

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних
інформаційних систем

_____ Євген СІДЕНКО

« ____ » _____ 2026 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
РОЗРОБКА 2D-ГРИ З ЕЛЕМЕНТАМИ ВІЗУАЛЬНОЇ
НОВЕЛИ НА РУШІІ RPG MAKER

Спеціальність 122 Комп'ютерні науки
Освітня програма «Комп'ютерні науки»

Здобувач

_____ Владислав ВРАДІЙ

« ____ » _____ 2026 р.

Керівник д-р техн. наук, професор

_____ Ірина КАЛІНІНА

« ____ » _____ 2026 р.

Миколаїв – 2026

Завдання на виконання кваліфікаційної роботи

Чорноморський національний університет імені Петра Могили

(повне найменування закладу вищої освіти)

Факультет	Комп'ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	122 Комп'ютерні науки
Освітня програма	Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем

_____ Євген СІДЕНКО

«___» _____ 2025 р.

ЗАВДАННЯ на кваліфікаційну роботу здобувача

ВРАДІЙ ВЛАДИСЛАВ СЕРГІЙОВИЧ

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Розробка 2D-гри з елементами візуальної новели на рушії RPG Maker.

Керівник роботи: Калініна Ірина Олександрівна, професор кафедри ПІС, д-р техн. наук, професор.

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

Затверджена наказом ЧНУ ім. Петра Могили від «25» грудня 2026 р. №353

2. Строк представлення кваліфікаційної роботи «___» червня 2026 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: Очікуваним результатом є демо версія гри на ігровому рушії RPG Maker MZ.

4. Перелік питань, що підлягають розробці:

- 1) аналіз предметної області розробки 2D-ігор та особливостей жанрів JRPG і візуальних новел;
- 2) огляд існуючих рушіїв та обґрунтування вибору RPG Maker MZ;
- 3) дослідження можливостей RPG Maker MZ та його компонентів;
- 4) проєктування структури ігрового застосунку та основних механік;
- 5) розробка діалогової системи та механіки вибору гравця;
- 6) реалізація базових ігрових механік, включаючи дослідження локацій та бойову систему;
- 7) створення та інтеграція графічних і аудіо ресурсів;
- 8) програмна реалізація ігрового застосунку засобами RPG Maker MZ;
- 9) тестування ігрового застосунку та аналіз отриманих результатів.

5. Перелік графічних матеріалів: Презентація

Керівник роботи

(Особистий підпис)

Ірина КАЛІНІНА
(Власне ім'я ПРІЗВИЩЕ)

Здобувач

(Особистий підпис)

Владислав ВРАДІЙ
(Власне ім'я ПРІЗВИЩЕ)

Дата видачі завдання «_22_» __ грудня __ 2025_ р.

КАЛЕНДАРНИЙ ПЛАН кваліфікаційної роботи

Тема: _____ Розробка 2D-гри з елементами візуальної новели на рушії RPG
Maker _____.

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КРБ	21.12.2025	24.12.2025	Виконано
2	Аналіз рушії RPG Maker та прикладів 2D-ігор з елементами візуальної новели	25.12.2025	30.01.2026	Виконано
3	Аналіз предметної області та особливостей жанру	31.01.2026	01.03.2026	Виконано
4	Розробка структури ігрового застосунку, сценарію та проектування ігрових механік	02.03.2026	01.04.2026	Виконано
5	Реалізація гри на рушії RPG Maker, тестування та аналіз результатів	02.04.2026	24.05.2026	Виконано
6	Перший попередній захист КР на засіданні комісії кафедри	25.05.2026	25.05.2026	Виконано
7	Корегування роботи за результатами попереднього захисту	26.05.2026	04.06.2026	Виконано
8	Другий попередній захист КР на засіданні комісії кафедри	05.06.2026	05.06.2026	Виконано
9	Доробка та остаточне оформлення КР	06.06.2026	19.06.2026	Виконано
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	20.06.2026	22.06.2026	Виконано

Керівник роботи _____

(Особистий підпис)

Ірина КАЛІНІНА

(Власне ім'я ПРІЗВИЩЕ)

Здобувач _____

(Особистий підпис)

Владислав ВРАДІЙ

(Власне ім'я ПРІЗВИЩЕ)

Дата складання календарного плану
«28» січня 2026 р.

АНОТАЦІЯ

до кваліфікаційної роботи
здобувача групи 401 ЧНУ ім. Петра Могили

Врадія Владислава Сергійовича

на тему: «**РОЗРОБКА 2D-ГРИ З ЕЛЕМЕНТАМИ ВІЗУАЛЬНОЇ НОВЕЛИ НА
РУШІЇ RPG MAKER**»

Актуальність теми кваліфікаційної роботи зумовлена розвитком інструментів для створення ігор, що дозволяють реалізовувати повноцінні ігрові застосунки без глибоких знань програмування. Поєднання жанрів JRPG та візуальної новели відкриває можливості для створення інтерактивного ігрового досвіду, який поєднує сюжетну глибину з ігровими механіками. Використання рушія RPG Maker MZ дозволяє дослідити сучасні підходи до розробки 2D-ігор та реалізації нелінійного сюжету в межах доступного середовища.

Кваліфікаційна робота присвячена розробці ігрового застосунку на базі RPG Maker MZ, що поєднує елементи візуальної новели та рольової гри. У результаті виконання роботи створено демо версію гри, яка демонструє основні механіки: діалогову систему, вибір гравця, дослідження локацій та базову бойову систему.

Об'єкт роботи: процес створення 2D-гри в жанрі JRPG.

Предмет роботи: методи та засоби реалізації ігрових механік у середовищі RPG Maker MZ.

Мета роботи: розробка 2D-гри у жанрі JRPG з елементами візуальної новели на базі рушія RPG Maker MZ.

У роботі використано методи аналізу предметної області, порівняльного аналізу програмних засобів, проєктування ігрових систем та практичної реалізації ігрових механік у середовищі візуального програмування.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків та переліку джерел.

У першому розділі проведено аналіз предметної області розробки 2D-ігор, розглянуто особливості жанрів JRPG та візуальних новел, виконано огляд ігрових рушіїв та обґрунтовано вибір RPG Maker MZ.

У другому розділі досліджено методи та технології розробки, зокрема реалізацію діалогових систем, механіки вибору гравця, базових ігрових механік та бойової системи.

У третьому розділі описано проектування структури ігрового застосунку, організацію даних та логіку роботи системи.

У четвертому розділі представлено програмну реалізацію гри, результати тестування та аналіз отриманих результатів.

У результаті виконання роботи розроблено демо версію 2D-гри, що реалізує основні ігрові механіки: дослідження локацій, діалогову систему, механіку вибору гравця та базову бойову систему.

Кваліфікаційна робота містить 81 сторінку, 11 таблиць, 48 рисунків та перелік із 31 джерел посилань

Ключові слова: 2D-гра, JRPG, візуальна новела, RPG Maker MZ, розробка гри, ігрові механіки, діалогова система, інтерактивний сюжет.

ABSTRACT

to the qualification work by the student of the group 401 of Petro Mohyla Black Sea National University

Vradii Vladislav Serhiyovych

«DEVELOPMENT OF A 2D GAME WITH ELEMENTS OF A VISUAL NOVEL ON THE RPG MAKER ENGINE»

The relevance of the topic of the qualification work is due to the development of tools for creating games that allow you to implement full-fledged game applications without in-depth programming knowledge. The combination of the JRPG and visual novel genres opens up opportunities for creating an interactive gaming experience that combines plot depth with game mechanics. Using the RPG Maker MZ engine allows you to explore modern approaches to developing 2D games and implementing a nonlinear plot within the capabilities of the engine

The qualification work is dedicated to the development of a game application based on RPG Maker MZ, which combines elements of a visual novel and a role-playing game. As a result of the work, a demo version of the game was created, which demonstrates the main mechanics: a dialogue system, player choice, location exploration and a basic combat system.

Object of the work: the process of 2D game development.

Subject of the work: methods and tools for implementing game mechanics in the RPG Maker MZ environment.

Purpose of the work: development of a 2D game in the JRPG genre with visual novel elements using the RPG Maker MZ engine.

The work uses the following methods of domain analysis, comparative analysis of software tools, game system design, and implementation of game mechanics within a visual programming environment.

The qualification work consists of an introduction, four chapters, conclusions, and a list of references.

The first chapter analyzes the domain of 2D game development, examines the features of JRPG and visual novel genres, reviews existing game engines, and justifies the choice of RPG Maker MZ.

The second chapter explores development methods and technologies, including the implementation of dialogue systems, player choice mechanics, core gameplay mechanics, and a turn-based combat system.

The third chapter describes the design of the game application structure, data organization, and system logic.

The fourth chapter presents the implementation of the game, testing results, and analysis of the obtained results.

As a result of the work, a demo version of a 2D game has been developed, implementing core gameplay mechanics such as exploration, dialogue system, player choice, and a basic combat system.

The qualification work contains 81 pages, 11 tables, 48 figure and a list of 31 reference sources.

Keywords: 2D game, JRPG, visual novel, RPG Maker MZ, game development, game mechanics, dialogue system, interactive narrative.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	4
ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Опис предметної області розробки 2D-ігор	6
1.2 Огляд існуючих рушіїв та інструментів розробки ігор.....	8
1.3 Приклади ігор, створених за допомогою RPG Maker	11
1.4 Еволюція та порівняльний аналіз рушія RPG Maker	15
1.5 Постановка задачі дослідження, специфікація вимог	16
Висновки до розділу 1	17
2 МЕТОДИ ТА ТЕХНОЛОГІЇ РОЗРОБКИ 2D-ГРИ	19
2.1 Особливості розробки ігор на рушії RPG Maker MZ	19
2.2 Методи реалізації ігрових механік	23
2.3 Технології та інструменти розробки	26
2.4 Обмеження рушія RPG Maker MZ	30
Висновки до розділу 2	30
3 ПРОЄКТУВАННЯ ІГРОВОГО ЗАСТОСУНКУ	32
3.1 Структура гри	32
3.2 Проєктування ігрових механік	37
3.3 Демонстрація роботи системи	43
3.4 Аналіз отриманих результатів	57
Висновки до розділу 3	59
4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ	61
4.1 Керівництво користувача	61
4.2 Тестування	65
Висновки до розділу 4	70
ВИСНОВКИ.....	71

Декларація про використанні ШІ. Під час підготовки наукової роботи (академічного тексту) було використано інструмент GPT-5.5. Відповідно до таксономії GAIDeT (2025), наведені нижче завдання були делеговані інструментам генеративного ШІ за повного людського нагляду:

- пошук і систематизація літератури;
- оптимізація ігрових формул та коду;
- очищення даних;
- генерація структур таблиць;
- формулювання підсумків.

Повну відповідальність за фінальний рукопис несе автор.

Інструменти генеративного ШІ не зазначаються як автори та не несуть відповідальності за кінцеві результати.

Декларацію подав: Врадій Владислав Сергійович.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ВН	– Візуальна новела
БД	– База даних
ПЗ	– Програмне забезпечення
ІС	– Інформаційна система
ПК	– Персональний комп'ютер
RPG	– Role-Playing Game
NPC	– Non-Player Character
DEV	– Developer
UI	– User Interface
MP	– Mana points
TP	– Tactical points

ВСТУП

Жанр JRPG дуже щільно пов'язаний з іграми у фентезійному сетингу та партійними рольовими в цілому.

RPG Maker – це досить давно відомий рушій, який дає починаючим розробникам доступ до широкого спектру можливостей, включаючи інтуїтивне візуальне програмування, підтримку зображень та аудіо, а також готові ігрові системи та механіки. При бажанні є можливість використовувати власні плагіни та системи, або взагалі переробити всі елементи з нуля.

Робота над 2D-грою в жанрі JRPG включає кілька етапів: розробку сценарію-концепту, дизайн персонажів і сетингу, а також створення ігрових локацій і ситуацій, в яких гравець братиме участь. Але для цієї роботи буде створена лише демо версія, з укороченим сюжетом та обмеженою варіативністю.

Об'єкт роботи: процес розробки 2D-гри з елементами візуальної новели на рушії RPG Maker.

Предмет роботи: технології та засоби розробки ігрового застосунку у жанрі JRPG.

Мета: розробка 2D-гри у жанрі JRPG з елементами візуальної новели, на рушії RPG Maker.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- дослідити можливості рушія RPG Maker MZ;
- розробити структуру та сценарій ігрового застосунку;
- створити чи знайти у відкритому доступі графічні та аудіо матеріали для гри;
- реалізувати прототип гри з використанням засобів рушія;
- здійснити аналіз і тестування реалізованих механік та оцінити їх ефективність;
- провести оптимізацію та забезпечити якісну роботу гри.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області розробки 2D-ігор

2D-ігри є одним із базових напрямів розвитку ігрової індустрії, який передбачає відображення самої гри у двох вимірах – «збоку» (часто як side-scroller), або «зверху» (частіше як isometric). На відміну від тривимірних ігор, 2D-ігри зазвичай характеризуються спрощеною графічною складовою, меншою вимогливістю до ресурсів пристрою та відносною простотою реалізації, що робить їх популярними серед незалежних чи починаючих розробників.

