Aria Of Sorrow» кожен перехід через двері замку Дракули викликає хвилювання і очікування нових відкриттів і випробувань. Гра заворожує своєю відмінною атмосферою, яка переповнена містикою і загадковістю, посиляючись до історичних легенд та фольклору. Кожна знахідка та кожен бій збагачує історію героя і розкриває нові аспекти таємничого світу, який так манить і вабить гравця до далеких і небезпечних пригод.

1.2 Виявлення та вирішення проблем

У всесвіті комп'ютерних ігор, вибір правильного ігрового застосунку - перший великий виклик, що постає перед гравцями. Шукаючи те, що відповідає їхнім очікуванням та інтересам, вони стикаються з багатьма варіантами: іноді важко обрати саме той, що зможе захопити на довгі години. Однак і навіть після виявлення ідеальної гри, можуть виникнути нові виклики, які впливають на їхнє сприйняття та задоволення від ігрового процесу.

Ці виклики можуть бути різноманітними: від технічних проблем до відчуття монотонності у геймплеї. Наприклад, проблеми з графікою або звуковим супроводом можуть порушувати імерсивність враження від заняття, тоді як невдало реалізовані механіки або нудна геймплейна лінія можуть зменшити інтерес до гри.

Щоб забезпечити гравців якісним ігровим досвідом, розробникам необхідно постійно працюють над вирішенням цих проблем, вдосконалюючи графіку, розширюючи механіки та створюючи нові захоплюючі сюжети. Важливою частиною цього процесу є не лише виявлення проблем, але й пошук ефективних шляхів їх вирішення.

Як результат аналізу ігрової індустрії та конкурентів даного програмного продукту, були сформовані переваги та недоліки кожного з аналогів гри:

До переваг пригодницько-рольової гри «Pentiment» можна віднести наявність таких особливостей та можливостей:

- унікальний сюжет в історичному світі;
- нетиповий основний спосіб взаємодії зі світом для рольових ігор;

- унікальний художній стиль;
- зрозуміла система прийняття рішень;
- прості елементи RPG.

На жаль, як і будь-яка програмна система, ця гра має певні недоліки, які необхідно було відзначити під час аналізу. Вони пов'язані зі зручністю під час гри та реалізацією сюжету:

- специфічні діалоги, які є достатньо нудними та незрозумілими для середньостатистичного гравця;
- неможливість зберігатися під час гри; збереження є автоматичними(вбудованими), що робить неможливим перегравати деякі сюжетні лінії;
- усі вибори гравця є ефемерними, тобто ні до чого не призводять;
- часто наявні малюнки зі зламанною перспективою та відсутністю анатомії.

До переваг екшн-платформеру «Apotheon» можна віднести дані пункти:

- унікальний стиль мистецтва, створений на основі античного грецького вазопису;
- цікавий сеттінг;
- різноманітність зброї та навичок;
- гра вдало передає атмосферу давньогрецьких міфів;
- система розвитку персонажа.

«Apotheon» так само має недоліки, як у порівнянні з іграми того ж жанру, так і у зручності геймплею.

Основними недоліками є:

- обмежена графіка;
- бойова система та загальний рівень гри можуть бути складними для середньостатистичного гравця;
- не дуже довга тривалість гри.

Гра за жанром метроїдванія-екшн «Blasphemous» славиться своєю майстерністю в дизайні рівнів, атмосферною графікою та високим рівнем викликів у бою, тому можна виділити такі її переваги:

- унікальний темний, містичний та атмосферний дизайн з ретро-підходами до анімацій та візуальних ефектів;
- привабливий сюжет, який глибоко вбудований у фентезійний світ гри та надає гравцеві інтригуючий натиск на розкриття подій;
- велика важкість та потреба в ретельному плануванні стратегій в бою, що може влаштовувати любителів високого рівня виклику;
- контрольований і складний бой заснований на вмілому використанні зброї та навичок, що надає багато можливостей для експериментів;
- розмаїття ворогів і рівнів.

Також, як і свої попередники, гра має певну кількість недоліків, яка може докучати гравцям.

Основними недоліками є:

- хоча для багатьох висока важкість може бути перевагою, для інших це унеможлиблює проходження;
- навантажена атмосфера: темний і містичний стиль гри може бути занадто депресивним;
- можливість виснаження;
- обмежений вибір взаємодій.

До переваг дія-пригодницької та ролевої гри є «Castlevania Aria Of Sorrow» можна віднести:

- глибокий сюжет та атмосфера;
- система «Tactical Soul», яка додає стратегічний елемент, дозволяючи гравцеві використовувати здібності ворогів, що робить геймплей різноманітним та цікавим;
- вражаючий візуальний дизайн рівнів і деталізація персонажів;

- «Aria Of Sorrow» вводить кілька новаторських елементів у жанрі, включаючи систему «Tactical Soul» та наявність двох різних закінчень;
- гра надає високий рівень виклику, особливо в бою з сильними ворогами та босами, що влаштовує любителів випробувань.

Оскільки гра є найстарішою з усіх перерахованих, вона звісно має більшу кількість недоліків порівняно із сучасними:

- тривалість гри є досить корою;
- специфічний стиль гри;
- застаріла графіка;
- нестача інновацій в інших аспектах, які є занадто традиційними у порівнянні з іншими тайтлами того часу;
- складнощі в навігації.

Структурований аналіз вищеперерахованих аналогів було наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння аналогів (таблиця виконана самостійно)

Критерії	Pentiment	Apotheon	Blasphemous	Castlevania : Aria of Sorrow
Унікальний сюжет в історичному світі	Високий	Помірний	Слабкий	Слабкий
Унікальний художній стиль	Високий	Високий	Помірний	Слабкий
Простота навігації у просторі	+	-	+	+

Кінець таблиці 1.1

Критерії	Pentiment	Apotheon	Blasphemous	Castlevania : Aria of Sorrow
Прості елементи RPG	+	+	+	+
Сюжетні діалоги	+	-	-	+
Можливість зберігання	-	+	-	+
Вплив виборів на гру	-	-	-	-
Інноваційність у геймплеї	+	-	-	+
Наявність альтернативних закінчень	-	-	+	+
Система розвитку персонажа	-	+	+	+
Розмір світу	Помірний	Низький	Високий	Низький
Різноманітність об'єктів	-	+	+	-

У кінці кінців, кожна з ігор пропонує унікальні можливості, які можуть бути притаманні або відсутні у конкурентів. Серед них - інноваційність у геймплеї та вплив гравця на розвиток подій, що може призвести до різних результатів гри. Під час оцінки даної роботи враховувалися всі переваги аналогів, такі як унікальний художній стиль, цікаві сюжети з можливістю вибору, системи розвитку персонажів та новаторство, що може збагатити жанр ігор.

1.3 Постановка задачі

В контексті створення ігрового проекту, постановка задачі є критичним етапом, що визначає успіх та ефективність усього процесу розробки. Необхідно детально розглянути ключові завдання, що допоможуть структурувати процес роботи, визначити їхню актуальність та специфіку, а також сформулювати цілі, які потрібно досягти для успішного втілення програмного застосунку.

1.3.1 Цільова аудиторія

За для створення ігрового застосунку, який би вразив користувачів своєю привабливістю, надавав їм позитивний досвід та стимулював розвиток, необхідно ретельно вивчити та зрозуміти цільову аудиторію гри. Це означає не лише визначити хто ці люди, але й з'ясувати їхні вподобання, потреби, інтереси та очікування. Лише глибоке розуміння аудиторії дозволить створити продукт, який точно задовольнить їхні потреби і буде відповідати їхнім очікуванням.

Після огляду статистики користувачів ігор-аналогів було виявлено, що програмний ігровий застосунок спрямований на аудиторію з такими певними характеристиками:

- вік від 13 до 25 років;
- стать переважно чоловіча, оскільки головним героєм є персонаж чоловічої статі, а найголовнішою механікою гри є поєдинки;
- наявність інтересу до вивчення історії;
- цікавість до слов'янського фольклору та міфології;
- мати будь-яку професію, працю або освіту;

- мати хоча б невеликий досвіт у роботі з гаджетами та з іграми.

Після встановлення цільової аудиторії даної гри можна стверджувати, що програмний ігровий застосунок має спрямовуватися на молоду чоловічу аудиторію, основними характеристиками яких є інтерес до вивчення історії, зацікавленість у слов'янському фольклорі та міфології, рівень освіти та невеликий досвіт у роботі з гаджетами та іграми. Ця інформація дозволяє зорієнтуватися під час створення продукту, що буде відповідати потребам та інтересам цільової аудиторії.

1.3.2 Визначення завдання

Після уважного аналізу як позитивних, так і негативних аспектів ігор-конкурентів, а також вивчення цільової аудиторії програмного продукту, можна визначити конкретні завдання для гри. Ці завдання стануть орієнтиром під час розробки та нададуть можливість створити унікальний застосунок, що повністю задовольнить очікування користувачів. За допомогою чітко прописаних пунктів, вдасться зосередитися на ключових аспектах гри та використати їх для створення неповторного і захоплюючого ігрового досвіду. Такий підхід також сприятиме підвищенню конкурентоздатності програмного продукту на ринку ігор.

Отже, можна виділити такі фундаментальні задачі:

- музичний та візуальний супровід, повинен відповідати атмосфері у грі;
- діалоги повинні бути нетиповими та цікавими, одночасно не бути нудними та затяжними;
- наявність речей, які будуть залежати від вибору гравця;
- наявність різноманітності ворогів з різними механіками;
- присутність вибору різних стратегій бою, за допомогою використання колекційних карт;
- світ повинен бути обмеженим, але мати велику кількість локацій;
- посилаючись на жанр Metroidvania, деякі кімнат повинні бути недоступними до певних сюжетних або пошукових подій;

- об'єднання у бойовій системі декількох механік, що зробить гру інноваційною;
- інтерфейс гри повинен бути простим та інтуїтивним;
- гра повинна бути побудована на унікальній історії, яка не була більше ні де реалізована;
- художній стиль повинен бути досить унікальним у своєму роді;
- наявність простих елементів RPG.

Результати аналізу і визначені завдання є критичними для успішного розвитку програмного ігрового продукту. Чітко встановлені цілі дозволяють зосередитися на ключових аспектах гри, які будуть створювати її унікальність та привабливість для гравців. Крім того, ретельний аналіз дозволяє врахувати попередні недоліки конкурентів і забезпечити їх виправлення у власному продукті, підвищуючи його конкурентоспроможність. Такий підхід сприяє створенню додатку, що не лише задовольняє очікування гравців, але й перевершує їх, роблячи гру бажаною та популярною серед аудиторії.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Формування вимог до програмної системи [6] є критичним етапом у процесі розробки будь-якого програмного продукту, включаючи ігрові застосунки. Їх правильне визначення дозволяє не лише забезпечити високу якість програми, а й зробити її більш корисною та привабливою для кінцевого користувача.

Необхідно розглянути процес формування вимог до програмного ігрового застосунку, зокрема, звернути увагу на платформи, на яких гра буде доступна, та встановити функціональні та нефункціональні вимоги, що будуть ключовими в отриманні користувацького досвіду та ефективності роботи програми. Далі потрібно приділити увагу процесу визначення потреб гравців, а також аналізу ризиків, що можуть вплинути на успішність розробки та експлуатацію програмного продукту.

2.1 Формування вимог

Аналіз предметної області під час формування вимог до програмної системи є ключовим етапом у процесі розробки, оскільки він дозволяє отримати глибоке розуміння контексту, в якому буде використовуватися програма. У випадку ігрового застосунку це означає вивчення системних потреб та платформ, на яких розміщуються існуючі ігрові аналоги.

Для з'ясування системних вимог було ретельно проаналізовано три раніше згаданих ігрових конкурента, що схожі за своєю наповненістю об'єктами, локаціями та персонажами, тривалістю, графічною та жанровою спрямованістю.

Така казуальна гра, як «Pentiment» має велику кількість схожих компонентів з розроблюваною програмною системою, які безпосередньо впливають на формування системних потреб гаджету. До таких елементів можна віднести:

- 2Д графіку;
- відсутність класичного інвентаря з предметами;
- наявність NPC, з якими можна взаємодіяти;
- ідентичне переміщення по локації та між локаціями.

Наступним кроком, було знайдено та проаналізовано на офіційних джерелах системні вимоги до даної гри:

- операційна система: Windows 7 (SP1) 64bit;
- процесор: Intel Core i3-3225;
- оперативна пам'ять: 4 GB;
- відеокарта: Nvidia GTX 650 Ti;
- місце на диску: 12 GB доступного місця.

Пригодницька гра «Blasphemous» також має низку компонентів, що перетинаються з даним програмним продуктом. А саме:

- однаковий жанр(Metroidvania);
- 2Д графіка;
- різноманіття музичного супроводу, яке змінюється в залежності від подій у грі;
- схожа діалогова система з NPC.

З приводу системних вимог можна відзначити мінімальні потреби за основними критеріями:

- операційна система: Windows 7, 8, 10, MacOS Sierra - 10.12.6, Ubuntu 16.04.01;
- процесор: Intel Core2 Duo E8400, AMD Phenom II x2 550 або 2.7 GHz Intel Core i5;
- оперативна пам'ять: 4 GB;
- відеокарта: GeForce GTX 260,Radeon HD 4850 або Nvidia GeForce GT 640M;
- місце на диску: 4 GB доступного місця.

Пригодницько-рольова гра «Apotheon» має особливу схожість у ігровому процесі оскільки за жанром є грою-платформером. І ось які можна відзначити спільні критерії:

- спосіб переміщення світом та наявність платформ;
- 2Д графіка;

- схожа бойова система.

Можна відзначити такі мінімальні рекомендовані системні вимоги до даного програмного застосунку:

- операційна система: XP/Vista/Windows 7, MacOS version 10.9 Mavericks or newer, SteamOS + Linux glibc 2.17+, 64-bit only;
- процесор: Dual Core 2.5 GHz;
- оперативна пам'ять: 1-3 GB;
- відеокарта: DirectX 9.0c compatible, OpenGL 3.0+ support (2.1 with ARB extensions acceptable);
- directX: версії 9.0c;
- звукова карта: DirectX 9.0c compatible sound cards;
- місце на диску: 2 GB доступного місця.

2.2 Виявлення та формування вимог

St.js [7], яка була обрана як середовище розробки для цієї гри, відзначається своєю високою ефективністю та простотою в експорті ігрового проекту на різні операційні системи, такі як Windows, Linux, Mac, Android та iOS. Зокрема, за допомогою Electron JS гра може бути відкрита як web-додаток, що розширює її доступність до широкого кола користувачів. Це забезпечує гнучкість та доступність програмного ігрового застосунку на різних платформах, що є важливим фактором для його успішності.

Після ретельного аналізу аналогів та вивчення можливостей оптимізації компонентів під час експорту з St.js, були встановлені мінімальні системні вимоги до програми. Цей аналіз дозволив точно визначити потреби програмного продукту і забезпечив його оптимальну продуктивність на різних пристроях та платформах. А саме, були встановлені рекомендовані вимоги для кожної операційної системи:

a) Windows:

- 1) операційна система: Windows 7, 8, 10, 11 64bit;
- 2) процесор: An Intel Pentium 4 або новіший, який підтримує SSE2;

- 3) оперативна пам'ять: 512 MB;
- 4) відеокарта: NVIDIA GeForce GT 640M;
- 5) звукова карта: DirectX Compatible;
- 6) місце на диску: 2 GB доступного місця;

б) Mac;

- 1) операційна система: macOS 10.10 і вище 64bit;
- 2) процесор: An Intel 64bit;
- 3) оперативна пам'ять: 512 MB;
- 4) відеокарта: OpenGL 3.0+;
- 5) місце на диску: 3 GB доступного місця;

в) Linux;

- 1) операційна система: Ubuntu 12.04/ Fedora 21/ Debian 8 і вище 64bit;
- 2) процесор: An Intel Pentium 4 або новіший, який підтримує SSE2;
- 3) оперативна пам'ять: 512 MB;
- 4) відеокарта: OpenGL 3.0+;
- 5) місце на диску: 3 GB доступного місця;

г) Android;

- 1) Версія: 7 або новіша;
- 2) Графічний процесор: 700 МГц або вище;
- 3) Оперативна пам'ять: 2 GB;
- 4) Внутрішня пам'ять: 200 MB;
- 5) Процесор: Octocore з частотою 1,7 ГГц або вище;
- 6) OpenGL: версія 3 або новіша;

д) iOS;

- 1) версія: 6S або новіша;
- 2) графічний процесор: PowerVR GT7600;
- 3) оперативна пам'ять: 2 GB;
- 4) внутрішня пам'ять: 200 MB;
- 5) процесор: Двохядерний з частотою 1,8 ГГц або вище.

Отже, після ретельного аналізу аналогів та вивчення можливостей оптимізації компонентів під час експорту з C#.js, було встановлено мінімальні системні вимоги до програми для операційних систем Windows, Mac, Linux, Android та iOS. Цей аналіз дозволив точно визначити потреби програмного продукту і забезпечити його оптимальну продуктивність на різних пристроях та платформах. Зазначені рекомендовані вимоги до обладнання та операційних систем гарантують плавний та ефективний запуск гри на різних пристроях, що дозволяє користувачам насолоджуватися ігровим досвідом без будь-яких перешкод чи обмежень.

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В даному програмному ігровому застосунку велике значення має розділ «Архітектура та проєктування програмного забезпечення». Цей етап є одним з найважливіших у процесі розробки, оскільки він визначає структуру та організацію всього продукту. У даному розділі буде детально розглянуто процеси, які стосуються створення архітектури програми, включаючи вибір технологій, моделювання компонентів, побудову діаграм та написання відповідної документації. Ознайомлення з цими аспектами допоможе забезпечити успішну реалізацію та високу якість ігрового застосунка.

3.1 UML проєктування ПЗ

У проєктуванні програмного забезпечення велику роль відіграє використання Unified Modeling Language (UML) [8]. UML є стандартною мовою моделювання, що дозволяє візуалізувати різні аспекти програмних систем, від структури до поведінки. Необхідно детально розглянути різні види діаграм UML, такі як діаграми прецедентів, стану, активності тощо, і визначити їх застосування у процесі розробки програмного забезпечення. Окрім того, потрібно визначити основні принципи та правила побудови діаграм UML, а також їхнє використання для аналізу, проєктування та документування програмних систем. Оволодіння цими концепціями та інструментами допоможе ефективно змодельовати ігровий застосунок для досягнення поставлених цілей та вимог.

По-перше, варто зазначити, що у програмному ігровому застосунку у жанрі Metroidvania з елементами карткової колекційної гри можна виділити три режими роботи програмного забезпечення, а саме:

- режим основної гри;
- режим карткового бою;
- режим діалогу та інтеракції.

Для того, щоб правильно спланувати роботу даних режимів, були створені UML діаграми.

Розглядаючи режим основної гри, можна сказати, що він охоплює широкий спектр інтерактивних елементів, які включають в себе не лише основний геймплей, а й бої з ворогами, збір колекційних карт для покращення персонажа та використання їх у бою та переміщення по рівнях гри. Основний геймплей відтворює центральну механіку гри та включає в себе серію завдань або викликів, які гравець повинен виконати для прогресу.

Щоб краще зрозуміти, які користувачі взаємодіють із системою та яким чином вони це роблять, була створена діаграма прецедентів [9] (див. рис. 3.1).