Самі 2D-ігри можуть реалізовувати різні жанрові підходи, серед яких займають свої місце також і рольові ігри (RPG), зокрема піджанр японських рольових ігор (JRPG). JRPG поєднує в собі набір із заздалегідь визначених персонажів, сюжетно орієнтований геймплей, зазвичай покрокову бойову систему, а також дослідження ігрового світу. Проте незважаючи на назву, вони не обов'язково повинні мати елементи RPG чи бути створеними в Японії. Одну з класичних JRPG можна побачити на прикладі FINAL FANTASY VI [1].



Рисунок 1.1 – Приклад 2D гри в жанрі JRPG [1]

Окремим напрямом у межах 2D-ігор є жанр візуальної новели. Він зосереджений на подачі текстової інформації, діалогів та сюжетних подій із використанням статичних чи рідше анімованих зображень, текстових блоків та звуків. Основною особливістю візуальних новел є надання гравцеві можливості впливати на розвиток подій через систему вибору, що формує нелінійний сюжет і декілька варіантів розвитку історії.

Проте ступінь інтерактивності в таких іграх зазвичай низький, тому деякі розробники можуть вбудовувати в них власні міні-ігри чи окремі механіки (як наприклад жанр Point and Click). Візуальні новели часто використовуються для глибокого розкриття персонажів та створення емоційного зв'язку. Прикладом такого підходу можна назвати багато робіт, одну з яких, «STEINS;GATE», можна побачити на рис. 1.2.



Рисунок 1.2 – Приклад кадру візуальної новели

Поєднання елементів JRPG та візуальної новели є поширеним підходом у розробці ігор, оскільки дозволяє об'єднати інтерактивність ігрового процесу з розвиненою сюжетною складовою, що сприяє підвищенню залученості гравця та різноманітності можливих ігрових відрізків.

1.2 Огляд існуючих рушіїв та інструментів розробки ігор

На сучасний момент існує багато інструментів для розробки ігор, які відрізняються як за рівнем складності та функціональністю, так і за ціною чи цільовою аудиторією. Вибір рушія, по суті своїй, залежить від поставлених задач, типу гри, а також рівня підготовки самого розробника.

Одним із найвідоміших та найпоширеніших рушіїв є **Unity**. Він є універсальним середовищем розробки, яке використовується для створення як 2D, так і 3D-ігор. Unity підтримує мову програмування C#, має велику кількість навчальних матеріалів, готових бібліотек та додаткових інструментів для роботи з графікою, фізикою і анімацією. Проте його використання потребує базових знань програмування та розуміння структури ігрових систем, що може ускладнювати процес розробки для початківців.

Іншим популярним інструментом є **Godot** – безкоштовне середовище з відкритим вихідним кодом. Він також підтримує створення 2D та 3D-ігор, і використовує власну мову програмування GDScript, яка за синтаксисом нагадує Python. Godot часто обирають за його невеликий розмір, гнучкість та активну спільноту користувачів. Однак більшість механік у ньому також необхідно реалізовувати самостійно, що знову-таки може бути складним для починаючих.

Для розробки ігор у жанрі візуальних новел широко використовується **Ren'Py**. Цей рушій орієнтований саме на створення інтерактивних історій із розгалуженим сюжетом та системою вибору. В ньому можна швидко реалізовувати діалогові сцени, керувати переходами між подіями та працювати з графічними ресурсами, навіть без складного налаштування систем. Водночас можливості рушія в основному зосереджені саме на жанрі візуальної новели, що ускладнює реалізацію складніших ігрових механік, таких як дослідження локацій або бойова система.

Детальніший порівняльний наліз рушіїв, в рамках потрібного жанру та обраної теми роботи, можна побачити в табл. 1.1.

Таблиця 1.1 – Порівняльний аналіз рушіїв для розробка 2D-гри

Характеристика	Unity	Godot	Ren'Py	RPG Maker
Тип рушія	Універсальний (2D/3D)	Універсальний (2D/3D)	Спеціалізований (ВН)	Спеціалізований (JRPG)
Мова програмування	C#	GScript, C#	Python	JavaScript (опціонально)
Рівень складності	Високий	Середній	Низький	Низький
Необхідність програмування	Обов'язкова	Бажана	Мінімальна	Мінімальна
Підтримка 2D	Повна	Повна	Обмежена	Повна
Підтримка 3D	Повна	Повна	Відсутня	Відсутня
Вбудовані механіки	Обмежені (потрібна реалізація)	Обмежені	Діалоги, вибір	Бій, інвентар, діалоги
Орієнтація	Універсальна	Універсальна	Візуальні новели	JRPG
Гнучкість	Висока	Висока	Низька	Середня
Швидкість розробки	Низька	Середня	Висока	Висока
Поріг входження	Високий	Середній	Низький	Низький

Як видно з таблиці 1.1, універсальні рушії, такі як Unity та Godot, надають значну варіативність і широкі можливості, проте потребують знань програмування та самостійної реалізації базових ігрових механік.

Ren'Py є зручним, але занадто вузькоспеціалізованим рішенням, оскільки призначений для створення конкретно візуальних новел, що обмежує його використання для реалізації складніших ігрових систем.

RPG Maker займає проміжне положення, поєднуючи достатню простоту використання з наявністю вбудованих механік, необхідних для створення 2D-ігор у жанрі JRPG.

Саме тому у контексті даної роботи було обрано рушій **RPG Maker**, а саме його останню версію «MZ». Детальніше про різницю між останніми двома версіями буде зазначено у розділі 1.4.

Основною причиною вибору саме цього рушія є його орієнтованість на створення 2D-ігор у жанрі JRPG, наявність вбудованих механік покрокових боїв, системи подій та інструментів візуального програмування.

Це дозволить реалізувати основні елементи гри без необхідності глибокого знання мов програмування, що і стало одним з основних критеріїв вибору. Інтерфейс рушія та приклад діалогового вікна можна побачити на рис. 1.3.

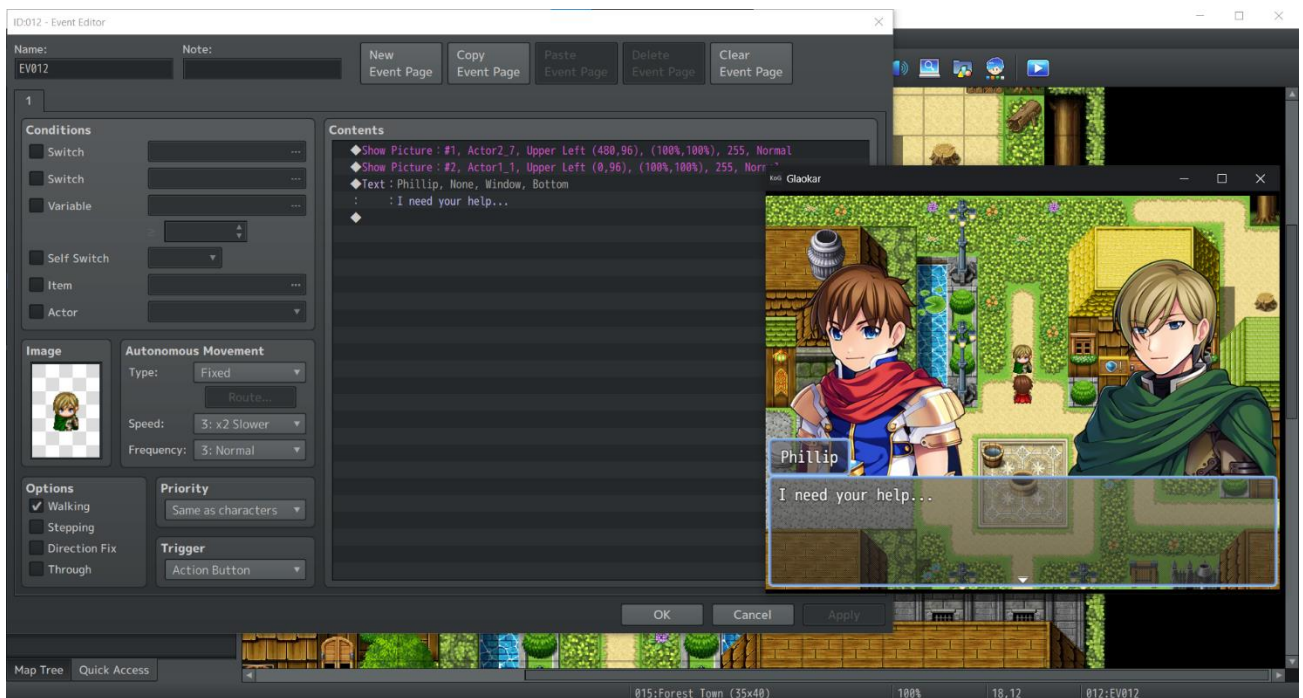


Рисунок 1.3 – Інтерфейс RPG Maker MZ

Підсумовуючи, вибір RPG Maker MZ зумовлений поєднанням доступності, наявності готових інструментів для реалізації ігрових механік та відповідністю поставленим задачам розробки, що робить його доцільним рішенням для створення прототипу 2D-гри з елементами візуальної новели, без необхідності розробки базових систем з нуля.

1.3 Приклади ігор, створених за допомогою RPG Maker

Рушій RPG Maker широко використовується вже багато років як незалежними розробниками, так і невеликими командами для створення повноцінних ігрових проєктів. Серед відомих ігор, створених за допомогою різних версій RPG Maker, можна виділити багато прикладів.

Таблиця 1.2 – Аналіз гри To the Moon

Рік виходу	2011
Розробник:	Freebird Games
Жанр	пригодницька гра, сюжетно-орієнтована RPG
Рушій	RPG Maker XP

«To the Moon» є прикладом гри, де основний акцент зроблено на сюжеті та емоційному впливі на гравця. Ігровий процес мінімізований і зводиться до дослідження локацій та взаємодії з об'єктами, тоді як ключову роль відіграють діалоги та поступове розкриття історії персонажів [2].

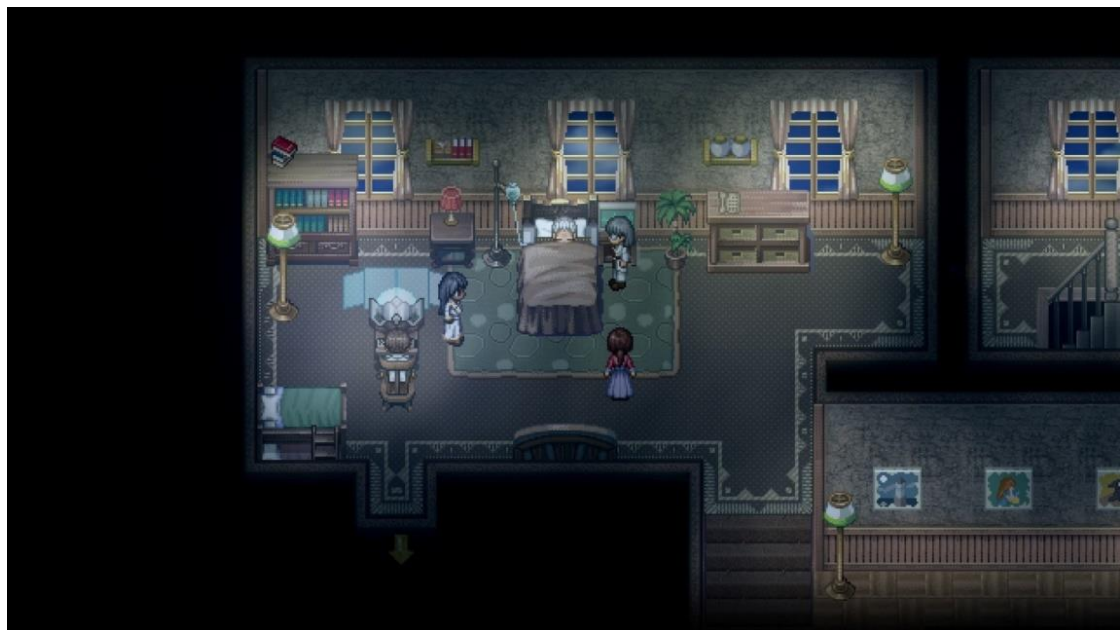


Рисунок 1.4 – Ключова сцена першої половини To the Moon [2]

Гра має лінійну структуру сюжету, що подається через послідовність подій та спогадів. Це робить її близькою до жанру візуальної новели, хоча вона зберігає елементи дослідження, характерні для JRPG.

Таблиця 1.3 – Аналіз гри Ib

Рік виходу	2012 (римейк у 2022)
Розробник:	Kougi
Жанр	пригодницький хорор
Рушій	RPG Maker 2000 (RPG Maker MV у римейку)

Гра «Ib» демонструє можливості створення атмосферних ігор із мінімальним використанням складних механік. Основу геймплею становлять дослідження локацій, розв’язання головоломок та взаємодія з оточенням [3].



Рисунок 1.5 – Діалогова сцена в Ib 2022 [3]

Особливістю гри є нелінійність, яка проявляється через декілька можливих кінцівок, що залежать від дій гравця. Атмосфера досягається за рахунок візуального стилю, музичного супроводу та обмеженого інтерфейсу.

Таблиця 1.4 – Аналіз гри OneShot

Рік виходу	2016
Розробник:	Future Cat
Жанр	пригодницька гра
Рушій	RPG Maker XP

«OneShot» є прикладом нестандартного використання рушія RPG Maker. Гра активно взаємодіє з користувачем, виходячи за межі звичного ігрового процесу, зокрема через пряме звернення до гравця та використання файлів його комп'ютера [4].



Рисунок 1.6 – Одна з локацій та зовнішній вигляд OneShot [4]

Особливістю OneShot є мета-нарратив, що руйнує «четверту стіну» та створює унікальний ігровий досвід. При цьому базові механіки залишаються доволі простими, зосереджуючись на самій історії та подорожі.

Таблиця 1.5 – Аналіз гри Omori

Рік виходу	2020
Розробник:	OMOCAT
Жанр	JRPG, психологічна RPG
Рушій	RPG Maker MV

«Omori» є прикладом повноцінної JRPG, створеної на рушії RPG Maker, яка поєднує сюжетно-орієнтований підхід із доволі розвиненою бойовою системою [5].



Рисунок 1.7 – Бойовий екран в OMORI [5]

Бойова система гри базується на покроковій взаємодії між персонажами та супротивниками, де кожен учасник має набір дій, включаючи атаки, навички та взаємодію з емоційним станом персонажів. Помітною особливістю є механіка емоцій, яка впливає на ефективність дій у бою, що додає стратегічної глибини.

Як видно з розглянутих прикладів, рушій RPG Maker може бути використаний для реалізації як простих «сюжетних» проєктів, так і більш складних ігор із розвиненими механіками.

1.4 Еволюція та порівняльний аналіз рушія RPG Maker

Рушій RPG Maker є одним із найстаріших та найвідоміших інструментів для створення 2D-ігор, історія якого розпочинається ще у 1990-х роках. За цей час він зазнав значних змін, розвинувшись від простих конструкторів рольових ігор до повноцінного середовища розробки. Основною ідеєю рушія завжди залишалась можливість створення ігор без глибоких знань програмування за допомогою візуальних інструментів.

Серед останніх версій рушія найбільш поширеними є **RPG Maker MV** та **RPG Maker MZ**. Обидві версії орієнтовані на створення 2D-ігор у жанрі JRPG та мають майже ідентичну архітектуру, яка базується на використанні системи подій та візуального програмування.

RPG Maker MV (2015) є більш ранньою версією рушія, яка отримала широку популярність завдяки стабільності та великій кількості доступних плагінів і ресурсів. Вона використовує технологію JavaScript для розширення функціональності та дозволяє створювати ігри з можливістю експорту на різні платформи.

RPG Maker MZ (2020) є більш новою версією рушія, яка розвиває можливості попередника та пропонує покращений інтерфейс користувача, оптимізовану продуктивність та розширений функціонал стандартних інструментів. Зокрема, у MZ було вдосконалено систему керування ресурсами, анімаціями та подіями, що спрощує процес розробки та зменшує потребу у використанні сторонніх плагінів.

Основною перевагою RPG Maker MZ є більш зручний робочий процес та наявність сучасніших інструментів, що робить його більш придатним для використання та швидкої розробки ігрових прототипів. З огляду на це, для реалізації даної роботи було обрано саме версію RPG Maker MZ.

Таблиця 1.6 – Порівняння RPG Maker MV та MZ

Характеристика	MV	MZ
Рік випуску	2015	2020
Мова розширення	JavaScript	JavaScript
Інтерфейс	Базовий	Покращений
Продуктивність	Стабільна	Оптимізована
Система анімацій	Обмежена	Розширена
Залежність від плагінів	Висока	Нижча
Зручність роботи	Середня	Вища

1.5 Постановка задачі дослідження, специфікація вимог

Актуальність теми полягає у зростанні популярності інструментів для розробки ігор, які дозволяють створювати повноцінні ігрові застосунки без глибоких знань програмування. Використання таких рушіїв, як RPG Maker, дає можливість дослідити ефективність сучасних підходів до створення 2D-ігор та реалізації інтерактивних сюжетів із використанням вбудованих механік рушія.

Метою роботи є розробка 2D-гри на рушії RPG Maker, що поєднує елементи візуальної новели та рольової гри, а також дослідження особливостей реалізації нелінійного сюжету, діалогових систем і базових механік настільно-рольових ігор у цифровому середовищі.

Об'єкт і предмет

Об'єктом дослідження є процес розробки 2D-ігор. Предметом дослідження є методи та засоби реалізації ігрових механік у середовищі RPG Maker MZ.

Призначення системи

Ігровий застосунок, що розробляється, призначений для створення інтерактивного ігрового досвіду, який поєднує елементи візуальної новели та рольової

гри. Основна увага приділяється подачі сюжету, взаємодії з персонажами та реалізації базових ігрових механік.

Основні функції системи

- взаємодія з ігровим світом через дослідження локацій;
- ведення діалогів із персонажами;
- вибір дій, що впливають на розвиток сюжету;
- участь у покрокових боях;
- збереження та завантаження ігрового прогресу.

Обмеження системи, вимоги до програмного забезпечення

Гра розробляється як прототип, тому її обсяг обмежений кількістю локацій, персонажів та загальної інтерактивності. Частина механік реалізується у спрощеному вигляді. Розробка здійснюється з використанням рушія RPG Maker MZ, що забезпечує реалізацію основних ігрових механік без необхідності глибокого програмування.

Висновки до розділу 1

У першому розділі було розглянуто предметну область розробки 2D-ігор, зокрема особливості жанру JRPG та візуальних новел, а також можливості їх поєднання в межах одного ігрового застосунку.

Проведено огляд сучасних інструментів розробки ігор, зокрема рушіїв Unity, Godot та Ren'Py, що дозволило визначити їхні основні можливості та обмеження. На основі проведеного аналізу обґрунтовано доцільність використання рушія RPG Maker, який найбільш відповідає поставленим задачам завдяки наявності вбудованих механік, простоті використання та орієнтації на створення 2D-ігор.

Також було здійснено порівняльний аналіз версій RPG Maker MV та RPG Maker MZ, що дозволило визначити переваги сучаснішої версії рушія, зокрема покращений інтерфейс, оптимізовану продуктивність та розширені можливості роботи з подіями й ресурсами.

Окрему увагу приділено аналізу прикладів ігор, створених за допомогою RPG Maker, які демонструють різні підходи до реалізації ігрового процесу – від сюжетно-орієнтованих проєктів до ігор із розвиненими механіками, зокрема покроковими бойовими системами. Це підтверджує гнучкість рушія та можливість його використання для реалізації проєктів різної складності.

Крім того, було сформульовано основні вимоги до ігрового застосунку, що розробляється, включаючи його функціональні можливості та обмеження, що створює основу для подальшого проєктування та реалізації системи.

У результаті проведеного аналізу визначено мету, об'єкт, предмет дослідження та сформульовано основні завдання дипломної роботи, що окреслюють подальші етапи розробки ігрового застосунку.

2 МЕТОДИ ТА ТЕХНОЛОГІЇ РОЗРОБКИ 2D-ГРИ

2.1 Особливості розробки ігор на рушії RPG Maker MZ

Рушій RPG Maker MZ є спеціалізованим середовищем для створення 2D-ігор, яке орієнтоване на розробників без глибоких знань програмування. Основною особливістю рушія є використання візуального програмування, що реалізується через систему подій, а також наявність вбудованих інструментів для створення ігрових локацій, персонажів та механік.

У межах даного підрозділу розглянуто основні компоненти рушія, які використовуються при розробці ігрового застосунку.

Система подій (Event System)

Система подій є ключовим елементом рушія, що дозволяє реалізовувати логіку гри без необхідності написання програмного коду. Події представляють собою набір команд, що виконуються за певних умов або при взаємодії гравця з об'єктами. Процес створення нової події видно на рис. 2.1.



Рисунок 2.1 – Створення нової події (new event) на певному тайлі мапи

Кожна подія може містити декілька сторінок, між якими здійснюється перемикання залежно від виконання умов, таких як змінні, перемикачі або стан гри. Це дозволяє реалізовувати складну поведінку об'єктів, наприклад:

- зміну діалогів після певних подій;
- одноразове отримання предметів;
- реакцію персонажів на дії гравця.

Система подій є основним інструментом реалізації ігрової логіки в рамках даної роботи і рушії в цілому. Вікно редактору подій видно на рис. 2.2.

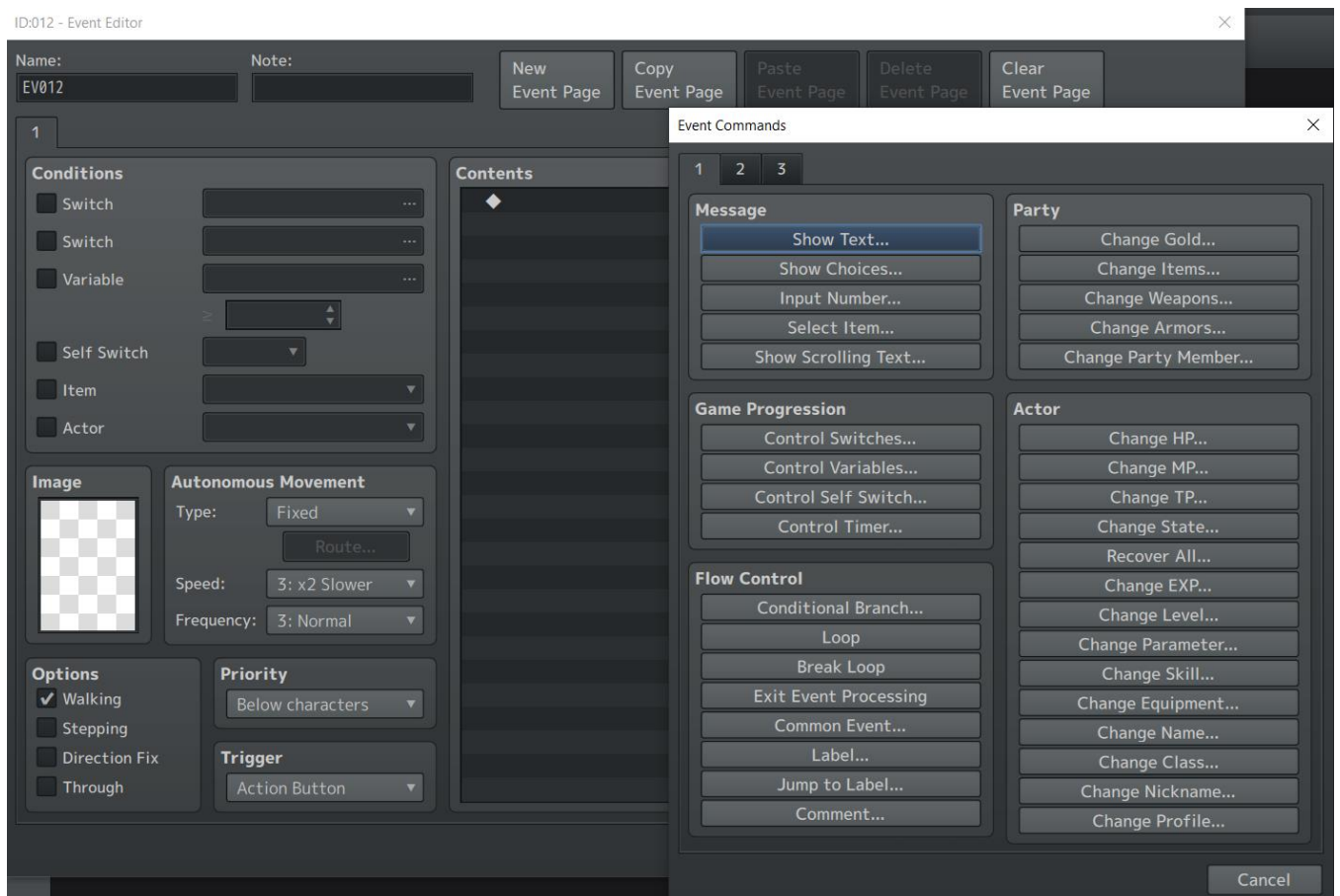


Рисунок 2.2 – Редактор подій, можливість їх налаштування

Мапи та тайлсети (Maps і Tilesets)

Ігровий світ у RPG Maker будується на основі мап (maps), які складаються з тайлів – невеликих графічних елементів, що формують середовище.