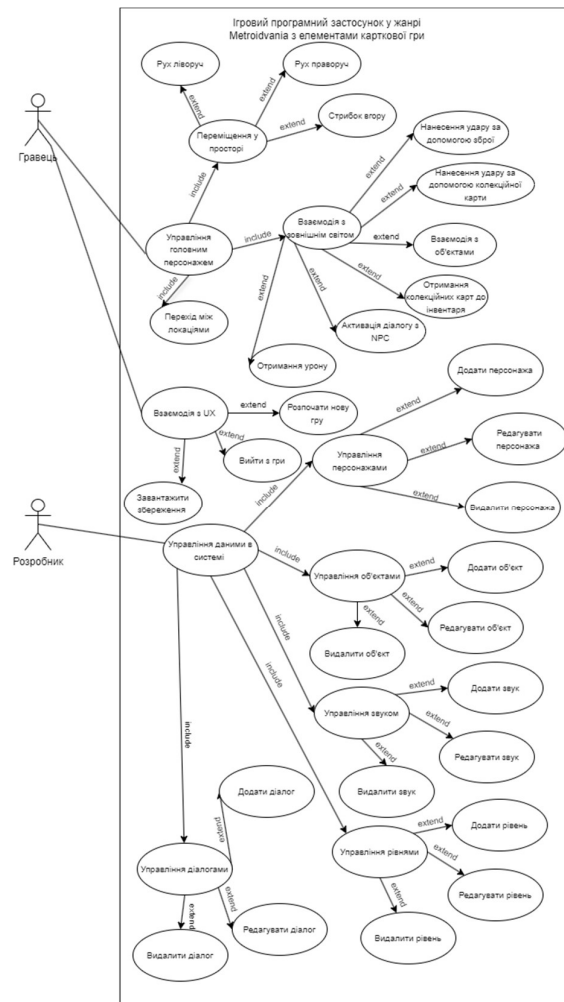


Рисунок 3.1 – Діаграма прецедентів ігрового програмного застосунку (рисунок виконаний самостійно)

Діаграма прецедентів - це інструмент у мові моделювання UML, який допомагає описати, як користувачі взаємодіють із системою. Вона фокусується на

тому, які дії виконують користувачі та як система на них реагує. Кожен прецедент це окремий функціональний аспект системи, що описується у контексті користувача та системи. Діаграма містить прецеденти (функції системи), акторів (користувачів або зовнішні системи) та зв'язки між ними, що показують взаємодію. Вона допомагає зрозуміти, як система повинна працювати для користувачів і є важливим інструментом для аналізу та проектування ПЗ.

Вище представлена Use-case діаграма відображає основну взаємодію із системою гравця та розробника. Гравець не може виконувати прямі дії з початковими ресурсами програми, але йому надана можливість керувати переміщенням та заздалегідь прописаними діями головного персонажа. Також він може взаємодіяти з UX, що є достатнім для запуску гри та виходу з неї. У свою чергу розробник має можливості додавати, редагувати та видаляти об'єкти, які будуть оточувати гравця та якими він буде керувати.

Для уявлення про взаємодію користувача з ігровим середовищем найбільш доцільно використовувати діаграму діяльності [10] (див. рис. 3.2).

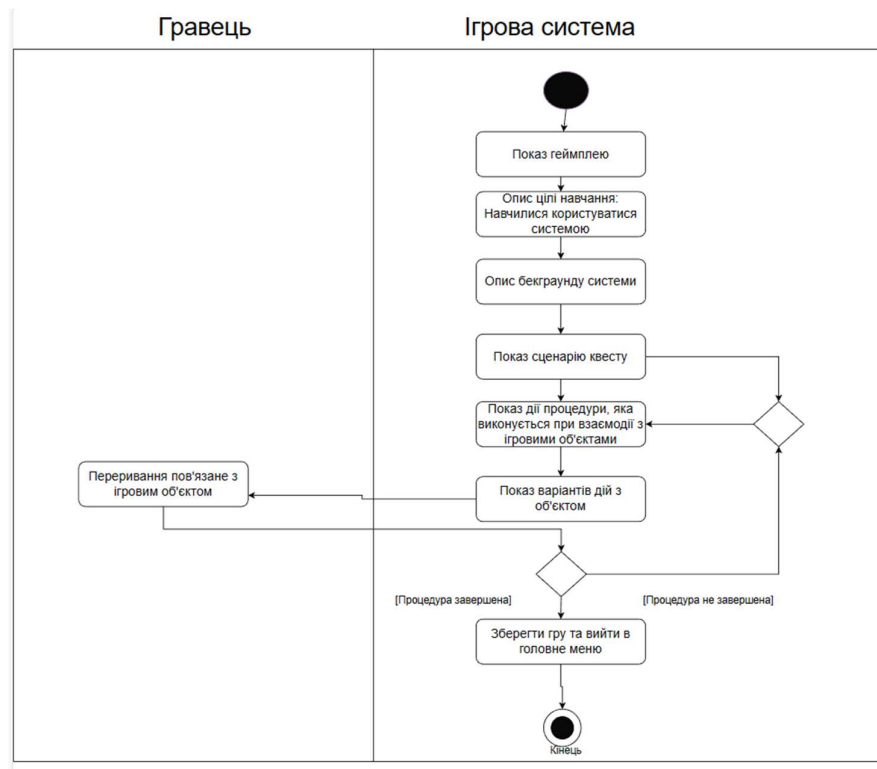


Рисунок 3.2 – Діаграма діяльності ігрового програмного застосунку (рисунок виконаний самостійно)

Діаграма діяльності - це графічний інструмент, який використовується для моделювання послідовності дій або процесів у системі. Вона дозволяє наглядно відобразити події чи кроки, що відбуваються у певній частині функціоналу. Ця діаграма використовується для уточнення деталей щодо того, як система повинна взаємодіяти з користувачем або іншими системними компонентами у різних сценаріях використання. Вона допомагає розуміти логіку та послідовність операцій, сприяючи кращому розумінню її функціональності та динаміки роботи.

У представлений Activity diagram демонструється послідовність виконання різних процесів в ігровій системі. На певному етапі програма пропонує гравцю певну взаємодію з неї, після чого процедура завершується. Один з можливих варіантів завершення - збереження гри та повернення до головного меню.

У режимі "Карткового Бою" гравець отримує можливість активно взаємодіяти з колекційними картами, використовуючи їх стратегічно під час боїв. Кожна карта може бути використана для поліпшення характеристик головного персонажа або для проведення атак проти ворогів. Цей режим дозволяє гравцеві виконувати тактичний підхід і обирати оптимальну стратегію битви, враховуючи поточну ситуацію на полі бою.

Розглядаючи режим "Діалог та Інтеракція", гравець має можливість взаємодіяти з NPC (неігровими персонажами) та іншими героями з метою отримання інформації або виклику подій, які впливають на розвиток сюжету. Цей режим створює можливість для гравця визначати долю головного персонажа шляхом вибору різних діалогових опцій та взаємодії з ігровим оточенням. Найкращим відображенням діалогової системи є діаграма стану (див. рис. 3.3).

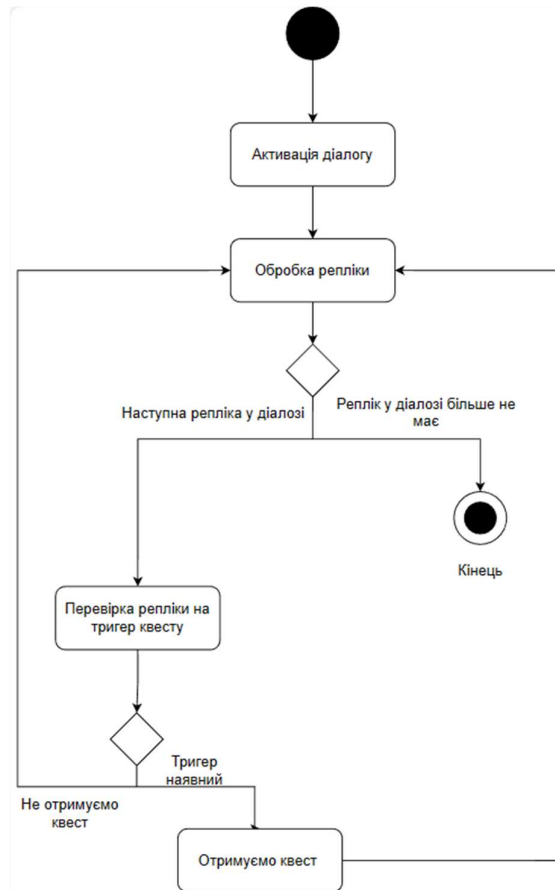


Рисунок 3.3 – Діаграма стану ігрового програмного застосунку(рисунок виконаний самостійно)

State diagram є важливим інструментом для моделювання поведінки програмного продукту у відповідь на різні вхідні події. Вона відображає можливі стани системи та переходи між ними в залежності від здійснених дій користувача або зміни у ній самій. Цей тип діаграми дозволяє краще зрозуміти роботу програми та визначити її реакцію на різні умови чи події. Його можна використовувати як інструмент для аналізу та проектування програмного забезпечення, допомагаючи розробникам зрозуміти логіку взаємодії компонентів системи.

На діаграмі стану приведеній вище відображена послідовність подій під час діалогу. Після активації кожної репліки відбувається дві перевірки: перша перевіряє чи є вона останньою, друга – чи тригерить вона отримання квесту. Під час першої умови діалог закінчується, якщо його продовження немає. Якщо

наступна репліка присутня, активується друга перевірка. Не важливо отримуємо гравець квест чи ні на наступній умові, обробка починається знов.

Для структурованого відображення організації збереження даних була створена діаграма пакетів [11] (див. рис. 3.4).

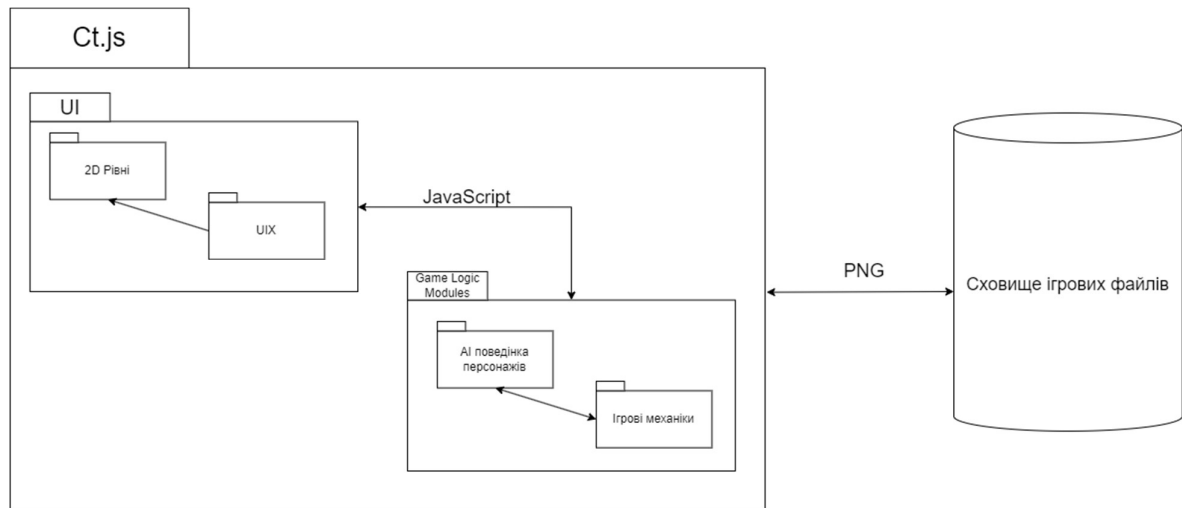


Рисунок 3.4 – Діаграма пакетів ігрового програмного застосунку(рисунок виконаний самостійно)

Package diagram є діаграмою, яка допомагає організувати компоненти програми та їх взаємозв'язки. Вона дозволяє ідентифікувати основні модулі та їх залежності між собою. Завдяки діаграмі пакетів розробники краще розуміють структуру програмного забезпечення, що сприяє його подальшому розвитку та підтримці.

На вище представлений діаграмі було відображено два основних модулі: файл програми st.js та сховище ігрових компонентів, які взаємодіють між собою за допомогою png файлів. Основний пакет можна поділити на UI та модулі ігрової логіки. За допомогою мови програмування JavaScript була побудована комунікація між ними. У свою ж чергу UI містить 2Д рівні та UIX, а моделі ігрової логіки – ігрові механіки та поведінку персонажів.

Подібна структура допомагає легко орієнтуватися між компонентами програми та налаштувати зв'язок між ними, сприяючи зручній роботі та підтримці програмного забезпечення. Розподіл на UI та модулі ігрової логіки дозволив чітко

виділити функціональність інтерфейсу користувача від основної логіки гри, що полегшує розвиток та вдосконалення програми. Такий підхід до організації програмного забезпечення сприяє покращенню продуктивності розробки та забезпечує більшу зрозумілість його структури.

3.2 Проектування архітектури ПЗ

У процесі розробки будь-якого програмного продукту, включаючи ігрові застосунки, одним з важливих етапів є проектування архітектури програмного забезпечення. Цей етап визначає структуру, компоненти та взаємозв'язки між ними, що визначає основні характеристики системи. Проектування архітектури вимагає вивчення вимог до програми, а також аналізу і розробки оптимальних рішень для забезпечення високої якості, ефективності та розширюваності програмного забезпечення.

3.2.1 Референси

Перш ніж перейти до створення програмного застосунку, вирішальним є вибір інструментів, що використовуватимуться для реалізації проекту. Середовище розробки `St.js`, що базується на `Electron`, було обране, як головна програма для побудови програмного ігрового застосунку. Вона відома як один з найкращих інструментів для створення 2Д ігор. Не лише його потужність, але й зручний, адаптивний інтерфейс з усім необхідним інструментарієм та детальна документація роблять його незамінним помічником у розробці. Крім того, його експортна система дозволяє легко адаптувати гру під різні операційні системи. Після докладного аналізу переваг `St.js` очевидно, що це оптимальний вибір для створення ігрового програмного застосунку.

Більш складний став вибір програмних систем для створення візуальної складової гри. Це не менш важлива частина у розробці продукту, оскільки жодна механіка не може бути повністю реалізована без графічного дизайну рівнів, об'єктів та персонажів. Тому було поставлено завдання вибрати оптимальні

інструменти для цього. Було проведено аналіз по трьом програмам для створення графіки та анімацій, зробивши акцент на їхньому функціоналі та можливостях.

а) програмні системи для дизайну:

- 1) Adobe Photoshop [12];
- 2) Krita [13];
- 3) Clip Studio Paint [14];

б) програмні системи для анімації;

- 1) Spriter Pro [15];
- 2) Spine [16];
- 3) Adobe After Effects [17].

Варто почати з усіма відомою програмою, такою як Adobe Photoshop, яка використовується для редагування та обробки графіки, а також широко поширена у галузі дизайну, фотографії та графічного мистецтва(див. рис. 3.5).



Рисунок 3.5 -Логотип Adobe Photoshop (за даними [12])

Основні переваги:

- широкий функціонал: Photoshop має вражаючий набір інструментів для редагування фотографій, створення графіки, обробки зображень та створення складних дизайнів;
- професійна якість: Це стандарт в галузі графічного дизайну, і його функції забезпечують високу якість результату;
- інтеграція з іншими продуктами Adobe: Легка взаємодія з великою

кількістю програм, такими як Illustrator, InDesign, After Effects, що полегшує обмін ресурсами та проектами;

- широкий спектр фільтрів та ефектів для творчої обробки зображень;
- підтримка роботи з шарами, що дозволяє працювати з різними елементами проекту окремо сприяючи організації та редагуванню.

Не можна обійти стороною недоліки цього програмного продукту, які пов'язані на самперед з його популярністю та вмістом великої кількості функціоналу.

Основними недоліками є:

- складність в освоєнні: Досить важко орієнтуватися у всіх функціях та можливостях програми через її обширний функціонал. Наявність зайвих речей, які ускладнюють взаємодію з необхідним інструментарієм;
- потреба в потужному обладнанні: Деякі функції можуть вимагати значних ресурсів комп'ютера, особливо при роботі з великими файлами.

Наступне програмне забезпечення, яке було проаналізовано, це Krita. Це потужне програмне забезпечення призначене для малювання та редагування графіки, що користується великою популярністю серед художників, ілюстраторів та аніматорів (див. рис. 3.6).

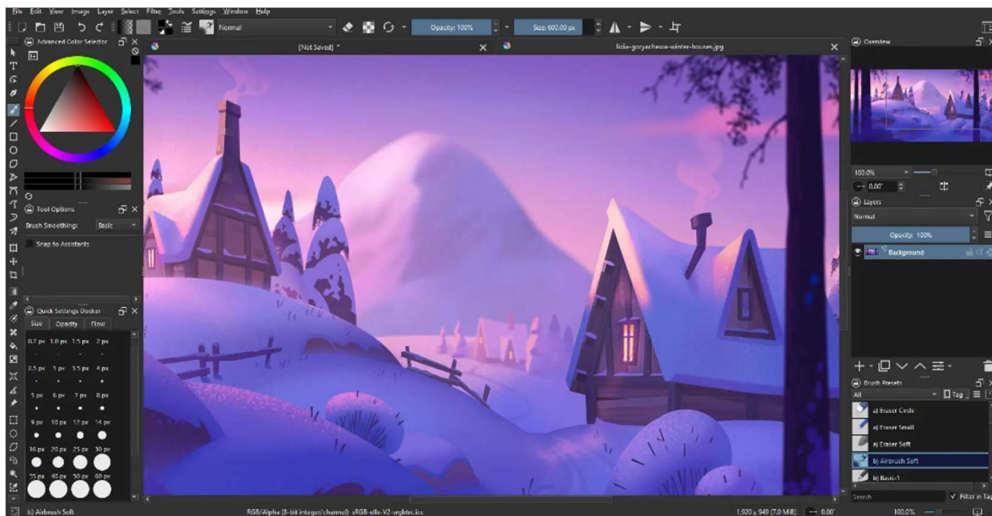


Рисунок 3.6 - Інтерфейс Krita (за даними [13])

Ось деякі переваги та недоліки Krita:

- великий інструментарій малювання, який включає в себе пензлі, текстури, фільтри та інші засоби для творчого виразу;
- наявність інструментів для створення простих анімацій;
- підтримка роботи з шарами;
- наявність зручного інструментарію для створення векторної графіки;
- зручний інтерфейс, який орієнтований першочергово на роботу з растровою графікою.

Krita також має певні недоліки:

- деякі функції можуть вимагати значних ресурсів комп'ютера, особливо при роботі з великими файлами або складними проектами;
- підтримка текстових елементів обмежена;
- можливі проблеми з сумісністю при взаємодії з іншими файлами.

Останнім у порівнянні програм для ілюстраторів стає Clip Studio Paint, який слугує для малювання та редагування графіки, спеціально розроблений для манги та коміксів, а також використовується для створення анімації (див. рис. 3.7).

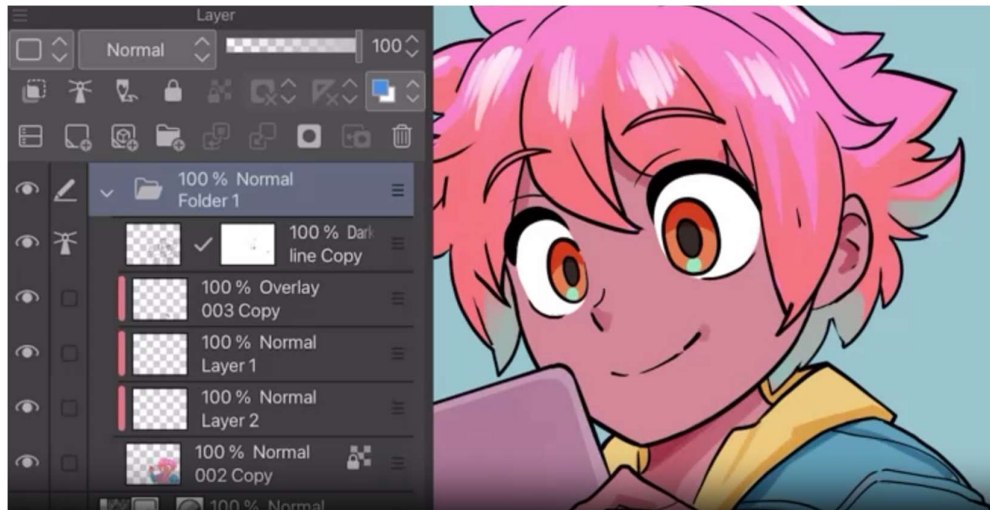


Рисунок 3.7 - Інтерфейс Clip Studio Paint (за даними [14])

Ось деякі переваги цього програмного продукту:

- орієнтованість на мангу та комікси: Програма оптимізована для створення манги та коміксів, надаючи спеціальні інструменти та функції для цього типу мистецтва;

- наявність розширеного інструментарію для рисунку, що дозволяє художникам створювати деталізовані та виразні ілюстрації;
- можливості для створення анімацій;
- широкий спектр інструментів для роботи з векторною та растровою графікою;
- підтримує зручне розміщення та стилізацію тексту, сприяючи роботі з діалогами та текстовою інформацією в коміксах.