Тайлсети (tilesets) являють собою набори графічних ресурсів, які використовуються для побудови локацій. Вони включають елементи ландшафту, будівель, інтер'єрів та інших об'єктів. Існує можливість створення власних графічних

елементів та завантаження готових із різних джерел. Набір базових тайлів для «Поля» можна побачити на рис. 2.3

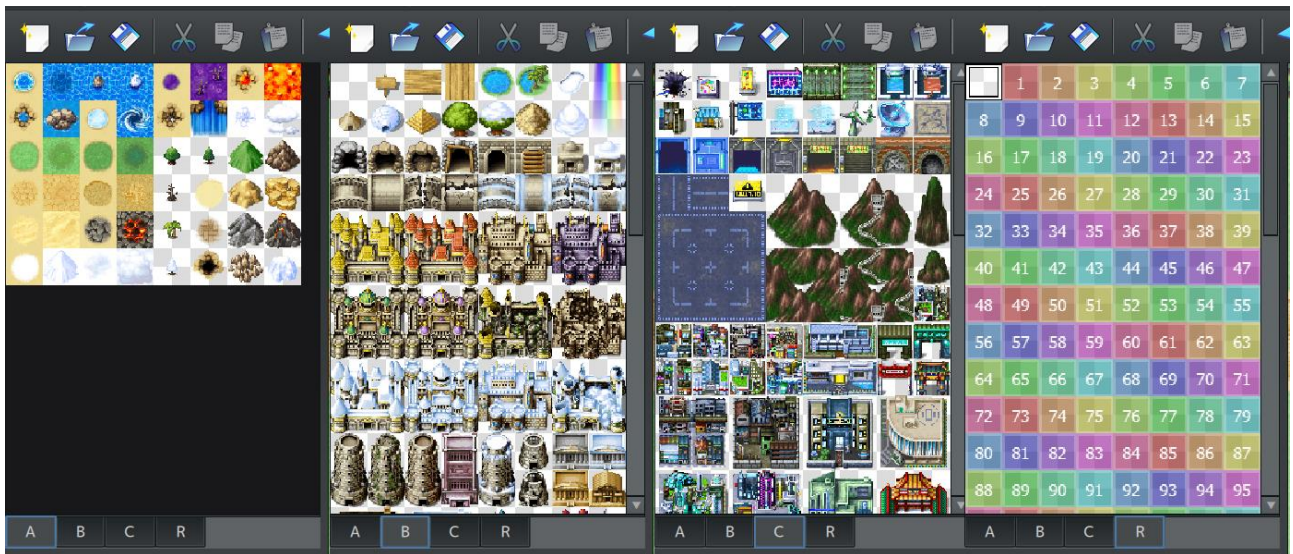


Рисунок 2.3 – Набір тайлів «Поле», та сторінка «Регіонів»

Кожна мапа може використовувати певний тайлсет, що визначає її візуальний стиль та основний напрям. Це дозволяє створювати різноманітні локації, такі як природні середовища, міські території чи приміщення. Їх перелік видно на рис. 2.4.

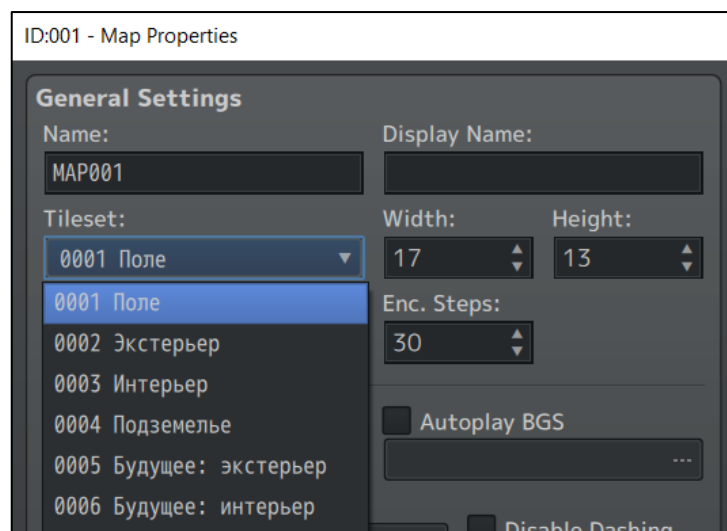


Рисунок 2.4 – Можливі набори тайлсетів

Базові ігрові механіки рушія

RPG Maker MZ надає набір вбудованих механік, таких як наприклад:

- система переміщення персонажа по мапі;
- взаємодія з об'єктами, інвентар та предмети;
- діалогова система;
- покрокова бойова система.



Рисунок 2.5 – Приклади можливостей рушії, вікна спорядження та бою

Як видно на рис. 2.5, рушій навіть в базовій комплектації має доволі детально оформлені системи та механіки, необхідні для готової гри. Це дозволяє реалізовувати

її логіку без написання програмного коду або з мінімальним використанням скриптів, і зосередитися на реалізації логіки та сценарію.

2.2 Методи реалізації ігрових механік

Діалогові системи

Діалогова система є одним із ключових елементів гри, оскільки саме через неї відбувається передача сюжетної інформації та взаємодія гравця з персонажами і об'єктами.

Самі діалоги реалізуються за допомогою подій (events), що містять команди відображення тексту. Для кожного персонажа чи об'єкта створюється окрема подія, яка активується при певній взаємодії від гравця. Приклад реалізації діалогу можна побачити на рис. 2.6 та 2.7.

```
◆Text : None, GL_Elanor(0), Window, Bottom  
:      : - How's the weather up there, big guy?  
◆Text : None, GL_Grid(0), Window, Bottom  
:      : - Windy, but the night weather will be fine
```

Рисунок 2.6 – Вигляд діалогу в Редакторі подій



Рисунок 2.7 – Вигляд діалогового вікна у грі

Загальну структуру можна ускладнити та поглибити багатьма способами, від простої реалізації варіантів питань та відповідей через цикл, до використання механізму перемикачів (switches), в яких реалізовані різні реакції на певні дії чи вибори діалогових опцій.

Реалізація вибору гравця

Стан гри формується через перемикачі (switches, що мають значення ON/OFF), змінні (variables, які мають числові значення) та сторінки подій (умови). Для невеликих сцен чи локацій може бути достатньо лише локальних перемикачів та сторінок подій, коли як для важливих виборів чи складніших систем знадобляться окремі глобальні змінні.

На відміну від простих перемикачів, за допомогою Змінних можна зберігати числові значення та застосовувати їх для більш гнучкої логіки, наприклад підрахунку виборів або визначення стану гри. Доступні опції для зміни стану гри можна побачити на рис. 2.8.

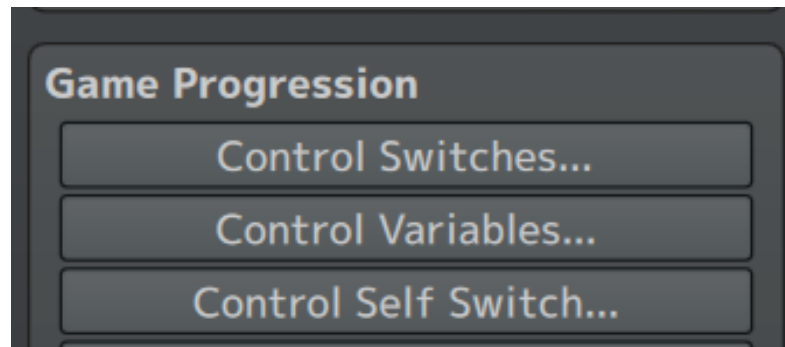


Рисунок 2.8 – Перелік можливих опцій для реалізації перемикачів

З їхньою допомогою, певний варіант вибору може призводити до виконання окремого набору подій, а також враховувати попередні дії гравця вже в майбутніх подіях, що дозволяє:

- змінювати діалоги в залежності від окремих подій чи виборів;
- запускати нові події;
- впливати на подальший розвиток гри.

Реалізація бойової системи

Бойова система реалізується з використанням вбудованих механік RPG Maker MZ, які підтримують покрокові бої у форматі JRPG. У межах даної роботи використовується базова конфігурація бойової системи з певними змінами параметрів персонажів, такими як:

- кількість здоров'я та характеристики, занижені з сотень до десятків;
- сила атаки зброї, також з врахуванням зниження загальних чисел;
- набір доступних дій, в рамках власноруч створених класів персонажів.

В рушії вже є вбудований набір класів, здібностей, станів, предметів, зброї та броні, проте їх можна легко змінити чи доповнити у Базі даних проекту через відповідні поля. Базовий вигляд елемента зброї можна побачити на рис. 2.9.

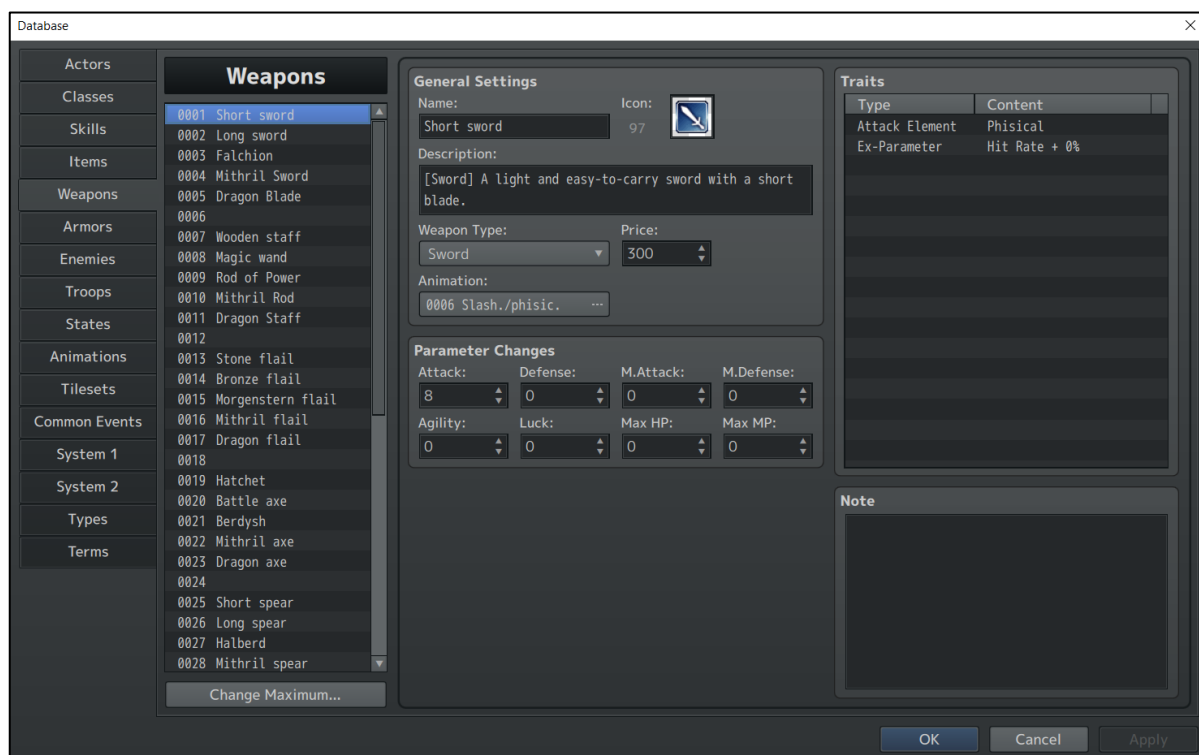


Рисунок 2.9 – Опис та параметри зброї в Базі даних

Бойові сцени ініціюються за допомогою подій, що викликають стандартну команду запуску бою (Управління сценами → Битва). Це може бути проста взаємодія зі спрайтом ворога на мапі, вибір опції в діалозі, прохід по тайлу і так далі.

У межах даної роботи бойова система використовується як допоміжна механіка для урізноманітнення ігрового процесу, хоча і була зроблена спроба перенести досвід настільно-рольової кампанії.

Варто також зазначити, що багато ігор на рушії доповнюють чи замінюють бої загадками, або взагалі сценами погоні, що найбільш притаманно жанру хоррору.

2.3 Технології та інструменти розробки

Основну роль у процесі розробки відіграє ігровий рушій, а також допоміжні графічні та аудіо матеріали.

Ігровий рушій

Основним інструментом розробки є рушій RPG Maker MZ, а саме його безкоштовна 30-денна пробна версія [6].