Програма має такі недоліки у порівнянні зі схожими з нею програмами:

- складний та не інтуїтивний інтерфейс з великою кількістю зайвих інструментів;
- майже не використовується за межами галузі створення манги та коміксів;
- можливі обмеження в експорті та імпорті файлів.

У контексті спеціалізованих програм для створення анімацій, варто звернути увагу на Spriter Pro, яка орієнтована на роботу з 2Д об'єктами. Це програмне забезпечення надає аніматорам можливість створювати складні анімації, ефективно використовуючи спрайти та скелетну техніку (див. рис. 3.8).

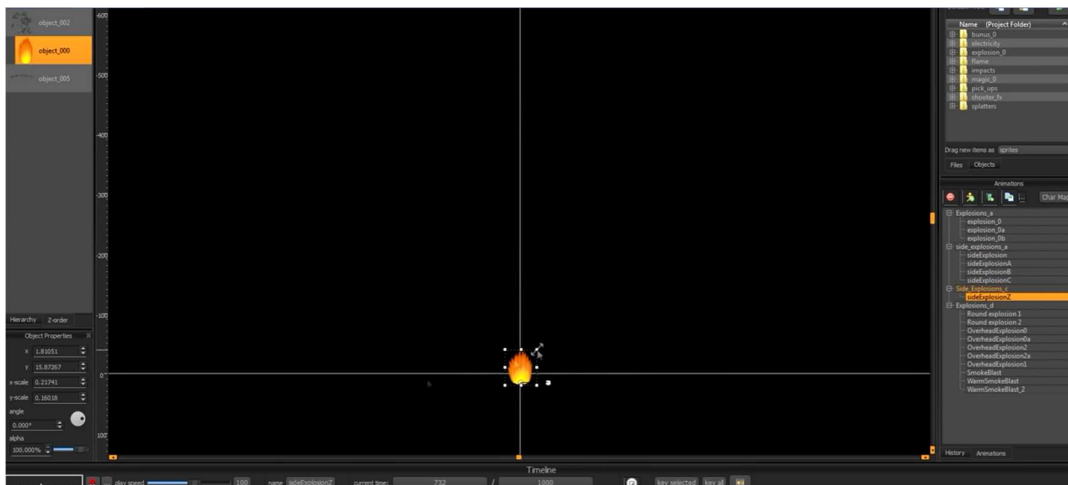


Рисунок 3.8 - Інтерфейс Spriter Pro (за даними [15])

Переваги Spriter Pro:

- наявність скелетної анімації, що полегшує роботу з рухом персонажів та інших об'єктів;

- легке створення переходів між анімаціями та застосуванням різноманітних ефектів для поліпшення візуальної привабливості;
- підтримка експорту анімацій у різні формати, що робить їх сумісними з великою кількістю ігровими двигунами та платформами;
- низькі вимоги до ресурсів системи, за вдяки яким можна працювати ефективно навіть на менш потужних комп'ютерах.

На жаль, як і будь-який програмний продукт Spriter Pro має свої недоліки у порівнянні з іншими програмами:

- має менш розгалужений функціонал;
- погана документація та мала кількість ресурсів для навчання;
- відсутність автоматичного розміщення елементів для правильної роботи скелетної анімації;
- можливі проблеми із сумісністю при використанні з деякими ігровими двигунами або програмами.

Наступним кроком була проаналізована програма Spine, яка є інструментом для створення скелетної анімації. Вона використовується в ігровій розробці та інших проектах, де потрібна ефективна робота з анімованими персонажами та об'єктами (див. рис. 3.9).

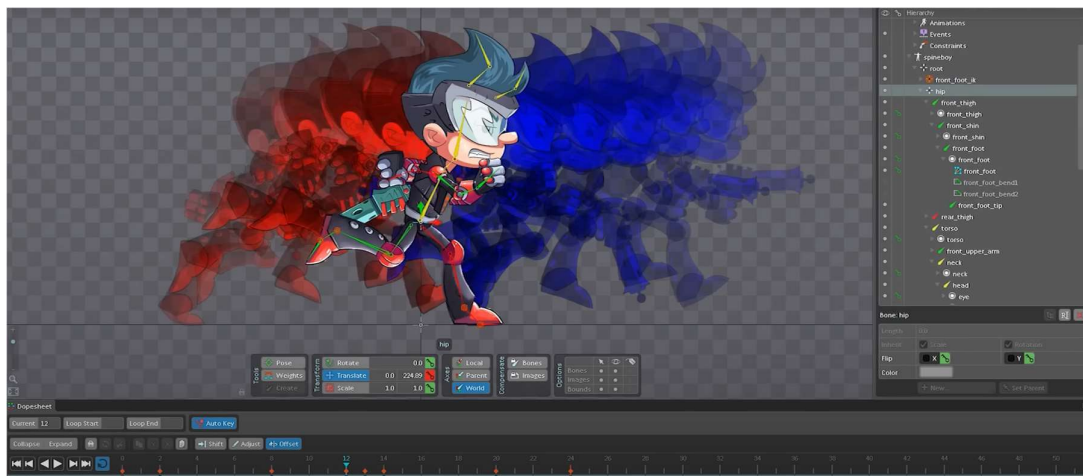


Рисунок 3.9 - Інтерфейс Spine (за даними [16])

До переваг програмного застосунку Spine можна віднести наявність таких особливостей та можливостей:

- орієнтація на скелетну анімацію, що полегшує створення реалістичних рухів для персонажів та об'єктів;
- наявність ефективних методів управління ресурсами та пам'яттю, що покращує оптимізацію ігрових застосунків;
- підтримка експорту анімацій у різні формати, що робить їх сумісними з різними ігровими двигунами та платформами;
- гарна та докладна документація.

Не можна також пройти повз недоліків Spine:

- не підтримує ні які види анімації або обробки окрім скелетної;
- відсутня підтримка декількох проєктів в одному вікні;
- не можливе додавання вже існуючих анімацій до нового скелету.

І на останок, Adobe After Effects - програмне забезпечення для створення анімацій, візуальних ефектів, титрів та іншої графіки для відео. Він широко використовується в індустрії відеопродакшену та має ряд корисних функцій (див. рис. 3.10).



Рисунок 3.10 – Інтерфейс Adobe After Effects (за даними [17])

До переваги Adobe After Effects можна віднести:

- наявність широкого функціоналу з безліччю інструментів для створення різноманітних ефектів, анімацій, титрів тощо;
- інтеграція з іншими програмами Adobe, що дозволяє легко обмінюватися проектами та ресурсами;
- наявність 3D простору під час створення анімацій, що дозволяє зробити рисунок багатоплановим з імітацією 3D;
- підтримка скелетної анімації.

Оскільки Adobe After Effects орієнтований в першу чергу на створення відео, він має певний ряд недоліків:

- великі вимоги до ресурсів, що може впливати на продуктивність роботи програми під час використання великих та складних проектів;
- багато зайвого функціоналу, які не стосуються створення анімацій;
- складний в освоєнні через неінтуїтивний інтерфейс;
- процес рендерингу займає значний час;
- орієнтований на векторну графіку, погана підтримка растрової;
- майже відсутня інтеграція з ігровими середовищами розробки.

Далі були порівняні між собою програми-аналоги, які використовуються для створення дизайну та анімації для ігрових середовищ розробки. У цьому порівнянні низький рівень відповідності отримав оцінку в 1 бал, середній – 2 бали, а високий – 3 бали.

Бали за критеріями для програм, в яких створюють ілюстрації, були наведені в таблиці 3.1.

Таблиця 3.1 – Порівняння систем для створення дизайну гри (таблиця виконана самостійно)

Критерії	Adobe Photoshop	Krita	Clip Studio Paint
Інструментарій для малювання(5)	Низький	Високий	Високий

Кінець таблиці 3.1

Критерії	Adobe Photoshop	Krita	Clip Studio Paint
Взаємодія із відкритими джерелами(4)	Висока	Висока	Середня
Невибагливість до ресурсів системи (5)	Низька	Середня	Середня
Інтуїтивність інтерфейсу(3)	Низька	Висока	Середня
Функціональність для малювання(5)	Висока	Висока	Висока
Підтримка форматів(4)	Висока	Висока	Висока
Інтеграція з ігровими двигунами(4)	Середня	Середня	Середня

В результаті розрахунків було отримано дані результати:

- Adobe Photoshop – 60 балів;
- Krita – 96 балів;
- Clip Studio Paint – 74 балів.

Аналіз за критеріями для анімаційних програмних застосунків був наведений в таблиці 3.2.

Таблиця 3.2 – Порівняння систем для створення анімації (таблиця виконана самостійно)

Критерії	Spriter Pro	Spine	Adobe After Effects
Інструментарій для анімації у грі(5)	Висока	Висока	Низька
Невибагливість до ресурсів системи (5)	Середня	Висока	Низька
Інтуїтивність інтерфейсу(3)	Середня	Висока	Низька
Орієнтованість під 2D скелетну анімацію(5)	Висока	Висока	Низька
Підтримка форматів(4)	Середня	Висока	Висока
Інтеграція з ігровими двигунами(5)	Середня	Висока	Низька

Після розрахунку даних аналізу, було отримано такі результати:

- Spriter Pro – 64 балів;
- Spine – 81 балів;
- Adobe After Effects – 35 балів.

У підсумку, Adobe може вражати своєю багатофункціональністю, але для розробки ігор вона не завжди відповідає потребам. Її широкий спектр можливостей ускладнює зрозумілість інтерфейсу, що робить її менш привабливою для цієї галузі. Натомість, програми, які були розглянуті, такі як Krita та Spine, відзначаються зручним інтерфейсом та необхідним функціоналом. Вони не вимагають потужного обладнання, підтримують різні формати файлів та мають весь необхідний інструментарій для розробки об'єктів для 2Д ігор.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

Створення ігрового програмного застосунку вимагає глибокого розуміння та володіння методами, які використовуються в розробці. Це не тільки дозволяє скоротити час на виконання завдань завдяки зручним інструментам, але й оптимізує різні процеси у грі. Важливо орієнтуватися у різноманітних програмах та знати, як вони можуть взаємодіяти між собою для досягнення кращих результатів. Крім того, володіння такими знаннями сприяє створенню більш стабільних та продуктивних застосунків, що відповідають сучасним стандартам індустрії ігор.

Робота над візуальними об'єктами гри є вкрай важливою, тому до неї необхідно підходити з максимальною відповідальністю. Створення спрайтів потребує зручного інтерфейсу та великого набору інструментарію, що дозволить розробляти більш якісні та унікальні зображення. Сучасні технології, такі як скелетна техніка, значно спрощують роботу з анімаціями, роблячи їх більш реалістичними та динамічними. Важливо також враховувати аспекти імпорту та експорту файлів, що забезпечують ефективну взаємодію між середовищами розробки. Для виконання цього важливого завдання були обрані дві програми: програма для створення ілюстрацій та спрайтів Krita та програма для створення анімацій Spine. Вони, порівняно з конкурентами, виконують свої обов'язки якнайкраще та відзначаються простий та зрозумілий UI і UX з передовим інструментарієм та технологіями.

4.1 Інструментарій Krita

Krita надає потужні інструменти для створення растрової ілюстрації. Вона пропонує широкий вибір пензлів та текстур, що дозволяє реалізувати найсміливіші ідеї.

4.1.1 Шари

Наявність шарів у програмах для створення графіки є фундаментальним інструментом. Розподіл елементів зображення на різні частини допомагає краще ними керувати, прибираючи щось зайве, або додаючи дещо нове. Однак це не єдина перевага шарів. Для впровадження текстур в анімації вкрай важливо розділяти їх на різні об'єкти та дотримуватися глибини. Даний інструмент активно використовувався абсолютно для всіх ілюстрацій (див. рис. 4.1).

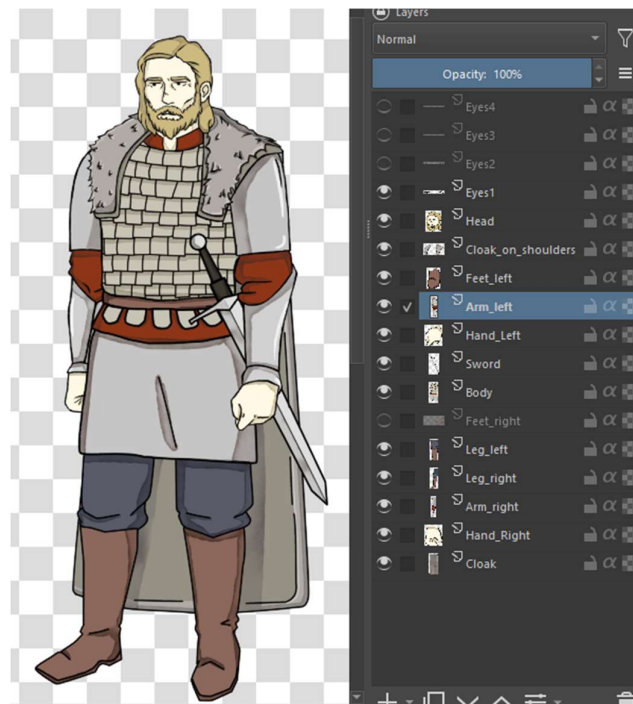


Рисунок 4.1 – Розбиття ілюстрації на декілька шарів (рисунок виконаний самостійно)

На прикладі було продемонстровано, як розміщується різні елементи головного героя за глибиною, а також в такому вигляді їх було імпортовано у Spine.

4.1.2 Використання фільтрів

Krita дозволяє використовувати фільтри до кожного окремого шару або до групи шарів. Дана функція має великий спектр застосувань: від додавання блюру

до створення карт нормалей. Під час малювання спайтів для ігрового програмного застосунку, використовувалися фільтри для коректування їх кольору (див. рис. 4.2).

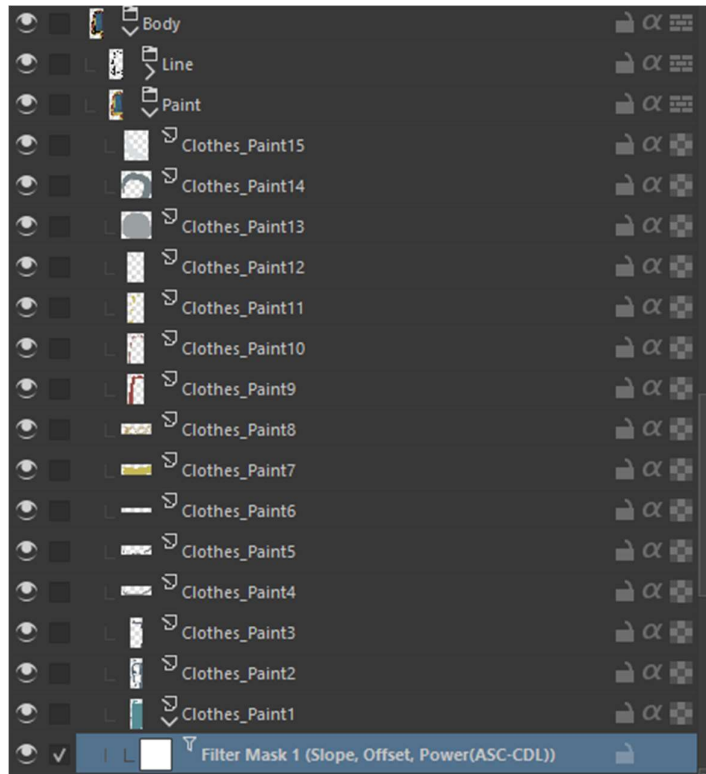


Рисунок 4.2 – Застосування фільтру до одягу персонажа (рисунок виконаний самостійно)

За для зміни кольору одягу був використаний такий інструмент, як ASC_CDЛ. Це стандартний спосіб опису метода кольорового балансу, який складається з трьох характеристик: Slope, Offset та Power (див. рис. 4.3).

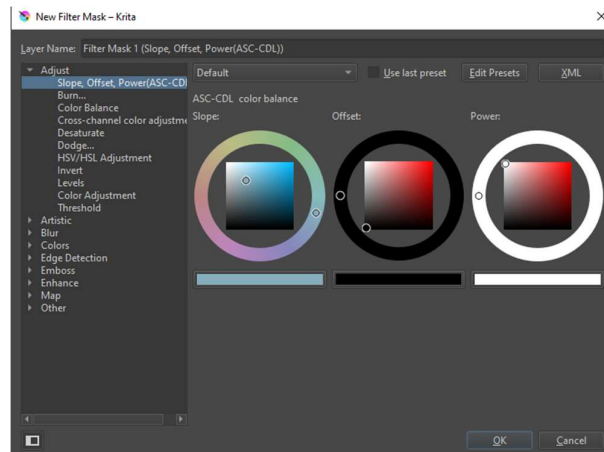


Рисунок 4.3 – Фільтр-маска ASC_CDЛ (рисунок виконаний самостійно)

Slope є множенням, що визначає налагодження більш яскравих кольорів зображення. В той же час, Offset визначає, наскільки нижня частина відносна гори, і, таким чином, отримується колір з найтемніших кольорів. Power є свого роду степеневою функцією та визначає налаштування середніх тонів до темних кольорів ілюстрації. Серед цих інструментів, був застосований тільки Slope для одягу князя Володимира, що змінило його колір з бірюзового на синій (див. рис. 4.4).



Рисунок 4.4 – Результат до та після застосування фільтру (рисунок виконаний самостійно)

Як результат, після використання ефекту персонаж отримав колір одягу, який гармонізує з іншими елементами. Таким чином, фільтри дозволять досить зручно підбирати потрібні відтінки, повільно перетягуючи курсор на колірній гамі, що значним чином вирішує проблему поєднання кольорів.

4.1.3 Створення власного пензля

Як раніше згадувалося, Krita орієнтована на створення растрових зображень, тому має величезний інструментарій для цього, включно із побудовою власного пензля. Незважаючи на те, що програма їх містить багато, для виконання вузькопрофільних завдань їх недостатньо. Для створення опудала ведмедя на рівні з тронним залом князя, був придуманий новий пензлик, який імітує форму хутра тварини (див. рис. 4.5).

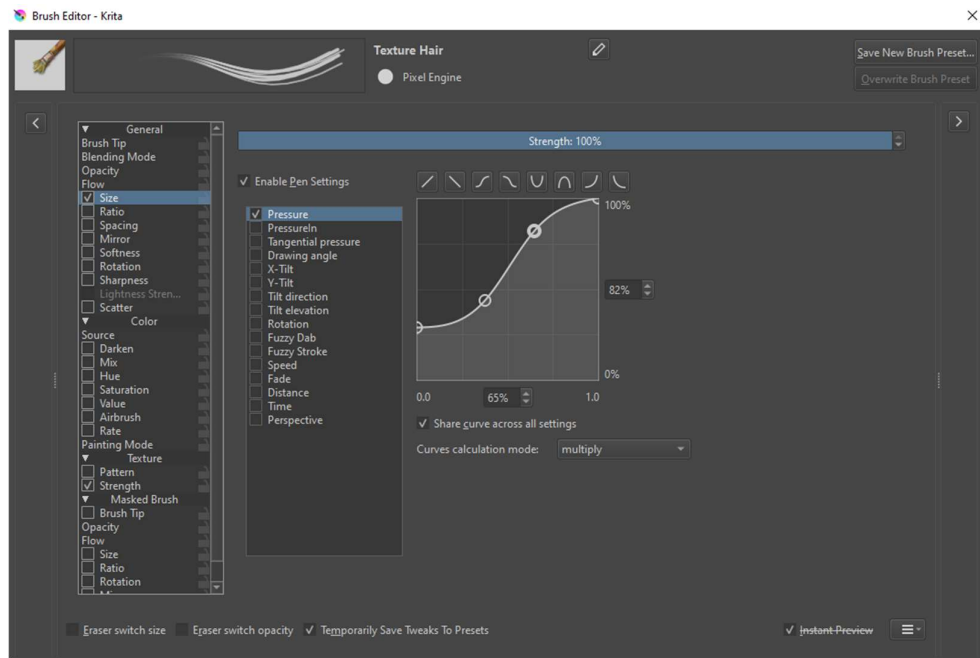


Рисунок 4.5 – Вікно налаштування пензля (рисунок виконаний самостійно)

За допомогою вбудованої у Krita текстури, яка має вигляд, як набір декількох крапок, можна одночасно робити багато мазків, а завдяки параметру Pressure, в якому при правильному налаштуванні, можна реалізувати, щоб лінія на кінці була більш тоншою, ніж на початку. Як результат, був створений спрайт опудала ведмеда, де було використано лише власноруч зроблений пензлик (див. рис. 4.6).



Рисунок 4.6 – Результат застосування нового пензля (рисунок виконаний самостійно)

Новий інструмент значно полегшив роботу над хутром, оскільки замість одного мазка, він будував декілька анатомічно правильних. Тому, створення нових засобів для малювання, може не тільки скоротити час, а й додати неповторну текстуру до об'єкту.

4.1.4 Інструментарій для побудови перспективи

При створенні будівель для ігрового програмного продукту, не знадобилося використання складної перспективи. Однак, все ж таки було необхідно дотримуватися чітких та рівних ліній. Для вирішення даного завдання, був використаний інструментарій Assistant Tool. Найнеобхіднішими функціями з нього стали горизонтальні та вертикальні Parallel Ruler для побудови перспективи будинків у селищі головного героя (див. рис. 4.7).

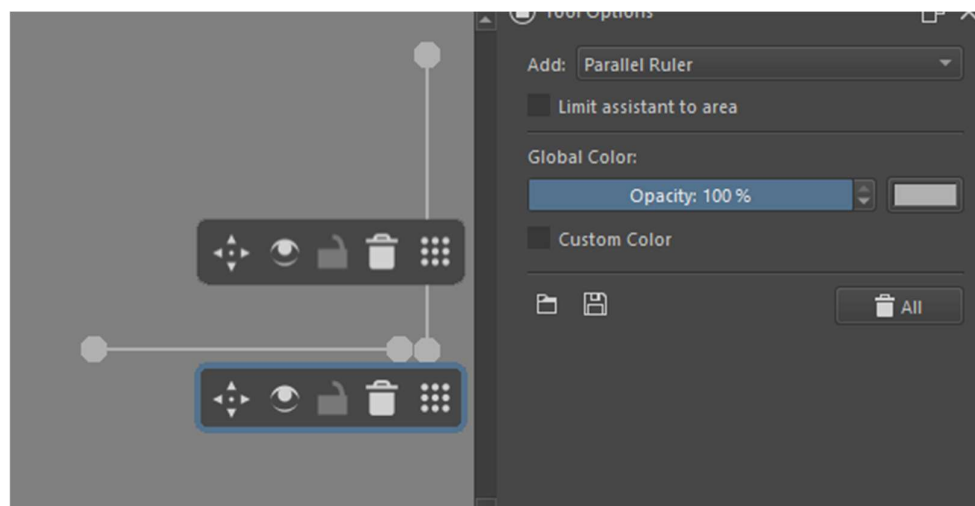


Рисунок 4.7 – Розміщення Parallel Ruler (рисунок виконаний самостійно)

За допомогою даних ліній, були створені хатинки, які мають рівні стіни та дахи (див. рис. 4.8).



Рисунок 4.8 – Результат використання Parallel Ruler на будівлях (рисунок виконаний самостійно)

Використання Assistant Tool гарантує, що перспектива буде побудована правильна. Завдяки цьому інструментарію, можна не тільки створювати чіткі об'єкти, а й додавати розмітку для майбутньої постанови усієї сцени.

4.1.5 Функція Wrap Around Mode

Найчастішим, що було потрібно створити для заднього плану ігрового програмного застосунку, це тайлову графіку. Вона є методом створення великих ілюстрацій, застосовуючи маленькі фрагменти однакового розміру. Оскільки рівні у грі для кваліфікаційної роботи є довгими, то було необхідно створити безшовні тайлові текстури, які можна було б повторювати. Та домогтись того, щоб кінець зображення одночасно був його початком з інших сторін. На допомогу, прийшов інструмент Wrap Around Mode, який робить полотно необмеженим не тільки з усіх кінців, а також тільки по горизонталі або по вертикалі (див. рис. 4.9).

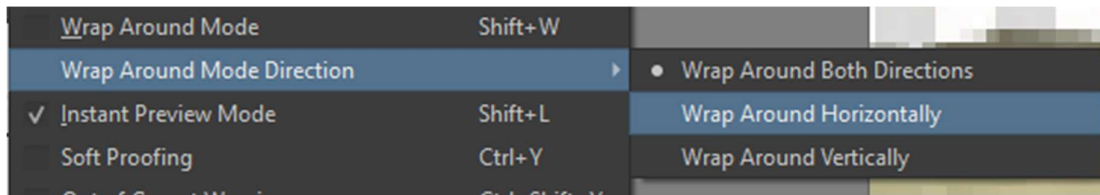


Рисунок 4.9 – Функція Wrap Around Mode (рисунок виконаний самостійно)

Завдяки даному інструментарію, було створено багато спрайтів, включаючи небо, землю та тропи (див. рис. 4.10).

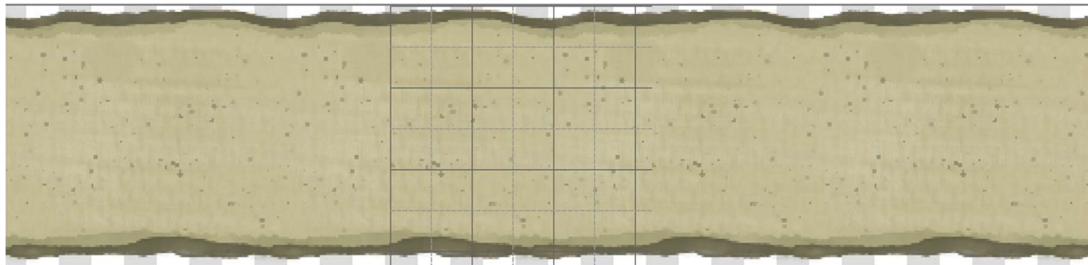


Рисунок 4.10 – Результат використання Wrap Around Mode (рисунок виконаний самостійно)

Як результат, були створені текстури для заднього фону, які можна нескінченно поєднувати між собою без візуальних переривань між ними. Функція Wrap Around Mode стала найціннішим помічником серед усього інструментарію Krita, яка не тільки пришвидшувала роботу, але й стала просто незамінною у даному питанні.

4.1.6 Інструментарій для анімацій

Основна робота з анімацією була адресована програмі Spine, але були спроектовані такі дії, які було б неможливо створити за допомогою скелету, бо втрачалася би їх природність. Таким винятком стала блискавка. Її довелося малювати покадрово, тому що кожен елемент на кожному кадрі повинен був бути індивідуальним. Ця особливість також обумовлена тим, що дане стихійне явище є надзвичайно швидким. Людина не може своїми очима вловити увесь поетапний цикл удару блискавки. Тому вона просто не може бути плавною, що якраз забезпечує скелетна анімація.

Для створення удару блискавки був використаний інструментарій для анімації у Krita (див. рис. 4.11).

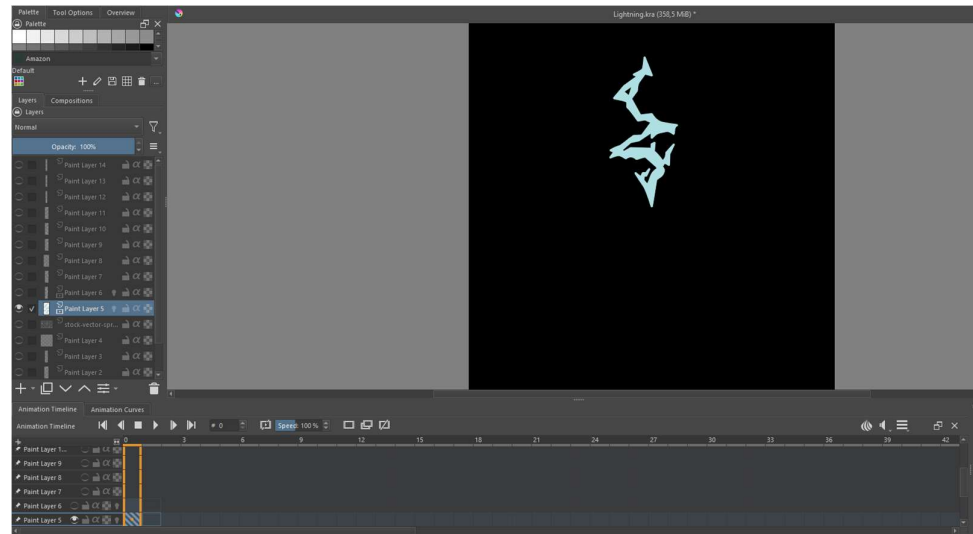


Рисунок 4.11 – Панель інструментів для створення анімацій у Krita (рисунок виконаний самостійно)

Він ідеально підійшов для створення простих зображень та їх покадрової демонстрації. Подібний результат, ще раз підтверджує те, що програма для створення та редагування графіки Krita, чудово підходить для роботи зі спрайтами, які будуть інтегровані у ігрові програмні застосунки.

4.2 Інструментарій Spine

Анімація є важливим елементом будь-якої гри, що дозволяє вдихнути життя в ілюстрації. Вона необхідна, як для виконання основних завдань (Наприклад, додавання рухів головному герою), так і для створення атмосфери, яку розробник бажає передати. В залежності від того, на скільки якісно буде створена та чи інша анімація, гравець отримає позитивний або негативний ігровий досвід. Світ повинен бути не тільки добре зображеним, але й мати певні дії, які зроблять його «живим». З цим чудово впорався програмний застосунок Spine, створивши для кожного об'єкту скелет та подарувавши їм рухи.

4.2.1 Використання скелетної анімації

Скелетна анімація першочергово замінила класичне перемальовування об'єкта на кожному кадрі. Створення рухів стало можливим завдяки побудові скелету, яка має деревоподібну структуру та ієрархічні відносини між кістками. Кожна кістка має зв'язок з попередньою, через що повторює за нею рухи. Вона використовувалася для створення майже всіх дій: від анімації головного героя до анімації туману. Однак, для того щоб зробити можливим трансформацію текстур з кістками, необхідно налаштувати взаємозв'язок між ними.

4.2.1.1 Робота з Mesh

Mesh або сітка – це компонент (поверхня моделі), яка складається з полігонів та їх вершин. Завдяки їй, кожна текстура «розуміє», як саме вона буде модифікуватися під впливом руху скелету (див. рис. 4.12).



Рисунок 4.12 – Створення сітки на текстурі голови головного персонажа (рисунок виконаний самостійно)

На прикладі головного героя, було проілюстровано створення індивідуального Mesh для кожної його текстури. Дана сітка повторює контури моделі, для того щоб модифікації проходила анатомічно правильно.

Наступним кроком, були створені взаємозв'язки між текстурами та кістками. Програма Spine автоматично налаштовує, в якій мірі кожні вершини будуть залежати від тієї чи іншої кістки. Однак, іноді вони потребують коректування. Для цього використовується такий інструмент, як Weights. Ваги потрібні для прив'язки вершин до кісток, що дозволяє скелету впливати на сітку (див. рис. 4.13).

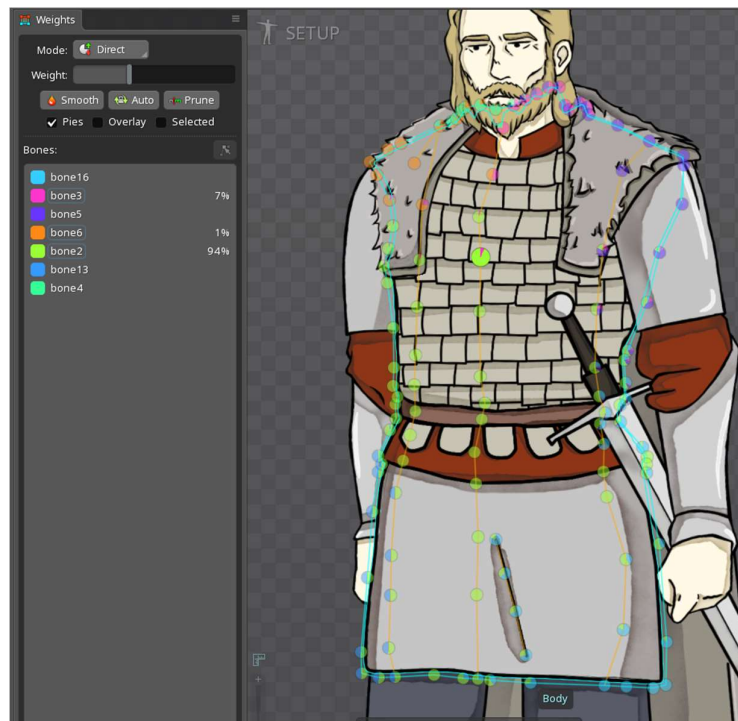


Рисунок 4.13 – Редагування взаємозв'язку вершин та кісток за допомогою Weights (рисунок виконаний самостійно)

Як було проілюстровано, кожна вершина змінюється під впливом однієї кістки або множини кісток в залежності від відсотку впливу.

4.2.1.2 Обмеження для кісток

Spine має три функції обмеження для кісток скелету, які є досить корисними при роботі з різними видами задач. Деякі з них є опціональними, а без інших майже не обходиться ні одна анімація. А саме це:

- IK Constraint [18];
- Transform Constraint [19];
- Path Constraint [20].

IK Constraint встановлює обертання таким чином, щоб кінчик кістки доторкався до цільової кістки або вказував на неї. Дана функція має багато застосувань, але найчастіше вона використовується для руху рук та ніг (див. рис. 4.14).

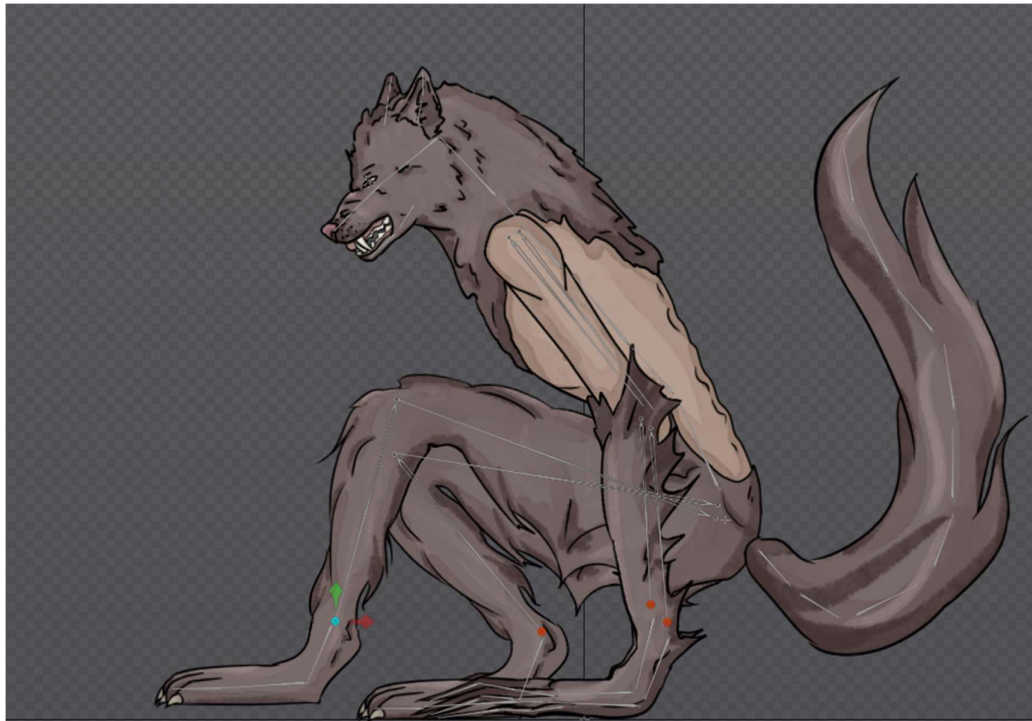


Рисунок 4.14 – IK Constraint пов’язана з двома кістками однієї ноги (рисунок виконаний самостійно)

Завдяки даному обмеженню є можливість анатомічно правильно пересувати ноги та руки Волкодлака або інших персонажів за допомогою однієї точки без застосування по черзі обертання кожної окремої кістки. Якщо в певний момент часу не потрібне використання IK Constraint, програма дозволяє змінити значення Міх на нульовий показник, що скасує дію обмеження (див. рис. 4.15).

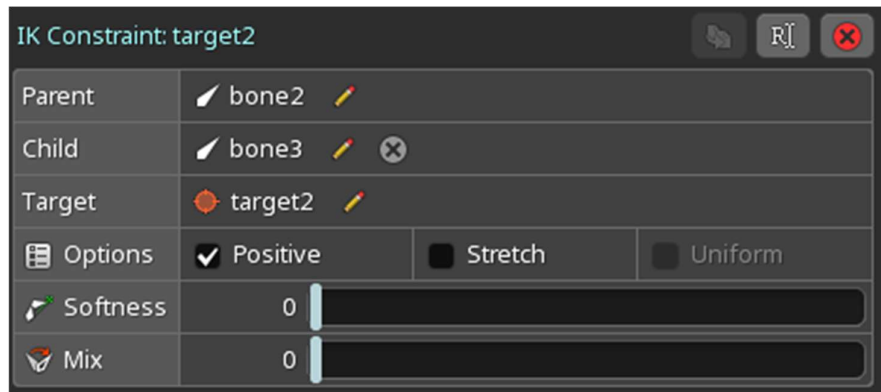


Рисунок 4.15 – Деактивація IK Constraint (рисунок виконаний самостійно)

IK Constraint це основне обмеження, яке застосується майже у кожному скелеті до кісток. Ця функція неабияк облегшує роботу, допомагаючи уникати використання зайвого обернення.

Transform Constraint має велику множину застосувань, але у випадку даного ігрового продукту була використана для створення взаємозв'язку між зброєю та рукою персонажа. Дана функція дозволяє переміщувати кістку разом з іншою за рахунок копіювання дій батьківської кістки (див. рис. 4.16).

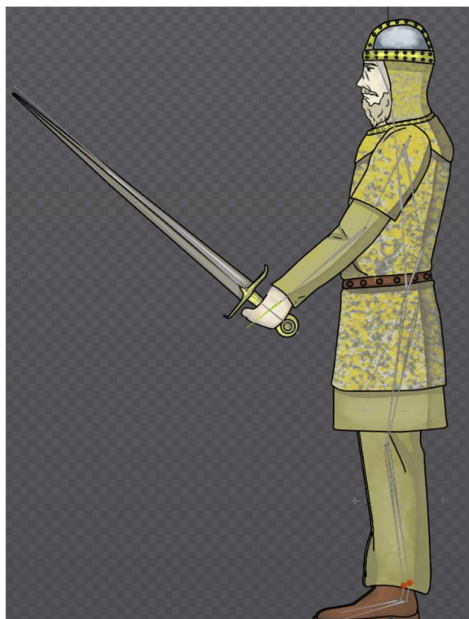


Рисунок 4.16 – Використання Transform Constraint для прив'язки меча до руки персонажа (рисунок виконаний самостійно)

Теоретично, можна було б просто при створенні скелета зробити так, щоб кістка меча була за ієрархією пов'язана з кистю руки. Однак, якщо б у майбутньому знадобилось покласти зброю у піхви, або жбурнути його, то не було б можливості це зробити. Тільки Transform Constraint дозволяє, коли потрібно зв'язувати та роз'єднувати кістки за допомогою параметра Mix (див. рис. 4.17).