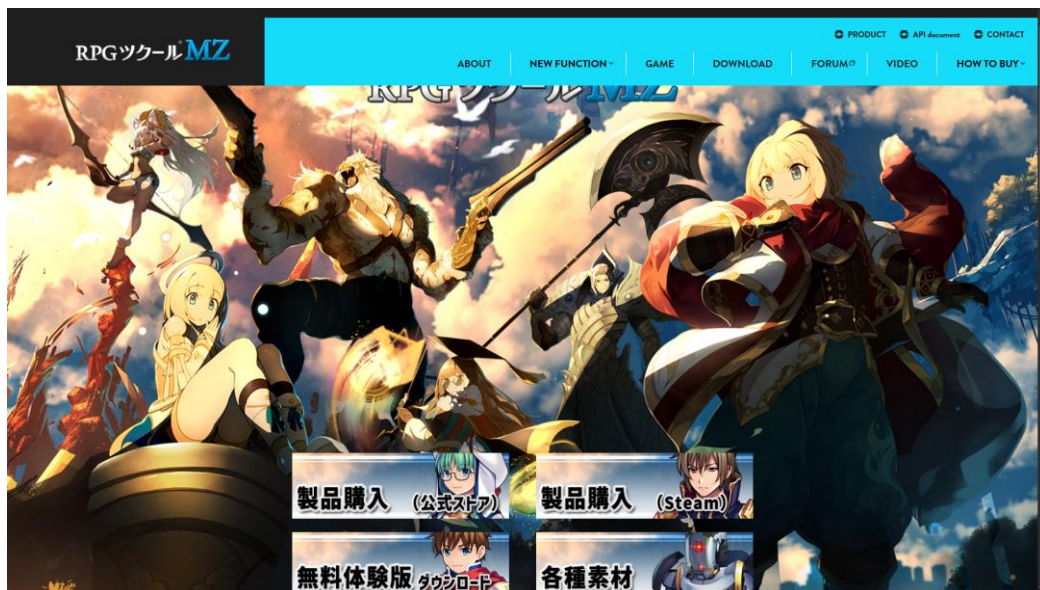


Рисунок 2.10 – Головна сторінка офіційного сайту RPG Maker MZ [6]

Сам рушій забезпечує:

- систему візуального програмування на основі подій та редактор мап;
- базу даних персонажів, предметів, навичок, графічних та аудіо-файлів;
- вбудовану бойову систему, покрокову чи в реальному часі;
- інструменти для роботи з діалогами, вибором гравця та плагінами на js.

Використання RPG Maker MZ дозволяє значно скоротити час розробки, оскільки більшість базових механік уже реалізовано у стандартному функціоналі рушії. Проте за потреби його функціональність може бути розширена за допомогою плагінів, що дозволяють додавати нові механіки або змінювати поведінку існуючих систем.

Графічні ресурси

Графічне оформлення гри базується на використанні тайлсетів, спрайтів персонажів та ілюстрацій.

У рамках проекту використовуються:

- стандартні графічні ресурси рушії;
- додаткові тайлсети, отримані з відкритих джерел;
- власні або відредаговані зображення.

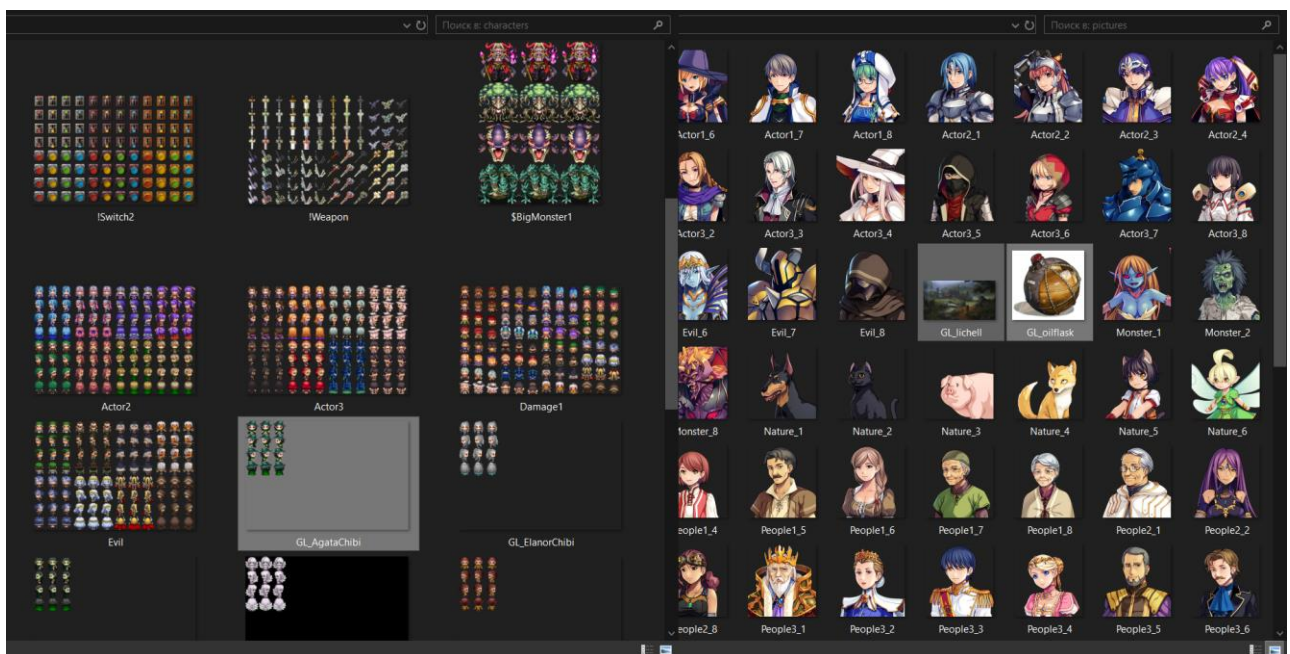


Рисунок 2.11 – Папки /img/characters та /img/faces

Як видно на рис 2.11, до великого переліку вже готових елементів можна легко додати власні зображення з розширенням PNG, чи імпортувати готові із різних вільних джерел.

Рушій також має вбудований генератор-редактор персонажів, який може легко зберігати та імпортувати отриманого персонажа в різних формах – аватар, chibi-версія персонажу та боковий спрайт для однієї з варіацій відображення бою. Зовнішній вигляд генератору можна побачити на рис. 2.12.

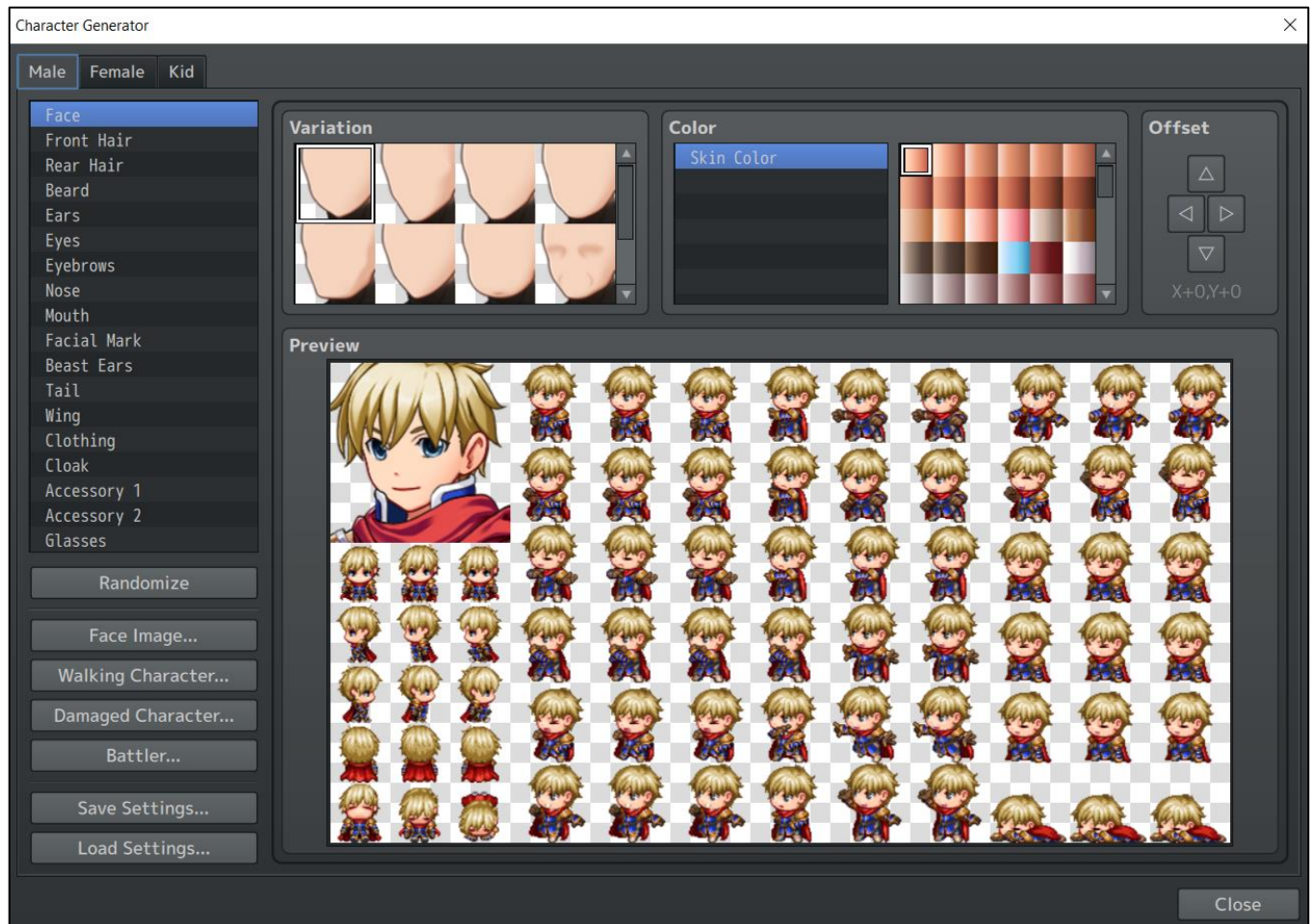


Рисунок 2.12 – Вбудований генератор персонажів

Якщо тайлсети застосовуються для побудови ігрових локацій, то спрайти використовуються для відображення персонажів на мапі. Також додаткові ілюстрації можуть використовуватись під час діалогових сцен, для покращення візуального сприйняття.

Аудіо супровід

Аудіо супровід відіграє важливу роль у створенні атмосфери гри, задаючи настрій і контекст сцени, включаючи в себе:

- фонову музику;
- звукові ефекти;
- звуки взаємодії з об'єктами.

У проєкті використовуються стандартні аудіо ресурси рушія, а також додаткові файли, отримані з відкритих джерел. Вміст папки \audio\bgm та приклад реалізації запуску фонові музики можна побачити на рис. 2.13.

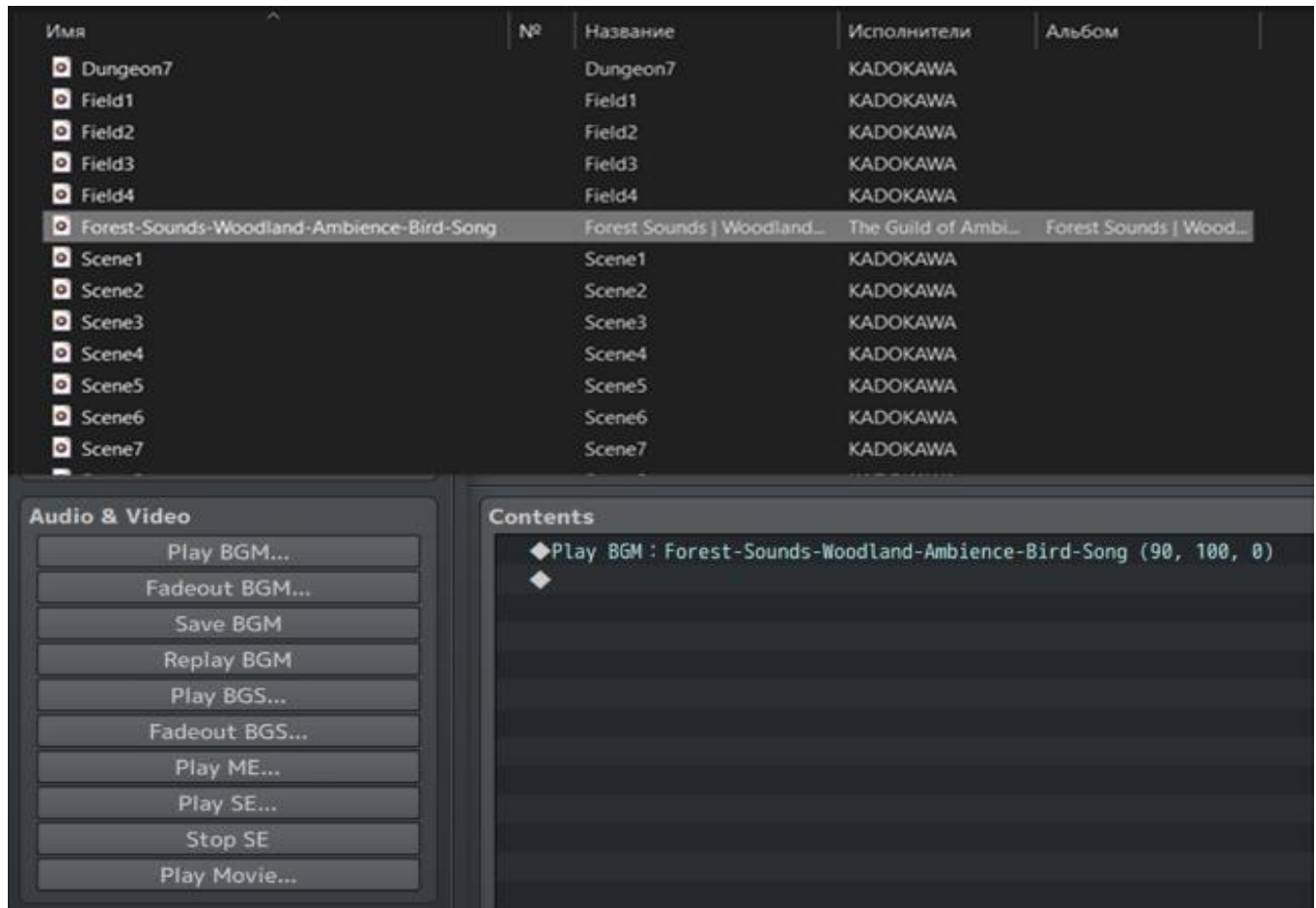


Рисунок 2.13 – Вміст папки bgm та запуск аудіо через редактор подій – пусте зображення, пріоритет «під персонажем» та тригер на запуск як «паралельно»

Застосування звукового супроводу дозволяє підвищити рівень занурення у ігровий процес та підсилити емоційне сприйняття подій.