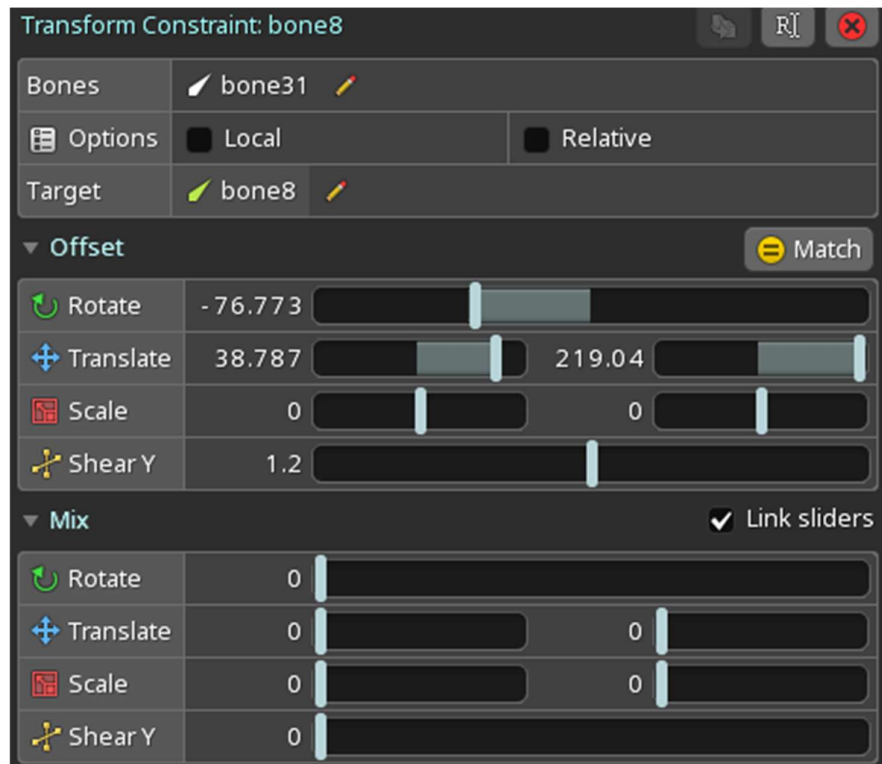


Рисунок 4.17 – Деактивація Transform Constraint (рисунок виконаний самостійно)

Якщо встановити значення Mix нульовим для всіх видів взаємодії, то кістка зброї від'єднається від батьківської та зможе переміщуватися незалежно.

Path Constraint був використаний для руху хребта, при створенні магії з колекційної карти, яка генерує атакуючого скелета (див. рис. 4.18).

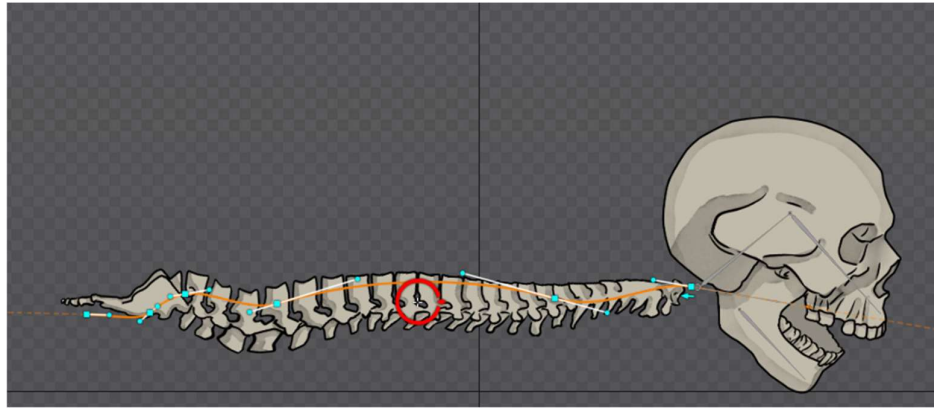


Рисунок 4.18 – Використання Path Constraint для руху хребта (рисунок виконаний самостійно)

Обмеження шляху коригує рух кісток за певною траєкторією. Хоча ця функція не є дуже популярною у використанні, вона ідеально підходить для того, щоб зробити рухи скелета плавними та усунути необхідність вручну керувати кожною окремою кісткою, особливо коли їх багато. Це особливо корисно в ситуаціях, коли певна дія повинна повторюватися у кількох анімаціях. Завдяки Path Constraint можна забезпечити, щоб скелет виконував ці дії точно та послідовно, зберігаючи при цьому високу якість і природність рухів.

4.2.2 Використання Skins

Skins це надзвичайно корисна функція у Spine, яка дозволяє змінювати текстури на одному й тому ж самому скелеті. Її можна використовувати для простої заміни нарядів одного персонажа або для збирання цілих нових героїв по частинам. Завдяки Skins можна не створювати знову скелет та анімації для подібних об'єктів, а просто замінити оболонки. Даний інструмент був використаний під час роботи над NPC пішоходами (див. рис. 4.19).



Рисунок 4.19 – Використання Skins для зміни вигляду подібних персонажів
(рисунок виконаний самостійно)

Оскільки всі пішоходи мають однакову анімацію ходьби, були створені дві оболонки: чоловіча та жіноча. Наступним кроком, був побудований скелет, який анатомічно підходить для обох персонажів. Завдяки вибору відповідного скіна, можна отримати унікального героя з власними текстурами. Функція Skins дозволила створити двох різних персонажів з однаковим скелетом і анімаціями, що вдвічі спростило роботу над ними.

4.2.3 Робота з анімаціями

Після завершення необхідної підготовки, а саме створення скелета та налаштування його взаємодії з текстурами, наступним етапом є безпосереднє створення анімації. У програмі Spine цей процес відбувається досить легко: використовується доріжка кадрів, де кожна кістка синхронізується з нею за допомогою ключів [21]. Ключі визначають час початку та завершення, а також значення переходу. Анімація скелета зводиться до встановлення поз, які задаються ключами на часовій шкалі.

4.2.3.1 Використання Offset

При роботі з анімаціями іноді доводиться рухати ключі на часовій шкалі. Однак, вона ускладнюється, коли необхідно перемістити не один ключ а декілька. Для цього ідеально підходить такий інструмент, як Offset.

Кнопка Offset використовується для переміщення ключів у анімації, яка має однаковий кінець та початок (зациклена) (див. рис. 4.20).

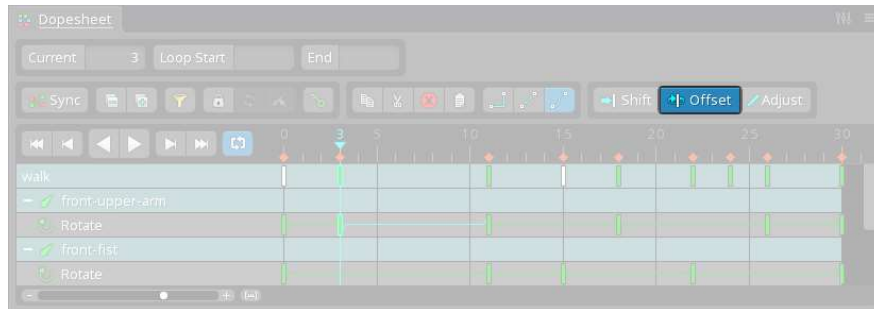


Рисунок 4.20 – Кнопка Offset (за даними [21])

Вона переміщає ключі у даному проміжку, не дозволяючи їм виходити за межі певного часу. Інструмент Offset використовувався для побудови анімації плаща головного героя (див. рис. 4.21).



Рисунок 4.21 – Використання Offset на плащі персонажа (рисунок виконаний самостійно)

Текстура плаща головного героя прив'язана до шести кісток, і для створення реалістичного руху необхідно, щоб тканина розвивалася послідовно: спочатку нижня частина, потім верхня. Для досягнення цього ефекту потрібно, щоб кожна кістка рухалася одна за одною та у певному порядку на шкалі часу. Offset перемістив кожний ключ у межах циклічної анімації, забезпечуючи послідовний і природний рух плаща.

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування ігрового забезпечення є наймовірно важливим етапом під час розробки гри, який допомагає забезпечити її якість, стабільність і привабливість для гравців. На цьому етапі виконується пошук та виправлення помилок, перевірка продуктивності, оцінка її зручності. Головною метою тестування є впевнитися, що ігровий програмний застосунок працює безперебійно на різних пристроях і платформах, а також що всі ігрові механіки працюють так, як було задумано. Також виконується перевірка, чи відповідає програмне забезпечення технічним і дизайнерським вимогам. Завдяки ретельному тестуванню можна уникнути негативного досвіду гравців і випустити на ринок якісний продукт.

5.1 Галузь застосування

Під час планування були виділені основні обласні, над якими було проведене тестування:

- інтерфейс гри;
- програвання анімацій;
- геймплей;
- ігрові механіки;
- інсталяція гри.

Ці основні частини ігрового програмного застосунку мають працювати злагоджено та безперебійно, щоб гравці могли насолоджуватися грою без будь-яких проблем. Тестування інтерфейсу гри включає перевірку зручності та інтуїтивності користування, що впливає на загальне враження від неї. Програвання анімацій має бути плавним і відповідати заданим параметрам, щоб забезпечити реалістичність і привабливість візуальних ефектів. Геймплей тестується на предмет відсутності багів і логічних помилок, що можуть зіпсувати ігровий процес. Ігрові механіки повинні працювати без збоїв, забезпечуючи належну взаємодію всіх елементів гри. Інсталяція повинна бути швидкою та простою, щоб користувачі могли легко встановити її на свої пристрої без технічних труднощів. Таким чином,

ретельне тестування кожної з цих областей є критично важливим для досягнення високої якості кінцевого продукту.

5.2 Типи тестування

У сучасній системі тестування програмних продуктів існує безліч різних типів тестування, проте не всі вони є універсальними. Тому, для даного ігрового застосунку було виділено шість відповідних видів:

- тестування інсталяції;
- функціональне тестування;
- тестування інтерфейсу користувача;
- тестування відмовостійкості та відновлення;
- тестування продуктивності;
- тестування конфігурації.

Отже, для забезпечення високої якості ігрового застосунку були обрані дані види тестування. Ці типи охоплюють всі критичні аспекти гри, від інсталяції та функціональності до продуктивності та конфігурації. Такий підхід гарантує, що гра буде стабільною, зручною у використанні та відповідатиме всім технічним вимогам, що сприятиме позитивному досвіду користувачів.

5.2.1 Тестування інсталяції

Метою даних випробувань вважається перевірка, що інсталяція, деінсталяція, а також повторна інсталяція пройшла успішно і безперебійно на персональний комп'ютер та смартфон.

Під час тестування інсталяції повинні бути пророблені певні технічні прийоми, такі як:

- запустити інсталятор ігрового програмного застосунку та перевірити, чи відбувається коректний запуск інсталяційного процесу;
- перевірити, чи є можливість встановлювати додаткові налаштування під час інсталяції, такі як вибір директорії, створення ярликів тощо;

- впевнитися, що процес завантаження ресурсів гри (графіка, звук, тощо) пройшов без помилок та інтерференцій;
- виконати відкриття гри та впевнитися, що вона правильно інтегрується з операційною системою і може запускатися безпроблемно після інсталяції;
- видалити гру з персонального комп'ютера або смартфона та перевірити, чи була деінсталяція коректна;
- запустити інсталятор та встановити гру знову. Повторно перевірити роботу додатку після завантаження.

Успішними критеріями завершення тестування інсталяції можна вважати:

- повноцінне завершення процесу інсталяції без повідомлень про помилки або проблем;
- ігровий програмний застосунок запускається і працює коректно після завершення інсталяції;
- відсутність конфліктів або проблем із завантаженням ресурсів під час гри.

5.2.2 Функціональне тестування

Ціллю даного тестування є впевнитися, що функціональні аспекти ігрового програмного застосунку працюють правильно та відповідають вимогам та очікуванням користувачів.

Його технічними способами можна вважати проведення тестування функціоналу таких розділів ігрового додатка, як:

- головне меню гри;
- збереження та завантаження;
- проходження рівнів;
- управління головним персонажем;
- взаємодія з оточенням.

Функціональне тестування можна вважати завершеним, якщо:

- ігровий програмний застосунок працює безконфліктно та безперебійно на різних етапах;

- усі елементи меню та інтерфейсу реагують коректно на дії користувача;
- геймплей відповідає очікуванням;
- функції збереження та завантаження працюють без помилок.

5.2.3 Тестування інтерфейсу користувача

Метою тестування інтерфейсу є переконатися, що UI та UX є інтуїтивно зрозумілими, ефективними, відповідають дизайну та вимогам ігрового програмного застосунку.

До технічних прийомів можна віднести:

- перевірка ефективності навігації в грі та переконання у тому, що всі елементи меню працюють правильно та є доступними;
- тестування різних типів введення, таких як миша, клавіатура та контролери, і отримання результату, де всі функції відповідають на введення користувача;
- оцінка графічного дизайну та анімацій, які повинні покращувати користувацький досвід під час проходження гри.

Успішним завершенням тестування інтерфейсу користувача можна вважати, коли:

- навігація в грі є інтуїтивно зрозумілою та легкою для користувача;
- усі елементи меню та кнопки реагують коректно на введення;
- графічний дизайн та анімації покращують загальний користувацький досвід.

5.2.4 Тестування відмовостійкості та відновлення

Можна вважати, що ціллю випробування є визначення, як програмний застосунок веде себе в умовах відмов та впевнення, що вона може відновлювати свою працездатність після непередбачених ситуацій.

До технічних прийомів даного тестування можна віднести:

- змушений виклик внутрішньої помилки (наприклад, помилки

завантаження ресурсів або помилки обробки даних) та перевірка, як ігровий програмний застосунок їх виявляє та відновлюється;

- забезпечення того, щоб програма коректно реагувала на відмови апаратних або програмних компонентів, таких як відключення пристрою введення, зменшення ресурсів пристрою тощо;
- закриття додатку або вимкнення системи під час гри, для того щоб переконатися, що програмний застосунок може відновлюватися та завантажувати гравця з останнього збереженого стану;
- створення великих файлів збереження, для того щоб впевнитися, що гра ефективно обробляє ці дані без втрати продуктивності.

Отже, критеріями завершення тестування можна вважати:

- ігровий програмний застосунок коректно виявляє та обробляє внутрішні помилки, надаючи користувачеві зрозумілі повідомлення;
- стійко та правильно поводить себе під час відключення окремих компонентів апаратного забезпечення або програмних помилок;
- відновлення гри після аварій або закриття протікає без проблем та веде гравця до останнього збереженого стану;
- ефективна обробка та завантаження великих обсягів даних без втрати продуктивності.

5.2.5 Тестування продуктивності

Метою даного тестування є оцінка продуктивності ігрового програмного продукту у різних умовах навантаження, а також забезпечення плавного та ефективного геймплею.

Можна виділити три технічних способи, які були використані під час тестування продуктивності:

- визначення швидкості роботи гри на різних конфігураціях ПК та впевнення у тому, що фреймрейт є прийнятним навіть на менш потужних системах;

- виконання тестування завантаження ресурсів гри, зокрема, отримання часу відтворення рівнів та текстур, щоб уникнути довгих затримок;
- проведення тестування продуктивності під великою кількістю одночасних подій, таких як масова взаємодія об'єктів на екрані.

Були відзначені оптимальні результати, які повинні бути після проведення тестування. А саме це отримання стабільного фреймрейту та високої якості графіки на різних конфігураціях персонального комп'ютера та смартфона, прийнятного часу завантаження рівнів та текстур, стабільного геймплею без великих затримок, стійкої продуктивності гри під високими навантаженнями.

5.2.6 Тестування конфігурації

Тестування конфігурації допомагає перевіряти, як ігровий програмний застосунок веде себе на різних конфігураціях обладнання та операційних систем, а також визначити його сумісність з різними налаштуваннями.

Були встановлені технічні прийоми, які визначили, як саме буде проходити тестування:

- встановлення та запуск програмного продукту на різних операційних системах, включаючи Windows 7, 8, 10, 11 Linux та MacOS, Android, Web та переконання, що гра коректно працює на кожній з них;
- перевірка сумісності ігрового програмного застосунку з різними конфігураціями персонального комп'ютера та смартфона, включаючи відмінності в обладнанні (графічні карти, процесори, обсяг оперативної пам'яті тощо);
- перевірка адаптування додатка до різних роздільних здатностей екранів та роботи в повноекранному та віконному режимах.

Як результат, були прописані критерії завершення даного виду тестування:

- ігровий програмний застосунок коректно запускається та працює на різних версіях операційних систем;
- задекларовані мінімальні та рекомендовані системні вимоги виконуються;

- наявність коректної адаптації до різних роздільних здатностей екранів та режимів відображення;
- успішна робота з різними конфігураціями обладнання.

5.3 Етапи тестування

Для кожного з видів тестування має бути відведений певний час, щоб гарантувати своєчасне завершення цього етапу розробки ігрового програмного продукту. Обсяг робіт і дати виконання для кожного цільового завдання надано в таблиці 5.1.

Таблиця 5.1 – План тестування ігрового програмного застосунку (таблиця виконана самостійно)

Цільове завдання	Обсяг робіт	Дата початку	Дата закінчення
План випробувань	1 день	14.05.2024	15.05.2024
Функціональне тестування	4 дні	15.05.2024	19.05.2024
Тестування інтерфейсу користувача	1 день	19.05.2024	20.05.2024
Тестування відмовостійкості та відновлення	2 дні	20.05.2024	22.05.2024
Тестування продуктивності	3 дні	22.05.2024	25.05.2024
Тестування конфігурації	3 дні	25.05.2024	28.05.2024
Тестування інсталяції	2 дні	28.05.2024	30.05.2024

Таким чином, для кожного виду тестування було виділено визначений час, щоб забезпечити своєчасне завершення цього етапу розробки ігрового програмного продукту. Відповідно до плану, кожне цільове завдання має чітко встановлені обсяги робіт і дати виконання. Це дозволяє організувати процес ефективно та гарантує високу якість кінцевого продукту.

5.4 Баг-репорт

Під час створення будь-якого програмного застосунку виникає той чи інший ряд помилок. Вони можуть бути пов'язані, як із логікою гри, так і з імпортом сторонніх ресурсів. Для ефективного виправлення багів усі виявлені помилки були систематизовані та докладно задокументовані:

- а) магічна сфера вилітає раніше, ніж відтворюється анімація головного героя:
 - 1) короткий опис: Після застосування магії на навчальних рівнях сфера (магія) вилітає набагато швидше, ніж головний герой піднімає руки для її створення;
 - 2) компонент програми: Клієнтська/front-end частина
 - 3) серйозність: Тривіальний;
 - 4) пріоритет: Низький;
 - 5) кроки відтворення;
 - підібрати колекційну карту здібностей на навчальному рівні;
 - натиснути на клавішу «1» на клавіатурі;
 - 6) фактичний результат: Сфера з'являється занадто рано, як тільки головний герой починає програвати анімацію (піднімати руки);
 - 7) очікуваний результат: Сфера з'являється тільки тоді, коли анімація головного героя стане відображати підняті прямо руки;
- б) при зміні будь-якої анімації на анімацію idle персонаж дивиться у протилежний бік;
 - 1) короткий опис: Після виконання будь-якої дії головний герой

переходить у стан очікування. Однак його напрям(бік, в який він був обернений) не співпадає з напрямом попередньої дії;

- 2) компонент програми: Клієнтська/front-end частина;
 - 3) серйозність: Незначний;
 - 4) пріоритет: Низький;
 - 5) кроки відтворення;
 - виконати переміщення вліво;
 - зупинитися;
 - виконати переміщення вправо;
 - зупинитися;
 - 6) фактичний результат: При переході з будь-якої анімації на анімацію idle персонаж завжди дивиться у протилежний бік;
 - 7) очікуваний результат: При зміні будь-якої анімації на анімацію idle персонаж дивиться у той бік, в який дивився перед цим;
- в) анімація idle головного героя виконується зі збоєм;
- 1) короткий опис: Під час програвання анімації idle головного героя вона відображається “зламано”. Анімація відрізняється від тієї, яка була створена у програмі Spine;
 - 2) компонент програми: Клієнтська/front-end частина;
 - 3) серйозність: Незначний;
 - 4) пріоритет: Низький;
 - 5) кроки відтворення;
 - запустити гру;
 - виконати дії переміщення та зупинитися;
 - 6) фактичний результат: Анімація idle головного героя не відповідає тій, яка була створена. Має «зламане» різке відображення;
 - 7) очікуваний результат: Анімація у грі відповідає створеній в програмі Spine. Виглядає плавною та естетичною;
- г) Відсутність зміни анімації удару на анімацію ходьби;
- 1) короткий опис: Коли головний герой програє анімацію ходьби, а потім

виконує удар, він не перестає переміщатися у просторі, а анімація нанесення урону повторюється нескінченно, допоки гравець не почне робити щось інше;

- 2) компонент програми: Клієнтська/front-end частина;
 - 3) серйозність: Значний;
 - 4) пріоритет: Середній;
 - 5) кроки відтворення;
 - натиснути на клавішу «D» або «A», щоб почати переміщатися;
 - натиснути ліву клавішу миші;
 - перестати натискати клавіши «D» або «A»;
 - 6) фактичний результат: Головний герой переміщується у просторі під час удару, а також анімація атаки програється нескінченно;
 - 7) очікуваний результат: Після натискання на ліву кнопку миші персонаж повинен перестати переміщатися у просторі і виконати удар один раз. Якщо клавіши «D» або «A» не були відпущені, то після удару головний герой повинен продовжити свій рух. Якщо ж клавіші переміщення не активовані, то після нанесення удару головний герой повинен переходити в стан очікування;
- д) під час ходьби головний персонаж сходить з доріг;
- 1) короткий опис: Під час переміщення головний герой повинен рухатися по дорозі. Оскільки стежка не є однакового розміру по всій площі локації, гравець сходить з неї;
 - 2) компонент програми: Клієнтська/front-end частина;
 - 3) серйозність: Незначний;
 - 4) пріоритет: Середній;
 - 5) кроки відтворення;
 - запуск першої або другої локації під час навчання на початку гри;
 - переміщення головного героя по цим локаціям;
 - 6) фактичний результат: Головний герой сходить з доріжки;
 - 7) очікуваний результат: Адаптація переміщення головного героя по

стежці: там, де дорога йде вниз, персонаж також спускається, де дорога йде вгору - піднімається;

- е) магія не відтворюється при подвійному натисканні;
- 1) короткий опис: Під час використання мани у головного героя програється відповідна анімація, а також відтворюється магія. При повторному застосуванні цієї здібності з'являється тільки магія без анімації персонажа;
 - 2) компонент програми: Клієнтська/front-end частина;
 - 3) серйозність: Незначний;
 - 4) пріоритет: Середній;
 - 5) кроки відтворення;
 - отримати колекційну карту на першому рівні;
 - натиснути на клавішу «1»;
 - повторно натискати на клавішу «1» до того моменту, як закінчиться мана;
 - 6) фактичний результат: Анімація не програється після появи магії;
 - 7) очікуваний результат: Анімація відтворення магії головного героя програється при кожному натисканні відповідної клавіші;
- ж) анімація стрибка повністю не відтворюється;
- 1) короткий опис: Анімація стрибка програється тільки у тому випадку, якщо затримати необхідну клавішу. В залежності від тривалості зажиму цієї кнопки, стільки часу і буде програватися дана анімація;
 - 2) компонент програми: Клієнтська/front-end частина;
 - 3) серйозність: Тривіальний;
 - 4) пріоритет: Низький;
 - 5) кроки відтворення;
 - натиснути на клавішу «W» або пробіл;
 - швидко відпустити дані клавіші;
 - затримати кнопки на більш довгий час;
 - 6) фактичний результат: При одноразовому натисканні на клавіші «W»

або пробіл(без затримки) у головного героя анімація стрибка програється не повністю;

7) очікуваний результат: Анімація стрибка програється повністю при одноразовому натисканні без затримки клавіш «W» або пробілу;

з) на початку рівня з будинком, головний герой падає з однієї висоти на іншу;

1) короткий опис: При переході на рівень з будинком, головний персонаж на початку підстрибує (а точніше ніби падає);

2) компонент програми: Клієнтська/front-end частина;

3) серйозність: Тривіальний;

4) пріоритет: Низький;

5) кроки відтворення;

– перехід за сюжетом на локацію з хатою головного героя;

– перегляд усього діалогу;

6) фактичний результат: Герой падає після закінчення діалогу;

7) очікуваний результат: Герой залишається на тому ж місці після перегляду діалогу на локації з будинком;

и) при відтворенні удару один раз, персонаж буде програвати цю анімацію нескінченно;

1) короткий опис: Коли гравець виконує нанесення урону, головний герой відтворює цю анімацію не посилаючись до кількості натискання відповідної клавіші, а нескінченно. При цьому урон іншим персонажам також постійно наноситься;

2) компонент програми: Клієнтська/front-end частина;

3) серйозність: Критичний;

4) пріоритет: Високий;

5) кроки відтворення;

– підійти до ворожого NPC;

– натиснути ліву клавішу миші;

6) фактичний результат: Гравець атакує та персонаж відтворює анімацію

нескінченно до її відміни іншою анімацією;

7) очікуваний результат: Гравець наносить удар та персонаж відтворює анімацію стільки разів, скільки гравець натиснув відповідну клавішу.

Анімація програється одна за одною, без переривання попередньої;

к) некоректне з'єднання спрайтів дороги;

1) короткий опис: Спрайти не є безшовними, тому їх з'єднання між собою на сцені не плавне та помітне;

2) компонент програми: Клієнтська/front-end частина;

3) серйозність: Тривіальний;

4) пріоритет: Низький;

5) кроки відтворення;

– перейти на локацію з селищем;

– переміщати головного персонажа вліво або вправо;

6) фактичний результат: Наявність стиків спрайтів головної стежки;

7) очікуваний результат: Спрайти дороги та землі поєднані між собою плавно та непомітно, шляхом створення безшовних аналогів.

6 ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Впровадження ігрового програмного застосунку є завершальним і надзвичайно важливим етапом у циклі розробки продукту. У цьому розділі розглядаються процеси, пов'язані з підготовкою до релізу, налаштуванням програмного забезпечення для кінцевих користувачів, а також заходи щодо забезпечення його надійної роботи після випуску. Впровадження включає фінальні перевірки, установку, налаштування та моніторинг функціонування гри на різних платформах і пристроях. Крім того, важливо забезпечити підтримку користувачів та виправлення потенційних проблем, що можуть виникнути після запуску. Це гарантує, що кінцевий продукт буде відповідати високим стандартам якості та задовольняти потреби гравців.

6.1 Просування у соціальних мережах

Для привернення уваги до ігрового програмного застосунку були створені акаунти у чотирьох популярних соціальних мережах, таких як Facebook, Instagram, Telegram та X (у минулому Twitter). На цих платформах періодично створювалися пости, які містили інформацію про сюжет гри, головних персонажах, механіках, а також процес розробки. Це допомогло створити спільноту зацікавлених гравців та підтримувати їх інтерес до гри. Активна присутність у соціальних мережах також дозволила швидко реагувати на відгуки користувачів і враховувати їхні побажання під час фінальної стадії розробки.

Facebook став важливою платформою для просування ігрового застосунку, слугуючи офіційним джерелом новин та анонсів (див. рис. 6.1).



Рисунок 6.1 – Заголовок профілю у Facebook

Завдяки своїй репутації як платформи для серйозного та офіційного контенту, Facebook допоміг залучити аудиторію, яка шукає глибокі та детальні відомості про гру. Хоча активність на ній була нижчою порівняно з іншими соціальними мережами, вона все ж відіграє ключову роль у створенні професійного іміджу ігрового програмного застосунку та підтримці зв'язку з гравцями, які цінують офіційні оновлення та новини.

Акаунт у соціальній мережі Instagram став однією з ключових платформ для взаємодії з аудиторією (див. рис. 6.2).

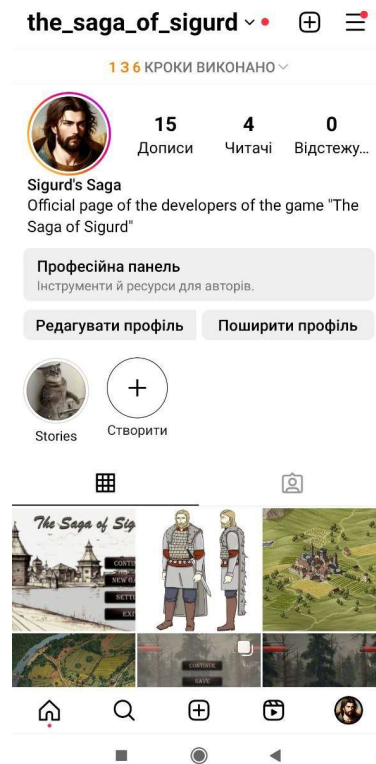


Рисунок 6.2 – Акаунт гри у Instagram

На сторінці наявні пости, що висвітлюють процес розробки програмного продукту, а також закріплені сторіс, де активно розміщувалися опитування, анонси та відео з процесами створення гри. Instagram дозволив демонструвати візуальні аспекти, залучаючи користувачів через інтерактивний контент, та отримувати зворотний зв'язок у режимі реального часу. Завдяки яскравим і привабливим публікаціям, було створено тісний зв'язок з підписниками та можливість підтримувати їхній інтерес до гри на всіх етапах її розробки.

Telegram став найпопулярнішою та найзручнішою соціальною мережею серед аудиторії ігрового застосунку, завдяки чому було отримано найбільшу активність саме в ній (див. рис. 6.3).

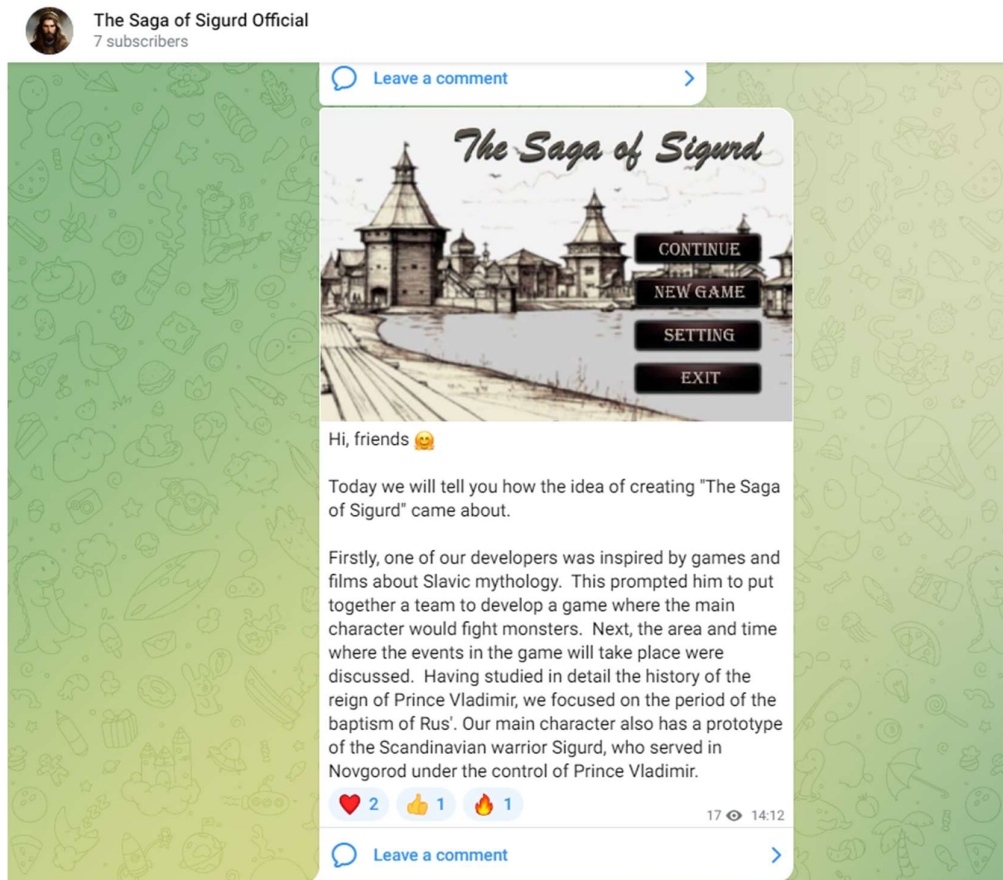


Рисунок 6.3 – Акаунт гри у Telegram

Акаунт зібрав найбільш зацікавлених користувачів, тому саме у ньому публікуються найсвіжіші оновлення про розробку гри. Також, у Telegram були активно проведені опитування, результати яких мали вагомє значення та внесли корективи у програмний продукт. Під кожним постом користувачі мають можливість залишати коментарі та отримувати відповіді від розробників, що сприяє живій та продуктивній взаємодії зі спільнотою.

Акаунт у соціальній мережі X став найбільш активним серед сторонньої аудиторії завдяки ефективним алгоритмам просування публікацій (див. рис. 6.4).



Рисунок 6.4 – Акаунт гри у соціальній мережі X

Майже всі пости набирали непогані показники переглядів, що свідчить про значний інтерес до гри (див. рис. 6.5).

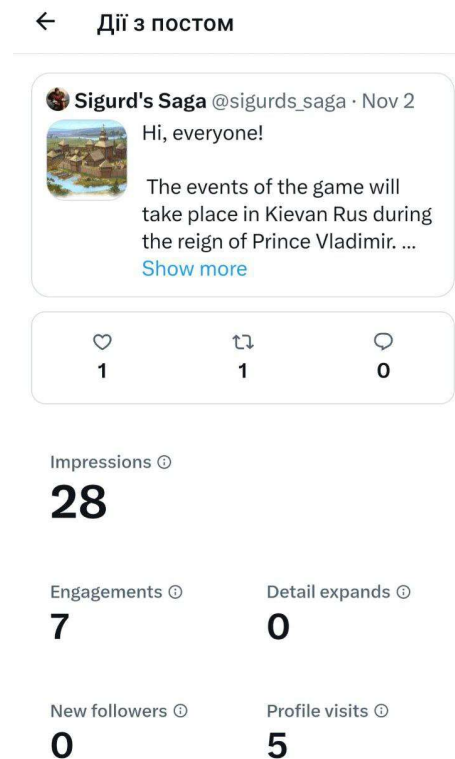


Рисунок 6.5 – Статистика публікації у соціальній мережі X

Хоча поки це не призвело до зростання великої аудиторії, акаунт у X має великий потенціал для поширення інформації про даний ігровий програмний

застосунок. Основні публікації в цій мережі містять роздуми про ігровий ринок, що спонукає глядачів брати участь у дискусіях та залучає нових користувачів. Таким чином, X слугує платформою для активного обговорення та взаємодії з аудиторією.

6.2 Вихід у продакшн

Як результат розробки будь-якого програмного продукту, важливо його впровадити на ринок технологій так, щоб кожна людина могла користуватися ним. У випадку ігор, пошук платформи для їхнього випуску має вирішальне значення. Для першої публікації вибір був зроблений на користь Itch.io. Цей сайт відомий своєю підтримкою незалежних розробників та гнучкими можливостями для публікації та продажу ігор. На даній платформі користувачі можуть знайти детальну інформацію про ігрові застосунки, переглянути скріншоти, трейлери, а також завантажити свою демо-версію програмного продукту для ознайомлення. Itch.io також дозволяє гравцям залишати відгуки та коментарі, що допомагає отримувати цінний зворотний зв'язок і вдосконалювати продукти.

Був створений особистий акаунт, де першою публікацією став ігровий програмний застосунок «The Saga of Sigurd» (див. рис. 6.6).

The Saga of Sigurd

This is a 2D game that will be in the Metroidvania genre. This game is about a hero of Scandinavian origin who lived during the times of Kievan Rus during the reign of St. Vladimir Svyatoslavovich. It was not an easy fate for him to become a mediator between the Slavic gods and people. The main character will have to plunge into battles with mythological monsters and, at the same time, look for a solution to save the Kyiv lands from the Christian faith and the eternal confrontations of pagan gods. Whether the main character will be able to restore the balance in Kievan Rus between the world of spirits and people remains to be seen...

[More information](#) ^

Published 3 hours ago

Status [Prototype](#)

Author [Elvira Lukhmenko](#)

Genre [Adventure](#), [Card Games](#), [Fighting](#)

Tags [2D](#)

Download

[Download Now](#) Name your own price

Click download now to get access to the following files:

The Saga of Sigurd.rar 273 MB

Comments

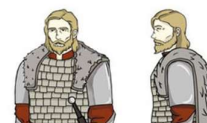


Рисунок 6.6 – Сторінка гри «The Saga of Sigurd» на платформі Itch.io

На платформі була представлена основна інформація про гру, а також прикріплені скріншоти, які демонструють її геймплей. Крім того, користувачам надається інструкція та файл програми для завантаження додатку на їхні персональні комп'ютери. Після отримання ігрового досвіду гравці мають можливість залишити коментар на офіційній сторінці, що сприяє формуванню об'єктивної думки про переваги та недоліки продукту. Цей зворотний зв'язок має велике значення для подальшого вдосконалення та розробки ігрового програмного застосунку «The Saga of Sigurd».

ВИСНОВКИ

Під час розробки ігрового програмного застосунку відбулася важлива та продуктивна робота. Кожен етап проекту сприяв утворенню цілісного та інноваційного продукту. В процесі роботи було проведено аналіз аналогічних ігор на ринку, що дозволило зрозуміти потреби цільової аудиторії та сформувавши чіткі вимоги до програми для різних операційних систем. Крім того, проектування архітектури програмного забезпечення було підкріплене створенням UML діаграм, що забезпечило чіткість та структурованість продукту. Важливим етапом був вибір оптимального середовища розробки та інструментів для ілюстрацій та анімацій, що сприяло ефективній роботі з ігровим двигуном. Цей комплексний підхід дозволив створити продукт, що виділяється функціонально та привертає увагу своїм вишуканим виконанням.

Початково була розглянута важливість ігрових програм та їх роль у сучасному технологічному світі. Особливо цінуються серед користувачів ігри, які базуються на науковому підґрунті, зокрема історичному. Вони відкривають широкі можливості для пізнання історії та розвитку культурного розуміння через захопливий та інтерактивний формат. Такі продукти стають не лише засобом задоволення дозвілля, але і потужним інструментом для освіти та пізнання.

Проведений аналіз аналогічних ігрових додатків мав за мету інтеграцію найкращих практик та уникнення помилок, які раніше були виявлені в них. Крім того, він сприяв формуванню чітких вимог до операційних систем, таких як Windows, Linux та Mac, забезпечуючи оптимальну роботу програми на різних платформах. Аналіз аналогічних продуктів дозволив не лише уникнути потенційних проблем, але й виявити можливості для інновацій та підвищення конкурентоспроможності. В результаті, цільова аудиторія була чітко визначена, що сприятиме успішній комерціалізації продукту.

За допомогою UML діаграм були створені діаграми пакетів, прецедентів, діяльності та стану, які ефективно проілюстрували основні режими функціонування гри. Ці діаграми стали важливим інструментом для розуміння та

управління різними аспектами програми, сприяючи його вдосконаленню та розвитку. Використання UML діаграм підкреслило важливість системного підходу до проектування програмного забезпечення, дозволяючи з легкістю спілкуватися та вирішувати складні завдання.

На етапі проектування архітектури програмного продукту було ретельно обрано оптимальне середовище розробки, включаючи `ct.js`, а також програми для створення ілюстрацій та анімацій - Krita та Spine, які визначилися як найкращі серед різноманітних аналогів у цій сфері. Використання Spine для створення 2Д анімації за допомогою побудови скелета значно полегшило та прискорило процес розробки. Крім того, Krita виділилася своїм розмаїттям інструментів для створення растрових зображень, забезпечуючи зручність малювання та найвищу якість графічного контенту.

У результаті цього дослідження вдалося довести, що виважений та комплексний підхід до розробки ігрового програмного забезпечення є ключем до успіху. Було використано передові методи та інструменти, включаючи аналіз ринку, UML діаграми та оптимальний вибір програмних середовищ, для створення продукту високої якості, який здатний задовольнити потреби сучасної аудиторії та залучити її увагу. Підсумковий результат — ігровий програмний застосунок у жанрі *Metroidvania* з елементами карткової гри, який поєднує у собі якість виконання, глибину геймплею та навчальну складову, що робить його не лише засобом розваги, але й потужним інструментом для освіти та розвитку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Термін Metroidvania [Електронний ресурс] – URL: <https://en.wikipedia.org/wiki/Metroidvania> (дата звернення: 30.04.2024)
2. Офіційна сторінка ігрового застосунку «Pentiment» [Електронний ресурс] – URL: <https://pentiment.obsidian.net/> (дата звернення: 30.04.2024)
3. Офіційна сторінка ігрового застосунку «Apotheon» [Електронний ресурс] – URL: <https://www.apotheongame.com/> (дата звернення: 30.04.2024)
4. Офіційна сторінка ігрового застосунку «Brasphemous» [Електронний ресурс] – URL: <https://www.blasphemous2game.com/> (дата звернення: 30.04.2024)
5. Офіційна сторінка ігрового застосунку «Castlevania aria of sorrow» [Електронний ресурс] – URL: <https://www.konami.com/games/castlevania/eu/en/> (дата звернення: 30.04.2024)
6. Харченко О. Г., Яцишин В. В., Райчев І.Е. Інструментальний засіб розробки та комунікації вимог якості до програмних систем. Науковий журнал «Інженерія програмного забезпечення». 2010. №2. С. 29-34.
7. Офіційна сторінка середовища розробки ct.js [Електронний ресурс] – URL: <https://ctjs.rocks/> (дата звернення: 30.04.2024)
8. Яковина В. С., Парфенюк Ю. І. Використання засобів UML для прогнозування надійності програмного забезпечення на етапі його проектування. Вісник Національного університету "Львівська політехніка". 2013. С. 151-156.
9. Свінцицька О. М., Граф М. С., Нікітчук Т.М. Метод Use Case в плануванні проєктів з інформаційних технологій. Технічна інженерія. 2022. С. 77-83.
10. Костюк Ю. В. Використання діаграми діяльності для моделювання та аналізу процесу прийняття рішень // Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення: матеріали Міжнар. наук. конф., м. Тернопіль, 8-9 червня 2023 р. Тернопіль, 2023. С. 60-62.
11. Тонелла П., Потрич А. Реверсна інженерія об'єктно-орієнтованого коду. Діаграма пакетів. Нью-Йорк, 2005. С. 133 – 154.