2.4 Обмеження рушія RPG Maker MZ

Незважаючи на значні переваги рушія RPG Maker, зокрема простоту використання та наявність великої кількості вбудованих інструментів, він також має певні обмеження, які слід враховувати під час розробки.

Одним із основних обмежень є відносно низька гнучкість стандартного функціоналу без використання додаткових плагінів. Базові механіки рушія орієнтовані переважно на класичні 2D JRPG, що може ускладнити реалізацію нестандартних ігрових рішень або унікальних механік.

Також важливою особливістю є залежність логіки гри від системи подій. Хоча вона дозволяє реалізовувати складну поведінку об'єктів без написання програмного коду, при збільшенні обсягу проєкту це може призводити до ускладнення структури подій та зниження зручності їх підтримки, а в найгіршій ситуації навіть до «поломки скриптів», що може зруйнувати проходження гри.

Крім того, обмеження можуть виникати у сфері продуктивності та оптимізації, особливо при використанні великої кількості подій, анімацій або сторонніх ресурсів. Це потребує більш уважного підходу до організації і оптимізації ігрового процесу та ресурсів.

Таким чином, RPG Maker MZ є ефективним інструментом для створення прототипів, невеликих і середніх за обсягом ігор, однак при розробці більш складних проєктів може виникнути потреба у використанні додаткових інструментів або інших ігрових рушіїв.

Висновки до розділу 2

У другому розділі було розглянуто методи та технології розробки 2D-гри на рушії RPG Maker MZ. Проаналізовано особливості роботи рушія, зокрема систему подій як основний інструмент реалізації ігрової логіки, а також принципи побудови ігрового середовища за допомогою карт і тайлсетів.

Досліджено методи реалізації основних ігрових механік, серед яких діалогові системи, механіка вибору гравця та покрокова бойова система. Встановлено, що використання перемикачів, змінних та сторінок подій дозволяє реалізувати базову нелінійність та адаптивну поведінку ігрових об'єктів без застосування складних програмних рішень.

Окрему увагу приділено технологічним аспектам розробки, зокрема використанню графічних і аудіо ресурсів. Визначено, що RPG Maker MZ забезпечує достатній набір інструментів для створення повноцінного ігрового прототипу.

Разом із цим було визначено основні обмеження рушія, пов'язані з гнучкістю реалізації нестандартних механік, організацією подій та оптимізацією продуктивності. Проте з урахуванням поставлених задач встановлено, що зазначені обмеження не є критичними для реалізації демонстраційної версії гри.

3 ПРОЄКТУВАННЯ ІГРОВОГО ЗАСТОСУНКУ

3.1 Структура гри

Організація локацій

В своїй основі гра складається з набору мап, в яких відбуваються певні події. В даному проєкті було зроблено три основних «ключових» локації, одна з яких має переходи на невеликі під-локації, які зазвичай є інтер'єром певних будівель. Схему гри можна побачити на рис. 3.1.

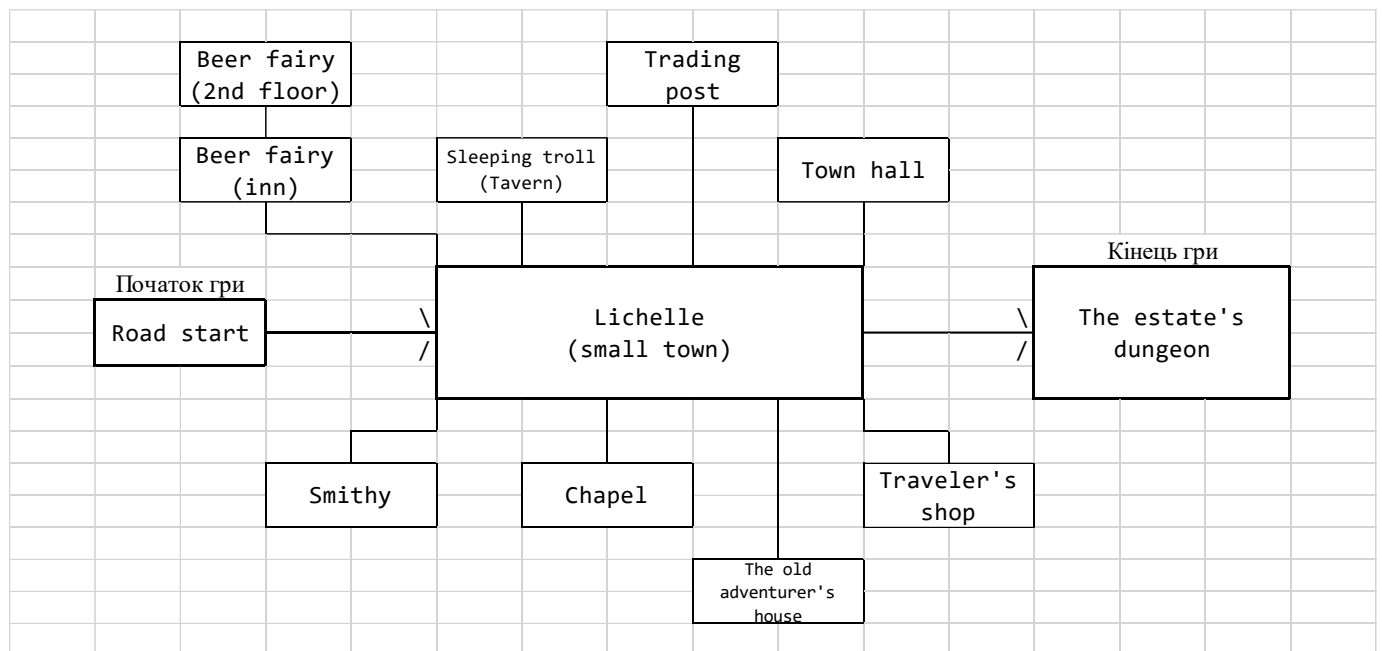


Рисунок 3.1 – Структура ігрового застосунку

Самі локації організуються в якості окремих мап, кожна зі своїм набором графіки (тайлсетом) – Поле, Інтер'єр, Екстер'єр чи Підземелля. Заповнюються вони за допомогою двох режимів редагування – малювання мапи та роботи з подіями.

Переходи між мапами робляться за допомогою Подій, та можуть викликатись коли та як розробнику потрібно – після вибору в діалозі, при взаємодії з об'єктом та багатьох інших випадках.

В зазначених раніше локаціях були зроблені наступні події, описані сценаріями в табл. 3.1:

Таблиця 3.1 – Список створених подій з їх сценаріями

Локація	Доступні взаємодії, події та їхні сценарії
1	2
Початок на головному тракті	<ol style="list-style-type: none"> 1) табличка, що надає гравцю експозицію; 2) персонажі-майбутні напарники, з якими можна поспілкуватись та дізнатись їхні характери; 3) візок з сюжетним предметом, на який видає завдання один з NPC на локації; 4) кінь, який стає активатором переносу в наступну локацію, після завершення певних подій.
Містечко, центральна локація демо-версії	<ol style="list-style-type: none"> 1) (при першому заході) кат-сцена з зупинкою, що направляє гравця в потрібну будівлю; 2) каплиця, що дає бонус до Удачі при пожертвуванні грошей; 3) наймач персонажа гравця, який служить для пояснення чим гравець та його напарники займаються в глобальному плані; 4) двері-переходи на інші локації.
Заїжджий двір	<ol style="list-style-type: none"> 1) (при першому заході) кат-сцена лайки одного з напарників і власника, що розкриває характер першого; 2) власник, в якого можна зняти кімнату, купити їжу та дізнатись новини, а також взяти завдання на розбирання з бандитами; 3) (після сну) приєднання одного з напарників до гравця в команду.
Інші локації	<ol style="list-style-type: none"> 1) кузня, де можна купити зброю чи броню; 2) торгова лавка, де можна купити та продати різні товари; 3) дім авантюриста, який розкаже історію містечка та дасть поради щодо підземелля; 4) ратуша, де можна взяти ще одне завдання та дізнатись про ситуацію; 5) рудна біржа, де можна взяти останнє завдання на демо-версію, яке виконається тільки під кінець зачистки підземелля.

Завершення таблиці 3.1

1	2
Таверна	1) (при першому заході) кат-сцена бійки між одним з напарників та бандитами; 2) перший бій проти чотирьох бандитів, з невеликим вибором посеред бою (захватити одного з них в полон чи ні).
Внутрішній двір маєтку	1) сцена пошуку таємного проходу та планування проникнення, з підказками якщо пощадили бандита раніше.
Підземелля маєтку	1) сцена зустрічі з монстром, захищаючим таємний прохід; 2) сцена обвалення одного з мостів, розташованих над розломом у землі; 3) сцена знаходження скарбу у розломі; 4) потенційні сцени знаходження тайних проходів, якщо гравець активує взаємодію у певних місцях мапи; 5) бої з ще трьома групами бандитів; 6) сцена знаходження полонених міщан; 7) сцена обшуку кабінету ватажка, яка має два можливі результати – гравець ігнорує скарб та настигає ворога, ініціюючи фінальну битву, або витрачає час і той встигає втекти.
Завершення демо-версії	1) сцена повернення з маєтку та подяки від міщан, після якої гравець може продати знайдений скарб та завершити раніше взяті завдання.

Структуру демо-версії ще можна виразити схемою, зображеною на рис. 3.2.

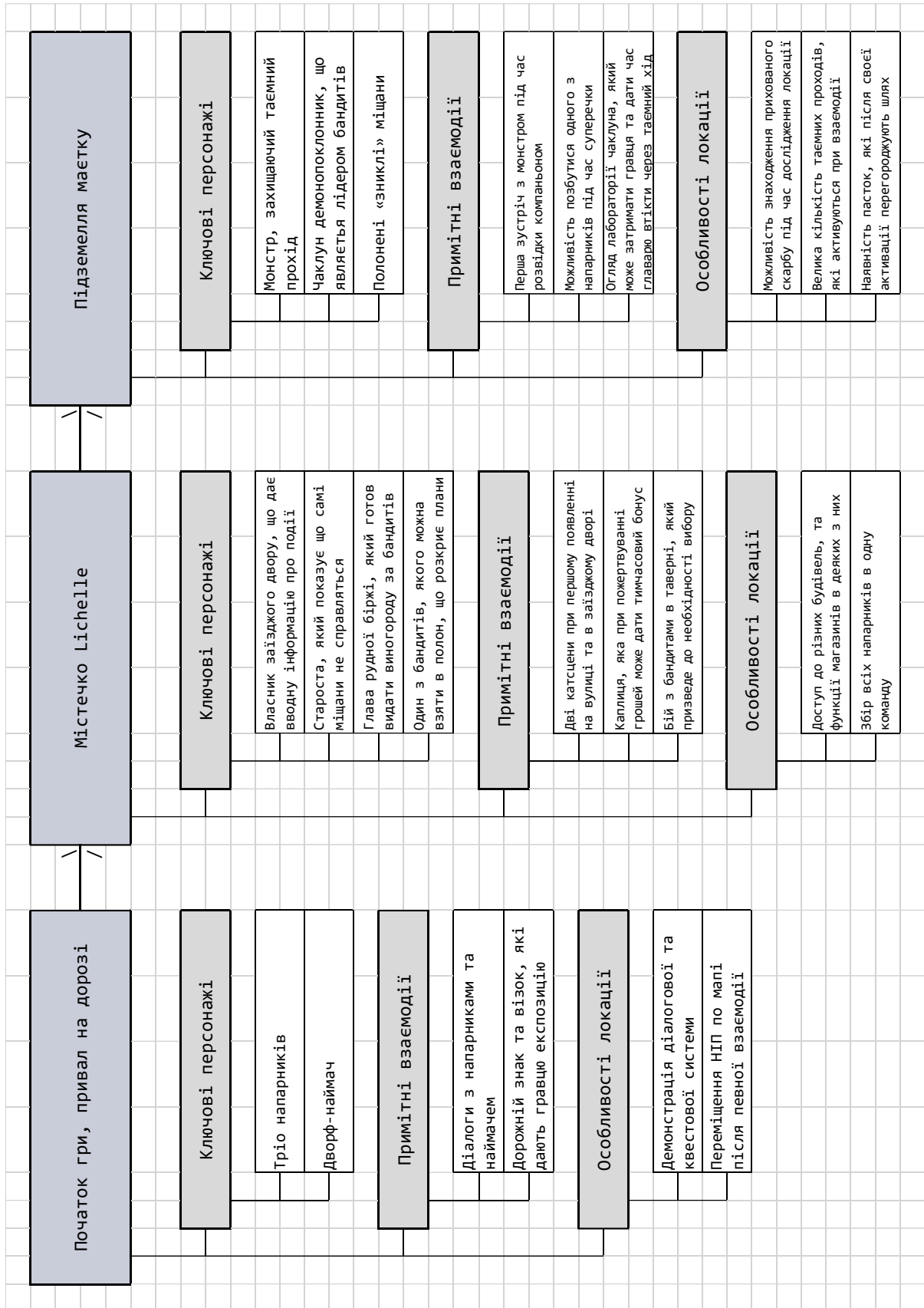


Рисунок 3.2 – Структура створеної демо-версії у вигляді схеми

Структура подій

Логіка гри організована через систему подій, які і відповідають за запуск діалогів, взаємодію з NPC, отримання предметів та зміну стану гри. Для складних сцен може використовуватись декілька сторінок подій, кожна з різними умовами активації, що дозволяє налаштувати їх відносно можливих дій гравця. Приклад багатоступеневої сторінки події можна побачити на рис. 3.3.

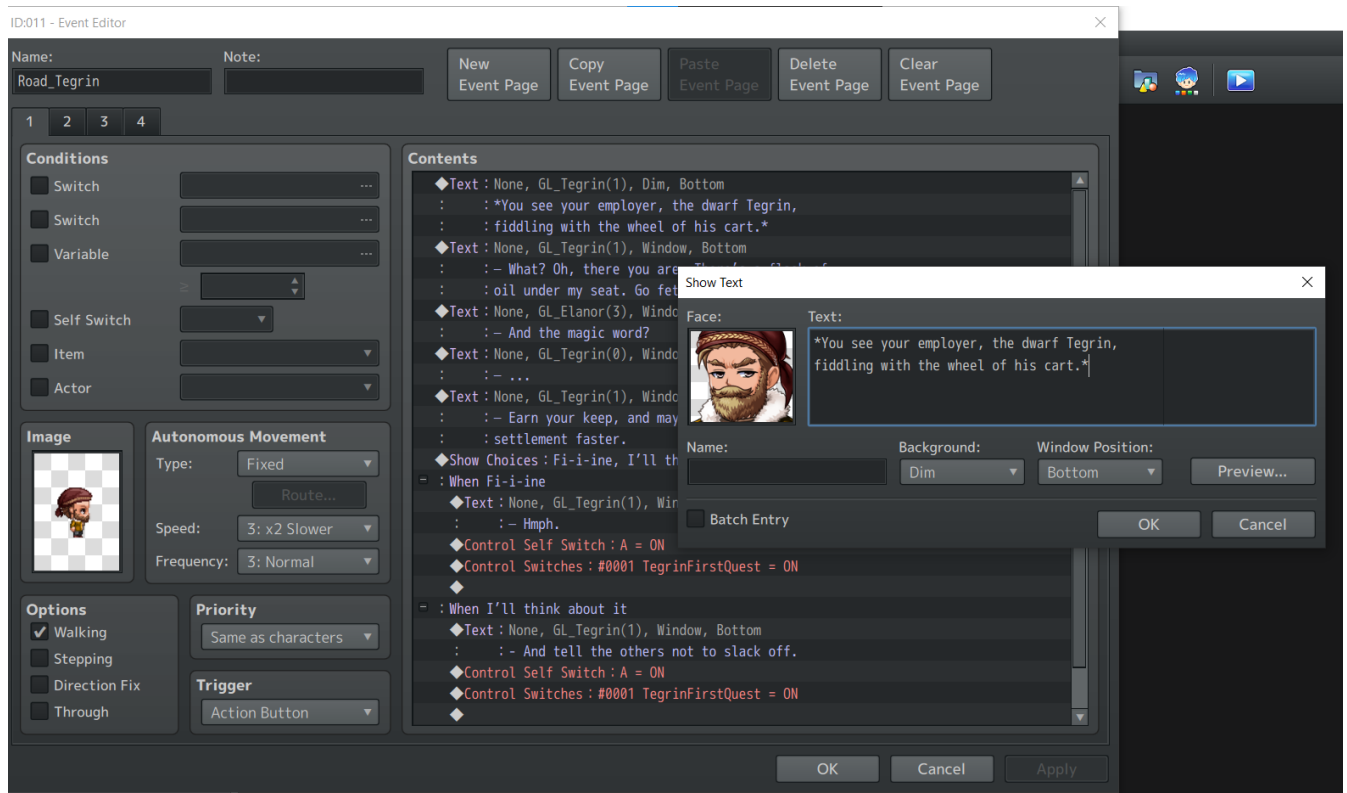


Рисунок 3.3 – Логіка взаємодії з персонажем Tegrin, перша з чотирьох сторінок

Організація ресурсів проєкту

Всі ресурси знаходяться в стандартній структурі проєкту на рушії, тобто графічні, аудіо та службові файли зберігаються у відповідних директоріях проєкту.

Варто зауважити, що власні графічні ресурси повинні бути збережені в формат PNG заради підтримки прозорості, аби при запуску гри не було зайвих «артефактів» графіки, та щоб об'єкти коректно відображались на різних шарах мапи.

Таблиця 3.1 – Основні папки проєкту RPG Maker MZ

Папка та шлях до неї	Призначення
1	2
img/characters	Спрайти персонажів та об'єктів
img/faces	Портрети для діалогів
img/tilesets	Набори тайлів для мап
img/system	Системні зображення UI та іконки
audio/bgm	Фонова музика
audio/se	Звукові ефекти
js/plugins	Плагіни на JS
data	Дані мап, предметів та подій

Сама система містить й інші папки з графікою, де розташовуються наприклад спрайти ворогів, звичайні зображення, візуальні ефекти чи взагалі окремі відео, проте в рамках даного проєкту в них будуть внесені лише мінімальні зміни.

3.2 Проєктування ігрових механік

Організація діалогової системи

Діалогова система, в основі своїй, побудована на використанні подій та текстових команд. Кожен NPC чи об'єкт, з яким взаємодії гравець, прив'язаний до окремої події, яка вже має свої активатори.

Враховуючи рамки проєкту та обраний жанр, було зроблено спробу оживити діалоги через зміну виразів обличчя основних персонажів щодо їх емоційного стану. Періодично гравцеві дається вибір реплік його персонажа, який не завжди має наслідки, але допомагає йому краще перейнятися своєю роллю.

Портрети основних персонажів, створених з використанням вбудованого генератора рушії, можна побачити на рис. 3.4.



Рисунок 3.4 – Портрети основних персонажів

Існує також можливість завантаження власних зображень на місце портретів, знайдених в мережі чи зроблених власноруч, просто додавши їх у відповідну папку, попередньо змінивши під потрібний розмір в 144x144 пікселя.

Реалізація вибору гравця

В межах проєкту використано базову систему вибору, яка дозволяє змінювати подальший перебіг окремо взятих подій. Логіка цих змін базується на перемикачах, змінних та умовах конкретних сторінок подій. Змінні також були використані для відслідковування ефектів, зав'язаних на реальному часі, та реалізації певних механік, що використовують Загальні події.

Організація квестових елементів

Враховуючи лінійність сюжету, більшість сюжетних тригерів прив'язані до конкретних взаємодій, очікуваних від персонажа, до яких його буде підштовхувати оточення чи певні NPC. Окремі квести будуть виконані в рамках однієї, максимум декількох локацій, проте в повноцінних проєктах такі обмеження можуть спокійно ігноруватись.

Також до бази даних був доданий обмежений набір предметів, які можуть бути використані під час гри для вирішення окремих ситуацій, чи допомогти під час бою. Їх перелік можна побачити на рис. 3.5.

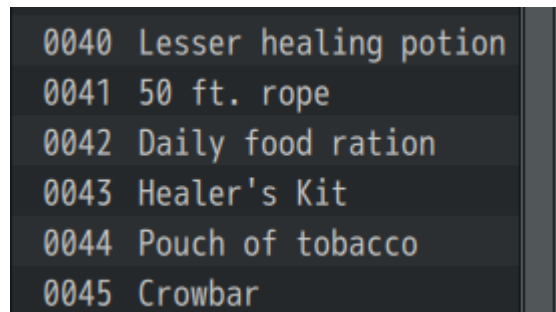


Рисунок 3.5 – Перелік предметів, доданих до загального списку

Для зручності їх було умовно розділено на використовувані предмети (Consumables), які можуть бути вільно використані гравцем, та звичайні «технічні», наявність та витрата яких буде здійснюватися у перевірках подій.

Організація бойової системи

Основними відміннями від стандартної системи є зниження значень параметрів з сотень до десятків (наприклад, з 400 одиниць здоров'я до лише 40), а також зміна ролі зброї з пасивного підвищення атаки на видачу окремої навички, прив'язаної до пошкоджень зброї.

Тобто якщо раніше всі персонажі мали однакову базову атаку, яка в системі позначалась приблизно як [Атака_персонажа * 4] - [Захист_цілі * 2], то зараз кожен вид зброї поділяється на декілька окремих типів за розкидом пошкоджень. Розберемо на прикладі формули короткого меча:

$$\text{Math.max}(1, 3 + \text{Math.ceil}(\text{Math.random()} * 6) + ((a.\text{atk} - 10)/2) - \text{Math.floor}(b.\text{def}/3)), \quad (3.1)$$

де $\text{Math.max}(1, \dots)$ – гарантує як мінімум 1 одиницю пошкоджень;

3 + – додаткові 3 фіксованих пошкодження;

$\text{Math.ceil}(\text{Math.random()} * 6)$ – видає випадкове число від 1 до 6, імітуючи 1d6;

$((a.\text{atk} - 10)/2)$ – модифікатор характеристики (10 АТК = +0, 12 = +1; 14 = +2).

При однакових параметрах 18 ATK vs DEF 12, отримаємо:

- $(18*4) - (12*2) = 72 - 24 = 48$ damage (по базовій формулі);
- $3 + 1d6 + 4 - 4 =$ від 4 до 9 damage (по модифікованій формулі).

Це полегшує балансування цифр та додає більшої цінності в отриманому спорядженні, додаючи також елемент випадковості до кожного бою.

За такою самою логікою було створено власні класи, з вже зміненими параметрами, як видно на рис. 3.6.

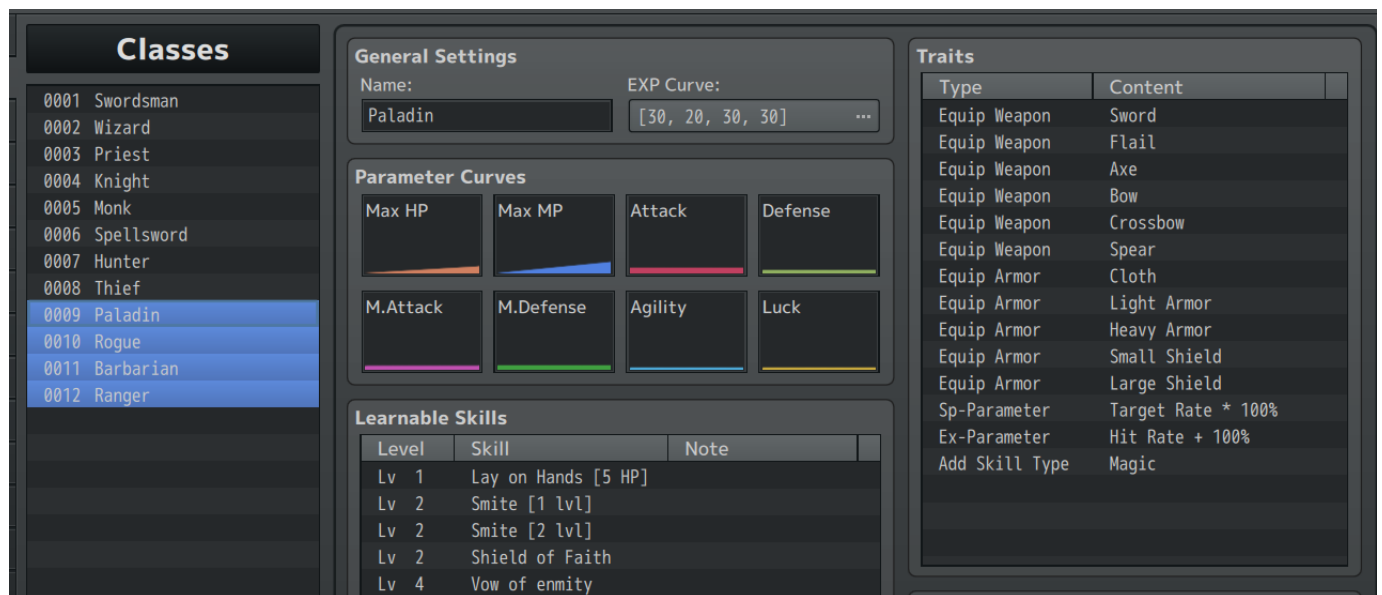


Рисунок 3.6 – Створені класи та їхні навички

В рамках демонстрації їх опрацьовано лише до п'ятого рівня, за двадцяти запланованих в рамках власної системи. Хоча сам двигун дозволяє розпланувати прогресію до 99 рівня, або й вище, якщо використовувати плагіни.