12. Офіційна сторінка програмного застосунку для створення графіки «Adobe photoshop» [Електронний ресурс] – URL: <https://www.adobe.com/products/photoshop.html> (дата звернення: 30.04.2024)

13. Офіційна сторінка програмного застосунку для створення графіки «Krita» [Електронний ресурс] – URL: <https://krita.org/> (дата звернення: 30.04.2024)

14. Офіційна сторінка програмного застосунку для створення графіки «Clip studio paint» [Електронний ресурс] – URL: <https://www.clipstudio.net/en/> (дата звернення: 30.04.2024)

15. Офіційна сторінка програмного застосунку для роботи з анімацією «Spriter pro» [Електронний ресурс] – URL: <https://brashmonkey.com/spriter-pro/> (дата звернення: 30.04.2024)

16. Офіційна сторінка програмного застосунку для роботи з анімацією «Spine» [Електронний ресурс] – URL: <https://uk.esotericsoftware.com/> (дата звернення: 30.04.2024)

17. Офіційна сторінка програмного застосунку для роботи з анімацією «Adobe after effects official» [Електронний ресурс] – URL: <https://www.adobe.com/products/aftereffects.html> (дата звернення: 30.04.2024)

18. Документація з застосуванням IK constraints [Електронний ресурс] – URL: <https://uk.esotericsoftware.com/spine-ik-constraints> (дата звернення: 01.06.2024)

19. Документація з застосуванням Transform constraints [Електронний ресурс] – URL: <https://uk.esotericsoftware.com/spine-transform-constraints#Transform-constraints> (дата звернення: 01.06.2024)

20. Документація з застосуванням Path constraints [Електронний ресурс] – URL: <https://uk.esotericsoftware.com/spine-path-constraints#Path-constraints> (дата звернення: 01.06.2024)

21. Документація з застосуванням Keys [Електронний ресурс] – URL: <https://uk.esotericsoftware.com/spine-keys#Keys> (дата звернення: 01.06.2024)

ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016311370

Дата перевірки:
02.06.2024 17:41:44 EEST

Тип перевірки:
Doc vs Library

Дата звіту:
02.06.2024 17:43:00 EEST

ID користувача:
100012353

Назва документа: 2024_Б_ПІ_ПЗПІ-20-1_Лухменко_Е_К

Кількість сторінок: 86 Кількість слів: 13514 Кількість символів: 107998 Розмір файлу: 2.96 MB ID файлу: 1016108234

4.3%
Схожість

Найбільша схожість: 1.61% з джерелом з Бібліотеки (ID файлу: 1016096604)

Пошук збігів з Інтернетом не проводився

4.3% Джерела з Бібліотеки 448

Сторінка 88

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0%
Вилучень

Немає вилучених джерел

ДОДАТОК Б
Слайди презентації

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кваліфікаційна робота бакалавра

Ігровий програмний застосунок у жанрі
Metroidvania з елементами карткової гри

Виконала:
ст. гр. ПЗП-20-1
Лухменко Е.К.

Науковий керівник:
ст.викл. кафедри ПІ
Саманцов О.О.

ОІ

Аналіз конкурентів



Pentiment



Apotheon

ІГРИ АНАЛОГИ

Castlevania Aria Of Sorrow



Blasphemous



Критерії	Pentiment	Apotheon	Blasphemous	Castlevania: Aria of Sorrow
Унікальний сюжет в історичному світі	Високий	Помірний	Слабкий	Слабкий
Унікальний художній стиль	Високий	Високий	Помірний	Слабкий
Простота навігації у просторі	+	-	-	+
Прості елементи RPG	+	+	-	+
Сюжетні діалоги	+	-	-	+
Можливість зберігання	-	+	-	+
Вплив виборів на гру	-	-	-	-
Інноваційність у геймплеї	+	-	-	+
Наявність альтернативних закінчень	-	-	+	+
Система розвитку персонажа	-	+	-	+
Розмір світу	Помірний	Низький	Високий	Низький
Романтичність в об'єкті	-	+	-	-

02

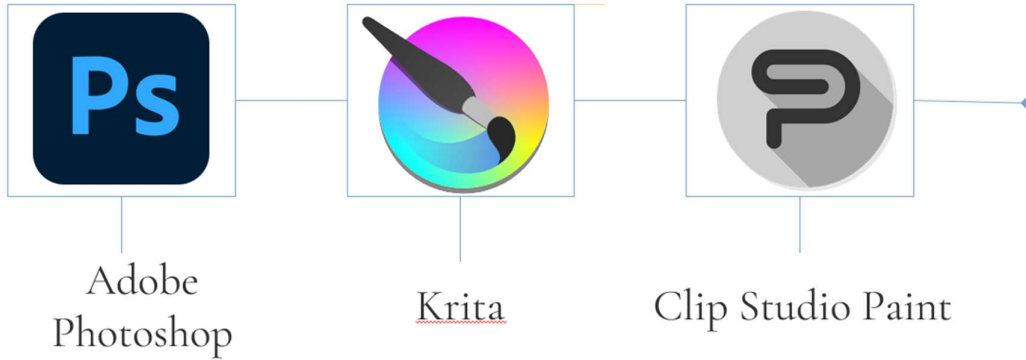
Вибір середовища розробки

Середовище розробки

St.js, що базується на [Electron](#), було обране, як головна програма для побудови програмного ігрового застосунку.



Програми для створення ілюстрацій



Аналіз програм для роботи з графікою

Критерії	Adobe Photoshop	Krita	Clip Studio Paint
Взаємодія із відкритими джерелами	Висока	Висока	Середня
Невибагливість до ресурсів системи	Низька	Середня	Середня
Інтуїтивність інтерфейсу	Низька	Висока	Середня
Функціональність для малювання	Висока	Висока	Висока
Підтримка форматів	Висока	Висока	Висока
Інтеграція з ігровими двигунами	Середня	Середня	Середня

Програми для створення анімацій



Spine



Spriter Pro



Adobe After
Effects

Аналіз програм для створення анімацій

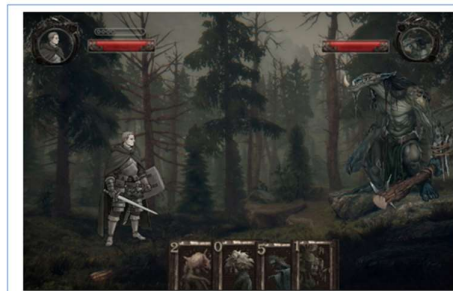
Критерії	Spriter Pro	Spine	Adobe After Effects
Інструментарій для анімації у грі	Висока	Висока	Низька
Невибагливість до ресурсів системи	Середня	Висока	Низька
Інтуїтивність інтерфейсу	Середня	Висока	Низька
Орієнтованість під 2D скелетну анімацію	Висока	Висока	Низька
Підтримка форматів	Середня	Висока	Висока
Інтеграція з ігровими движунами	Середня	Висока	Низька

Моделювання інтерфейсу гри

Діалогова система



Система бою



04

Створення
спрайтів та
анімацій

Основні інструменти, які були використані у програмах...

Krita:



Застосування фільтрів



Створення власних пензлів



Використання інструментарію для побудови перспективи



Застосування Wrap Around Mode



Створення покадрової анімації

Spine:

Робота з сіткою

Застосування обмежень до кісток

Використання Скінів

Застосування ключів на часовій шкалі



05

Інтерфейс користувача

Типи тестування

01

Тестування
інсталяції

02

Функціональне
тестування

03

Тестування інтерфейсу
користувача

04

Тестування
відмовостійкості
та відновлення

05

Тестування
продуктивності

06

Тестування
конфігурації

Баг-репорт

- 2.1 **Назва:** При зміні будь-якої анімації на анімацію idle персонаж дивиться у протилежний бік.
- 2.2 **Короткий опис:** Після виконання будь-якої дії головний герой переходить у стан очікування. Однак його напрям(бік, в який він був обернений) не співпадає з напрямом попередньої дії.
- 2.3 **Компонент програми:** Клієнтська/front-end частина.
- 2.4 **Серйозність:** S4 Незначний.
- 2.5 **Пріоритет:** P3 Низький.
- 2.6 **Кроки відтворення:**
 - 2.6.1 Виконати переміщення вліво.
 - 2.6.2 Зупинитися.
 - 2.6.3 Виконати переміщення вправо.
 - 2.6.4 Зупинитися.
- 2.7 **Фактичний результат:** При переході з будь-якої анімації на анімацію idle персонаж завжди дивиться у протилежний бік.
- 2.8 **Очікуваний результат:** При зміні будь-якої анімації на анімацію idle персонаж дивиться у той бік, в який дивився перед цим.
- 2.9 **Прикріплений файл:** Посилання на Google Drive зі скриншотами та відео, де були відображені баги
<https://drive.google.com/drive/folders/1IAwfMXbPgXJ6wg4YbDf6E6SqrXSMKiU?usp=sharing>

07


Результат роботи

Ігровий програмний застосунок
був представлений на платформі
«Itch.io» під назвою «The Saga of
Sigurd»

The Saga of Sigurd

This is a 2D game that will be in the Metroidvania genre. This game is about a hero of Scandinavian origin who lived during the times of Kievan Rus during the reign of St. Vladimir Svyatoslavovich. It was not an easy fate for him to become a mediator between the Slavic gods and people. The main character will have to plunge into battles with mythological monsters and, at the same time, look for a solution to save the Kyiv lands from the Christian faith and the eternal confrontations of pagan gods. Whether the main character will be able to restore the balance in Kievan Rus between the world of spirits and people remains to be seen...

More information ^

Published  3 hours ago
Status [Prototype](#)
Author [Elyra Lushmenko](#)
Genre [Adventure](#), [Card Game](#), [Fighting](#)
Tags [2D](#)

Download

[Download Now](#) Name your own price


Click download now to get access to the following files:

The Saga of Sigurd.rar 273 MB

Comments

Write your comment...





Дякую за увагу !

ДОДАТОК В

Специфікація ПЗ

1. ВСТУП

1. Огляд продукту

Продукт представляє з себе відеогру в жанрі *Metroidvania*, яка ґрунтується на слов'янській міфології і естетиці. Ігрові події відбуваються у Київській Русі за часів Володимира Великого. Головний герой – воїн, який був добре знайомий із князем.

Геймплей гри відбувається в 2D світі, розділеному на окремі локації. Гравець може вільно пересуватися світом, шукати магичні картки, щоб використовувати їх у бою. Для переміщення у окремій локації гравець може використовувати платформи, щоб можна спускатися вниз чи підніматися на них вгору.

Бойова система є класичною для жанру *Metroidvania*: герой може використовувати меч у ближньому бою або магичні карти, витрачаючи ману. В грі є діалоги, які просувають сюжет та інформують гравця про ігровий світ.

2. Мета

Говорячи про мету створення продукту необхідно відзначити, що існує досить багато ігор жанру *Metroidvania*, але існуючі аналоги, попри їх популярність, мають свої недоліки. Тому створення нової гри в цьому жанрі виправдане через кілька ключових причин. По-перше, нова гра може оживити жанр і зацікавити гравців, які шукають нових вражень. Це вирішує проблему втоми від старих ігор. По-друге, нові потреби гравців, як-от більша складність і інноваційні механіки, можуть бути враховані в новій грі. Свіжі ідеї й унікальні підходи можуть привернути увагу як фанатів жанру, так і нових гравців. Випуск нової гри також може відродити інтерес до жанру, який може втрачати популярність через перенасиченість або брак нових проектів.

3. Межі

Гра розробляється як автономний продукт без необхідності підключення до серверів для основного геймплею, крім випадків завантаження оновлень. Всі збереження гри зберігаються локально на пристрої гравця. Немає підтримки хмарних збережень.

ПЗ розробляється тільки під наступні операційні системи та платформи: Windows, macOS, Linux, Android, iOS. Кожна з платформ має відповідати власним мінімальним системним властивостям.

Команда розробників складається з 3 осіб, що відповідають за проєктування, дизайн, написання коду та тестування ПЗ.

Проєкт має бути завершений до 12.06.2024.

4. Посилання

<https://docs.ctjs.rocks/ct-concepts.html> – офіційна документація до ігрового двигуна ct.js.

<https://en.m.wikipedia.org/wiki/Metroidvania> – історія та опис жанру Metroidvania.

<https://uk.esotericsoftware.com/> - офіційна сторінка програмного застосунку для роботи з анімацією “Spine”

<https://krita.org/> - офіційна сторінка програмного застосунку для створення графіки «Krita»

5. Означення та аббревіатури

JS – JavaScript

TS – TypeScript

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

CG – Computer Graphics

2. ЗАГАЛЬНИЙ ОПИС

1. Перспективи продукту

В першу чергу варто сказати, що хоча даний програмний продукт є закінченим, його розвиток та підтримка мають бути навіть після публікації та випуску першої версії гри. Як вже було зазначено, з точки зору ринкових перспектив програмного продукту, основними можна назвати такі як вихід на міжнародний ринок та публікацію на усіх основних та найпопулярніших ігрових майданчиках, таких як Steam, Epic Games Store та інші. Окім цього, слід зазначити, що в майбутньому, при умовах успіху гри на основних цільових платформах, у перспективі можливо створити окрему версію гри для найпопулярніших ігрових консолей, таких як PlayStation та Xbox.

2. Функції продукту

При ввімкненні гри головними опціями для гравця є налаштування ігрового процесу, продовження гри у тому місці де він закінчив минулого разу, початок нової гри та вихід із гри.

В свою чергу, при переході в режим самої гри, користувач має досить багато різного функціоналу на вибір. Так, гравець може досліджувати світ гри, дізнаватися нові деталі ігрового сюжету та спілкуватися із неігровими персонажами. Окрім цього, гравець може шукати картки та артефакти, які можуть полегшити ігровий процес та покращити здібності головного героя.

Також важливою складовою є бойова система. Так, гравець також може або тренуватися на ворогах та досліджувати їх слабкі сторони, або проходити сюжет гри, яким вимагає зіткнення із ворогами. Важливою частиною бойової системи є те, що вона дозволяє герою комбінувати фізичні атаки із магичною системою.

3. Характеристики користувачів

Як вже було зазначено, основною цільовою аудиторією гри є чоловіки віком від 18 років. Така категорія була обрана з урахуванням того, що така цільова аудиторія є найбільш зацікавленою у подібних іграх, а також, окрім цього, є найбільш платіжоздатною, що має позитивно вплинути на монетизацію та комерційний успіх нашого програмного продукту.

4. Загальні обмеження

- Для розробки гри має використовуватися ігровий двигун C#.js;
- Гра має працювати на Windows, Unix, MacOS, Android, IOS та Web;
- Гра повинна мати мінімум 30 кадрів на секунду;
- Гра має бути локалізована англійською мовою.

5. Припущення і залежності

- Користувачі заздалегідь зацікавлені у програмному продукту;
- Ігрові платформи користувачів оновлено до останньої версії;
- Гра має визначений набір функціоналу, який має бути реалізовано для першої версії;
- Команда розробників має ліцензії на програмне забезпечення, що використовується.

3. КОНКРЕТНІ ВИМОГИ

1. Вимоги до зовнішніх інтерфейсів

1. Інтерфейс користувача

UI у ігровому програмному застосунку поділяється на 2D рівні та UIX складові. User Experience виключно представлений у головному меню, де гравець за допомогою кнопок може виконувати навігацію у грі, а саме почати нову гру, завантажити поточку та вийти на робочий екран (див. рис. 3.1).



Рисунок 3.1 - Головне меню гри

Уся інша навігація у ігровому програмному застосунку виконується за допомогою клавіатури та комп'ютерної миші.

Застосунок має декілька рівнів, де гравець, керуючи головним героєм, може взаємодіяти з об'єктами, ворожими та нейтральними персонажами.

Перший рівень представлений у вигляді нижнього світу, де користувач проходить навчання та веде внутрішній діалог із самим собою (див. рис. 3.2).



Рисунок 3.2 - Діалог у грі

Головний герой має показники здоров'я та магичних здібностей, які супроводжують його протягом усієї гри. Також на першому рівні він отримує колекційну карту, яку може використовувати під час бою з супротивником (див. рис. 3.3).



Рисунок 3.3 - Навчальна локація гри

У ігровому програмному застосунку наявна локація внутрішнього приміщення будинку головного героя, де розташовані побутові речі та його дружина (див. рис. 3.4).

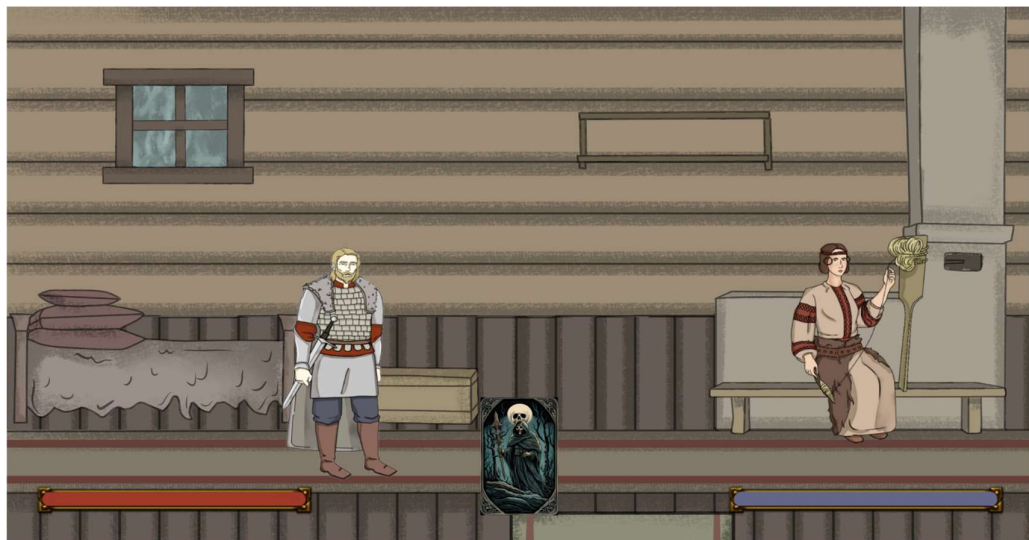


Рисунок 3.4 - Будинок головного героя

Наступною локацією для переходу є селище, де розташовані будинки NPC (див. рис. 3.5).



Рисунок 3.5 - Локація селища

Користувач може відвідати рівень з мостом над річкою, яка є способом переміщення героя до верхнього або нижнього світу (див. рис. 3.6).

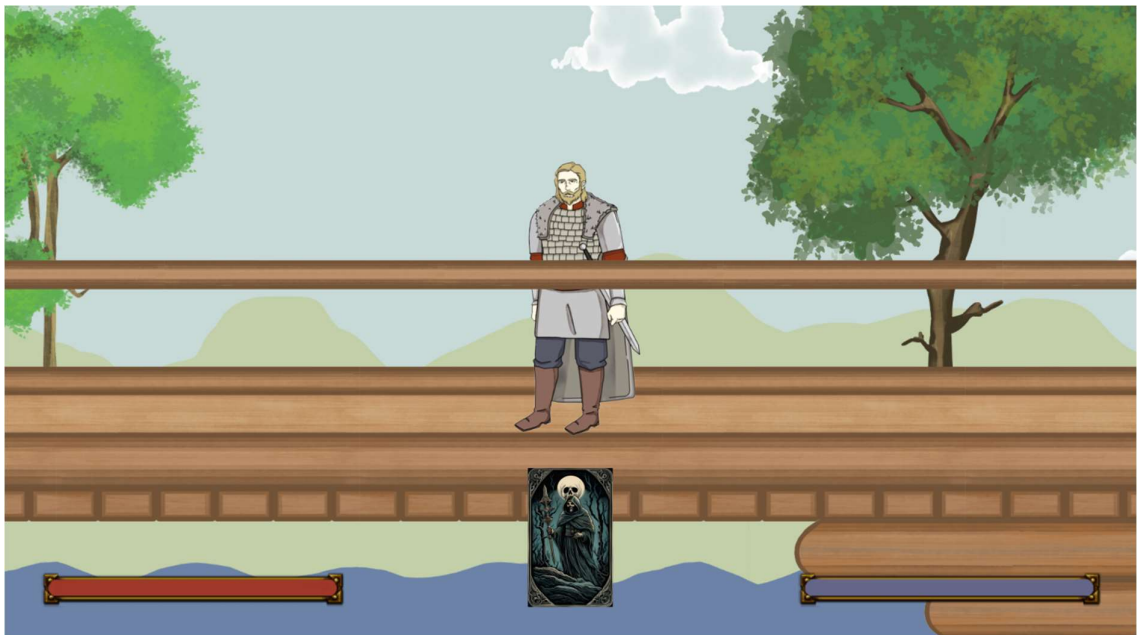


Рисунок 3.6 - Локація з мостом

У локації з помешканням князя Володимира, гравець може почати взаємодію з ним (див. рис. 3.7).



Рисунок 3.7 - Локація з помешканням князя

З лівого боку від селища, користувач може відвідати локацію з капищем та зустріти волхва (див. рис. 3.8).



Рисунок 3.8 - Локація з капищем

Для того щоб відвідати князя Володимира, гравцю необхідно пройти рівень з лісом, де його очікують такі ворожі моби, як Волкодлак та Ирка (див. рис. 3.9).



Рисунок 3.9 - Локація з лісом

З великим шансом, користувачу ігрового програмного застосунка необхідно буде вступити у бій, щоб здолати супротивників зі своїми унікальними механіками. Ця локація найкраще демонструє бойову систему гри.

2. Комунікаційний протокол

Компоненти ігрового програмного застосунку певним чином взаємодіють один з одним, що демонструє їх комунікацію. Програму можна поділити на два основних середовища: сховище ігрових файлів та файл програми `st.js` (див. рис. 3.10).

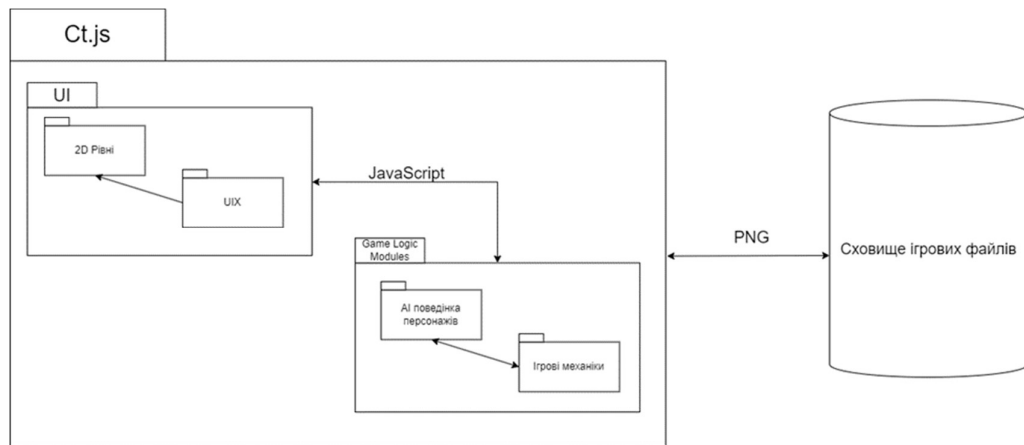


Рисунок 3.10 - Діаграма, яка демонструє комунікаційний протокол

Між собою вони взаємодіють за допомогою файлу з розширенням .png. Можна також розділити основну програму на два підрозділи: UI та Модуль з ігровою логікою. Вони мають окремий метод комунікації, а саме за допомогою мови програмування JavaScript.

3. Обмеження пам'яті

Були визначені мінімальні обмеження пам'яті під час завантаження та запуску гри:

- Оперативна пам'ять: 512 MB;
- Місце на диску: 1 GB.

4. Операції

Операції у програмному ігровому застосунку можна поділити на три види: взаємодія з UIX, введення даних з апаратного пристрою, взаємодія з оточенням. Під час взаємодії з User Experience гравець може натискати на кнопки представлені у головному меню. Коли користувач вводить певні клавіші на клавіатуру та на комп'ютерній миші, він отримує певні зміни у інтерфейсі. Головний герой може взаємодіяти з ігровим світом, на що програма буде відповідати.

4.1 Початок нової гри

4.1.1 Представлення

Ігровий програмний застосунок повинен надавати можливість почати проходити нову гру.

4.1.2 Вхідні дані

Користувач натискає кнопку “New Game” у головному меню.

4.1.3 Обробка

Пошук рівня, з якого починається нова гра.

4.1.4 Результат

Відображення першого рівня гри.

4.2 Завантаження збереження

4.1.1 Представлення

Ігровий програмний застосунок повинен надавати можливість завантажувати рівень, на якому востаннє був гравець.

4.1.2 Вхідні дані

Користувач натискає кнопку “Load Game” у головному меню.

4.1.3 Обробка

Пошук рівня, на якому користувач був останній раз.

4.1.4 Результат

Відображення рівня та усіх даних, які мав гравець в останній раз під час проходження.

4.3 Вихід з ігрового застосунку

4.1.1 Представлення

Ігровий програмний застосунок повинен надавати можливість вийти з нього до робочого екрану.

4.1.2 Вхідні дані

Користувач натискає кнопку “Exit” у головному меню.

4.1.3 Обробка

Завершення роботи програми.

4.1.4 Результат

Користувач вийшов до власного робочого стола пристрою з повним закриттям ігрового програмного застосунку.

4.4 Рух персонажем вліво

4.1.1 Представлення

Ігровий програмний застосунок повинен надавати можливість керувати головним героєм переміщуючи його вліво.

4.1.2 Вхідні дані

Користувач натискає клавішу“А” на клавіатурі.

4.1.3 Обробка

Перевірка введених даних з клавіатури.

4.1.4 Результат

Головний герой переміщується вліво до повного відпускання певної клавіші.

4.5 Рух персонажем направо

4.1.1 Представлення

Ігровий програмний застосунок повинен надавати можливість керувати головним героєм переміщуючи його вправо.

4.1.2 Вхідні дані

Користувач натискає клавішу“D” на клавіатурі.

4.1.3 Обробка

Перевірка введених даних з клавіатури.

4.1.4 Результат

Головний герой переміщується вправо до повного відпускання певної клавіші.

4.6 Стрибок персонажа

4.1.1 Представлення

Ігровий програмний застосунок повинен надавати можливість керувати головним героєм переміщуючи його догори.

4.1.2 Вхідні дані

Користувач натискає клавішу“W” або пробіл на клавіатурі.

4.1.3 Обробка

Перевірка введених даних з клавіатури.

4.1.4 Результат

Головний герой переміщується вліво до завершення анімації стрибка.

4.7 Удар зброєю

4.1.1 Представлення

Ігровий програмний застосунок повинен надавати можливість керувати головним героєм завдаючи удар зброєю.

4.1.2 Вхідні дані

Користувач натискає ЛКМ.

4.1.3 Обробка

Перевірка введених даних з комп'ютерної миші.

4.1.4 Результат

Головний герой наносить удар противнику.

4.8 Активація магії

4.1.1 Представлення

Ігровий програмний застосунок повинен надавати можливість керувати головним героєм використовуючи магичні здібності з колекційних карток.

4.1.2 Вхідні дані

Користувач натискає клавішу“1” на клавіатурі.

4.1.3 Обробка

Перевірка введених даних з клавіатури.

4.1.4 Результат

Головний герой відтворює магію з корекційної картки відповідно до її механік.

4.9 Закінчення гри

4.1.1 Представлення

Ігровий програмний застосунок повинен надавати можливість завершувати гру шляхом смерті головного героя.

4.1.2 Вхідні дані

Головний герой отримує урон, що змушує показник його здоров'я дорівнювати нулю.

4.1.3 Обробка

Перевірка показника здоров'я головного героя.

4.1.4 Результат

Перенесення гравця до головного меню.

4.10 Взаємодія з NPC або об'єктами

4.1.1 Представлення

Ігровий програмний застосунок повинен надавати можливість взаємодіяти з NPC або об'єктами за допомогою системи діалогу або натиснення клавіші на клавіатурі.

4.1.2 Вхідні дані

Користувач натискає клавішу "E" на клавіатурі.

4.1.3 Обробка

Перевірка чи відповідає об'єкт у грі взаємодії із застосуванням відповідної клавіші.

4.1.4 Результат

Відтворення діалогу або переміщення між локаціями.

5. Функції продукту

Ігровий програмний застосунок містить основних функцій для коректної та правильної роботи додатка:

5.1 Функція відтворення діалогу

Дана функція відтворюється під час активації діалогу з певними NPC або на початку рівнів. Вона відтворює репліки до поки не спрацьовує тригер отримання квесту (див. рис. 3. 10).

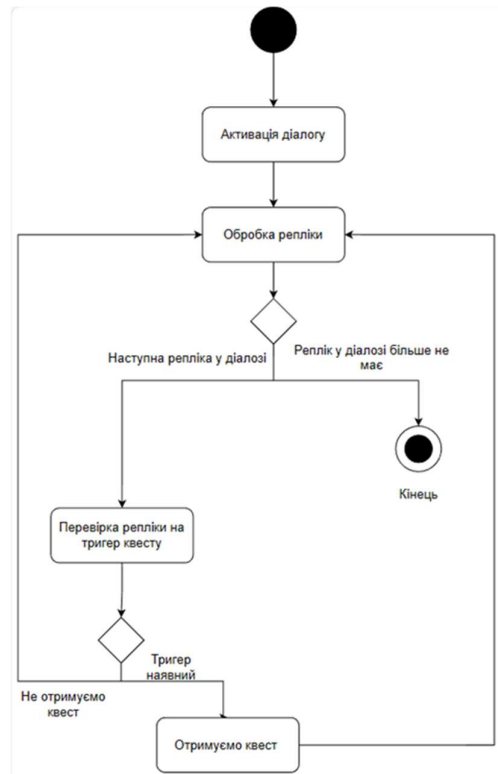


Рисунок 3.10 - Діаграма, яка демонструє діалогову функцію

5.2 Функція переміщення головного персонажа

Вона реагує на введення даних з клавіатури та комп'ютерної миші. Відповідно до дії користувача система перевіряє даний запит та реагує на нього певними змінами у інтерфейсі користувача.

5.3 Функція побудови бойової системи

Дана функція містить у собі логіку, за допомогою якою відбуваються дії персонажів у бою. Для кожного окремого NPC механіки можуть бути різними в залежності від його характеристик.

5.4 Функція взаємодії з об'єктами

У певні моменти часу ігровий програмний застосунок пропонує гравцю певну взаємодію з нею, після чого процедура завершується. Одним з можливих

варіантів завершення є збереження гри та повернення до головного меню (див. рис. 3.11).

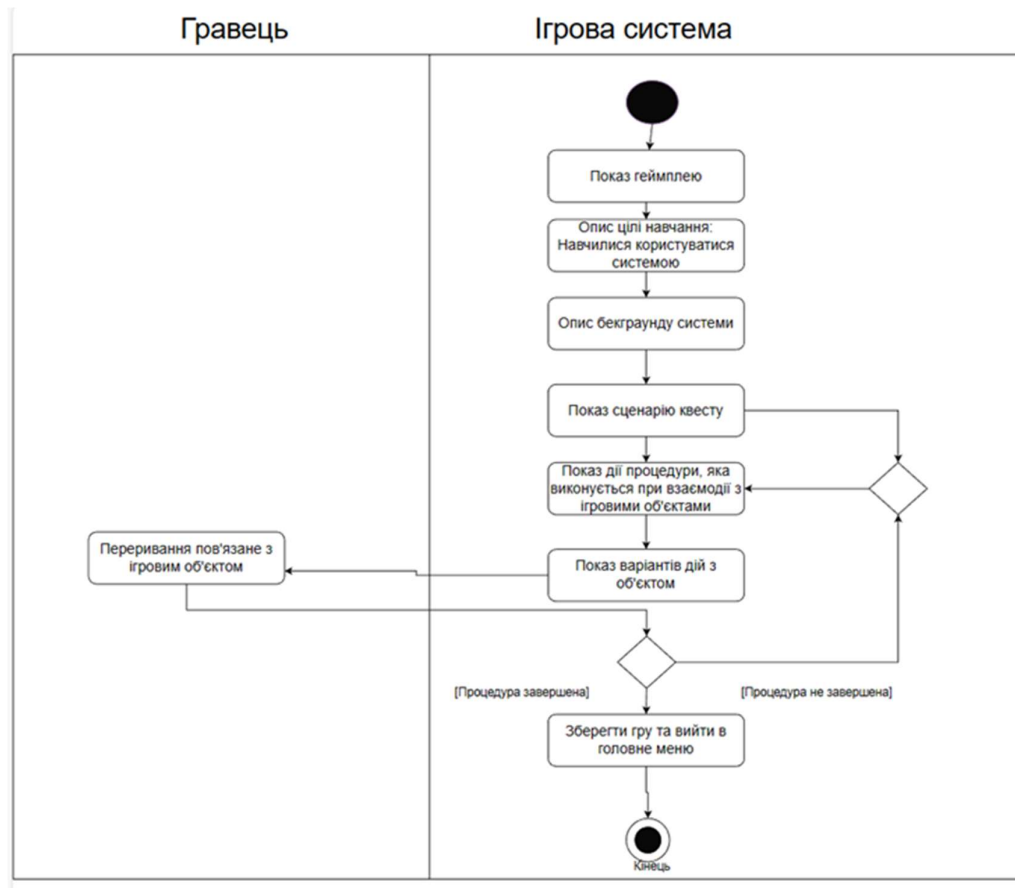


Рисунок 3.11 - Діаграма, яка демонструє взаємодію гравця із системою

6. Припущення й залежності

До припущень та залежностей можна віднести такі ситуації:

Користувачі будуть використовувати гру на персональному комп'ютері, мобільному пристрої або у Web браузері.

Користувачі будуть керувати головним героєм шляхом натискання відповідних клавіш на клавіатурі.

У певний момент часу користувачі захочуть вийти з гри та перейти до свого робочого екрану.

Користувачам буде необхідне графічне відображення кожної дії, яку вони зробили.

Користувачам буде необхідно завантажити збереження проходження гри після виходу з неї або смерті головного героя.

2. Властивості програмного продукту

Розробляємий продукт є ігровим застосунком в жанрі Metroidvania з глибоким оригінальним сюжетом, гарним стилем, захоплюючою бойовою системою та атмосферним музичним супроводом.

У грі герой може переміщатися по горизонталі за допомогою бігу та по вертикалі стрибаючи по платформах. Також є можливість переміщатись між локаціями, взаємодіяти з предметами та персонажами. Взаємодія з персонажами представляє собою діалог, що надасть гравцю інформація про світ гри та вкажіть на сюжетні завдання. У бою з ворожими істотами є можливість атакувати у ближньому бою мечем чи у використувувати магію знайдених карток.

Застосунок має гарний та зрозумілий дизайн, що впливає на відчуття ігрового досвіду користувачів. Дизайн застосунку допомагає гравцю зорієнтуватися та зрозуміти всі функції представлені у грі, що прямо вплине на рівень їх задоволення. Крім цього дизайн дає можливість поглибитися в ігровий світ та повноцінно відчувати його атмосферу.

Продукт повинен має можливість бути запущений на платформах Windows, macOS, Linux, Android та iOS. Ігровий програмний застосунок ефективно витрачає системні ресурси на різних пристроях.

3. Атрибути програмного продукту

3.1 Надійність

ПЗ, що розробляється, повинна мати автоматичне збереження ігрового прогресу, щоб мінімізувати втрати даних при збої, при переході на новий рівень. Після перезапуску гри користувач повинен мати можливість продовжити з останньої точки збереження.

Гра повинна фіксувати всі критичні помилки в лог-файлі для подальшого аналізу та інформувати гравців про помилки у інтерфейсі. Також додаток повинен підтримувати можливість моніторингу використання ресурсів у реальному часі.

3.2 Доступність

Кажучи про доступність, як було зазначено раніше, даний ігровий програмний продукт відповідає основним принципам евристики, що робить його доступним для нових користувачів завдяки зрозумілому та простому програмному інтерфейсу.

Так, інтерфейс нашого додатку чітко дозволяє зрозуміти що він очікує від гравця та дозволяє йому легко взаємодіяти з програмною системою. Окрім цього, як було зазначено вище, в перспективі гра буде локалізована на більшість найпоширеніших мов, що зробить гру доступнішою для кінцевих користувачів із різних країн.

Також дуже важливим елементом доступності є навчання. Так, коли гравець починає нову гру, його зустрічає навчальний рівень, який проводить гравця та навчає його основних ігрових механік, а також показує як керувати персонажем та взаємодіяти з навколишнім середовищем ігрового світу. Це значно знижує криву навчання і дозволяє гравцеві швидше включитися у ігровий процес.

3.3 Безпека

Стосовно безпеки, у нашій грі імплементовано основні принципи, які захищають кінцевих користувачів. Так, наша гра не збирає персональну інформацію про користувачів та не оперує нею, що значно поліпшує загальну безпеку та у найгіршому випадку може значно знизити ризики компрометації даних гравців.

Окрім цього, наша гра розташована на популярних ігрових майданчиках та покладається на захист, який надають ці системи. Стосовно самої гри, оскільки значна її частина працює на мові програмування JavaScript та платформі Electron, то нашу гру неможливо зворотно зібрати, оскільки гра не запускається нативно, а

працює в окремому захищеному середовищі, а також не взаємодіє із файловою системою та операційною системою платформи, на якій гру було запущено. Такий підхід дозволяє захистити гру, в тому числі, від модифікацій і чит кодів, які могли би зруйнувати ігровий процес для гравця.

3.4 Супроводжуваність

У розробляемому додатку має бути система відстеження помилок і пропозицій користувачів, а також організовано службу технічної підтримки з відповідними інструкціями та процедурами. Повинно бути наведеною детальна технічна документація з описом всіх модулів та інструкції для користувачів з прикладами використання. Система має регулярно оновлюватися нові версії мають проходити юніт-тести для всіх ключових модулів.

3.5 Переносимість

Переносимість ПЗ - це важливий аспект, оскільки забезпечує можливість використання розробляемого ПЗ на різних платформах, пристроях, операційних системах тощо. Ігровий застосунок має коректно працювати на операційних системах Windows, macOS, Linux, Android та iOS. А також підтримувати пристрої управління цих платформ.

3.6 Продуктивність

При використанні додатку має бути стабільний фреймрейт та висока якість графіки на різних конфігураціях ПК, прийнятний час завантаження рівнів та текстур, стабільний геймплей без великих затримок, стійка продуктивність гри під високими навантаженнями.

ДОДАТОК Г
Приклади кодів програми

```
from PIL import Image
import os

def flip_and_save_images(input_folder, output_folder):
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)

    files = os.listdir(input_folder)

    for file in files:
        if file.endswith(".jpg") or file.endswith(".png") or file.endswith(".jpeg"):
            img_path = os.path.join(input_folder, file)
            img = Image.open(img_path)

            flip_img = img.transpose(Image.FLIP_LEFT_RIGHT)

            output_file = os.path.join(output_folder, file)

            flip_img.save(output_file)

    print(f"The {file} image has been successfully reflected and saved to
{output_folder}")
```