Окрім звичайних вмінь класу, які будуть використовувати обмежені MP або генеровані під час бою TP, було створено унікальні вміння з обмеженою кількістю використань, що перезаряджаються лише за умовного відпочинку. Це було досягнуто за допомогою виклику Загальної події (Common Event), окремої Змінної, в якій зберігається значення «зарядів використання» та перенесенню логіки дії в саму

загальну подію. Приклад навички-заклинання та реалізації логіки вміння з зарядами, що витрачаються, можна побачити на рис. 3.7 та 3.8.

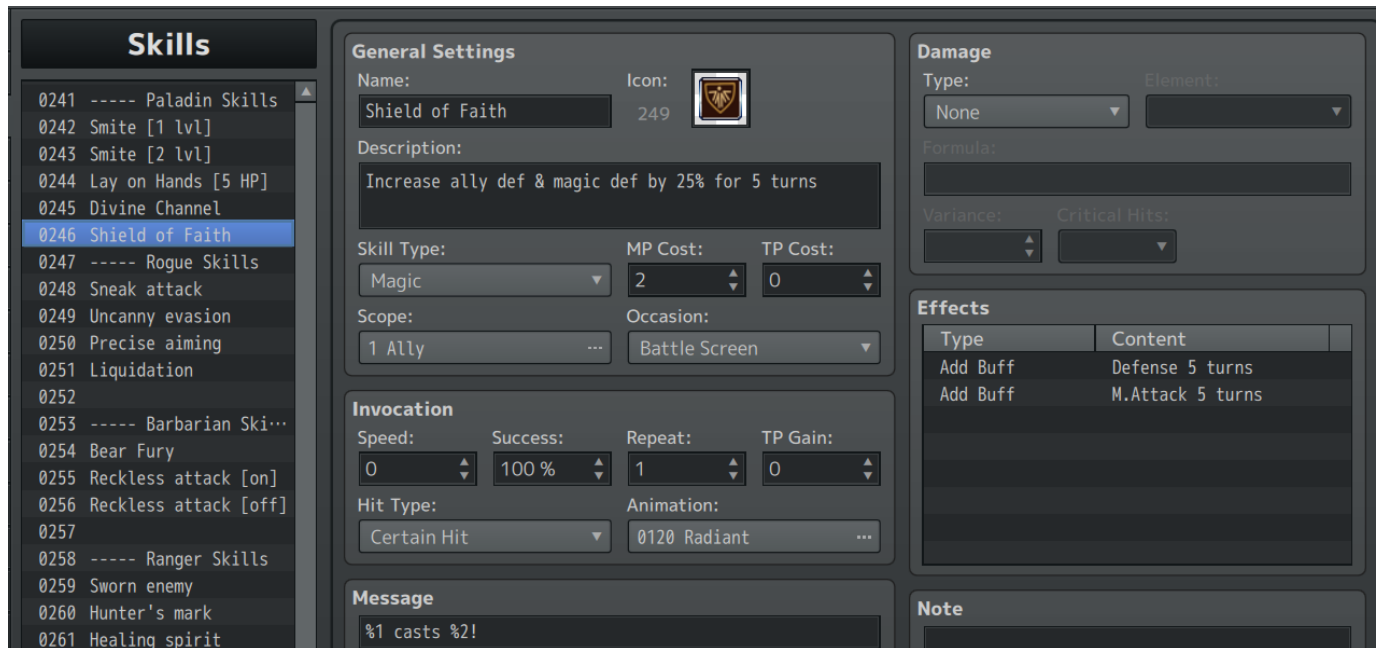


Рисунок 3.7 – Створені для демонстрації навички класів

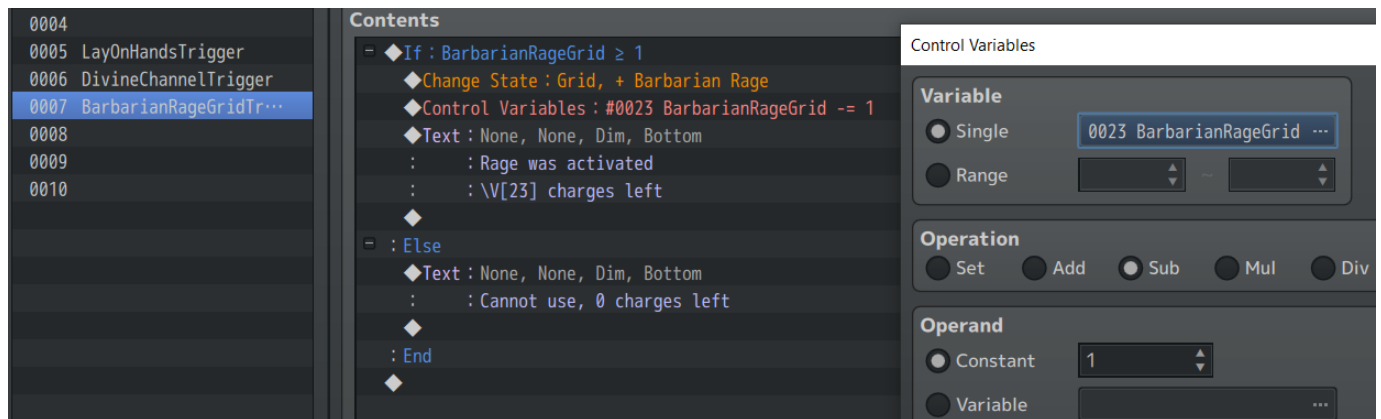


Рисунок 3.8 – Логіка вміння з зарядами, прописана в Common Events

Для виклику потрібної Загальної події треба в Effects навички обрати меню Others, в ній Common Event та виставити потрібну подію. Щоб «заряди» потрібних умінь додалися персонажу, треба виставити значення відповідних Variables в окремій Події. Для їх подальшого відновлення знадобиться окрема Подія, наприклад відпочинок, в якій окрім Recover All будуть заново виставлятися потрібні значення.

Створення та налаштування плагінів

Спільнота розробників на MV і MZ неймовірно велика, тому якщо розробнику знадобиться якась функція, якої йому не вистачає в рушії, то в більшості випадків її вже хтось реалізував і ймовірно виклав у мережу. Але буває і так, що необхідна функція надто специфічна чи потрібна у конкретному виконанні, тому на допомогу приходить можливість написання та використання власних плагінів на JavaScript.

Оскільки тріал-версія не містить плагінів, які можна отримати при покупці повноцінної гри та її доповнень, виникла необхідність додати кілька ключових плагінів. Серед них можливість використання WASD для переміщення по мапі, оскільки за замовчуванням рух персонажа прив'язаний до «стрілочок», і не може бути змінений в рамках рушія або у грі.

Знайшовши кілька інструкцій на відкритих форумах та в документації рушія, отримуємо приблизно такий вміст файлу .js:

```
(function () {  
    var replacedKeyMapper = {  
        87: 'up',  
        65: 'left',  
        83: 'down',  
        68: 'right',  
        69: 'pagedown',  
    };  
    for (i in replacedKeyMapper) {  
        Input.keyMapper[i] = replacedKeyMapper[i];  
    }  
})();
```

Зберігаємо його по шляху Ваш_проект\js\plugins, запускаємо сам проєкт та переходимо в Plugin Manager, через меню Tools або натиснувши F10. У вікні, що відкрилося, натискаємо на будь-який рядок та обираємо Edit. Потім шукаємо кнопку під Name, натискаємо вже її та шукаємо у списку назву файлу js, який створили вище.

Вигляд загального шляху по додаванню плагінів можна побачити на рис. 3.9.

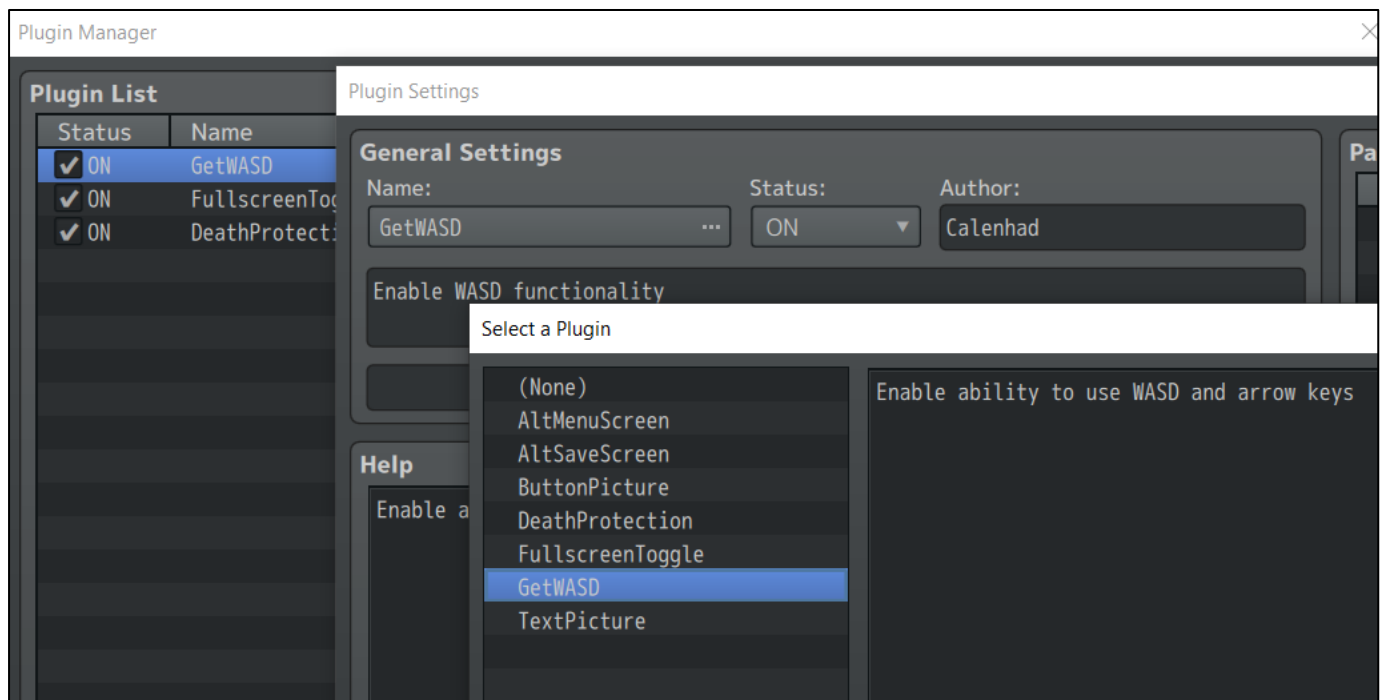


Рисунок 3.9 – Підключення плагінів до проєкту

Використовуючи плагіни, створені іншими розробниками, треба приділяти щільну увагу до їхніх Умов використання (Terms of Use), якщо вони ними зазначені.

3.3 Демонстрація роботи системи

Стартова локація

Гра стартує на невеличкій мапі, куди гравця закидають без будь-яких пояснень. На ній знаходиться певна кількість анімованих персонажів та об'єктів, з якими він може взаємодіяти та отримати експозицію. А також провести діалоги, познайомившись з основною партією персонажів, які по ходу гри приєднуються до нього як члени команди.

Паралельно цьому за кулісами, які не бачить гравець, спрацьовують декілька подій – запуск BGM, активація фонових перемикачів для Загальних подій та видача

йому стартових предметів, таких як валюта, зілля та похідний інвентар. Зовнішній вигляд локації можна побачити на рис. 3.10.

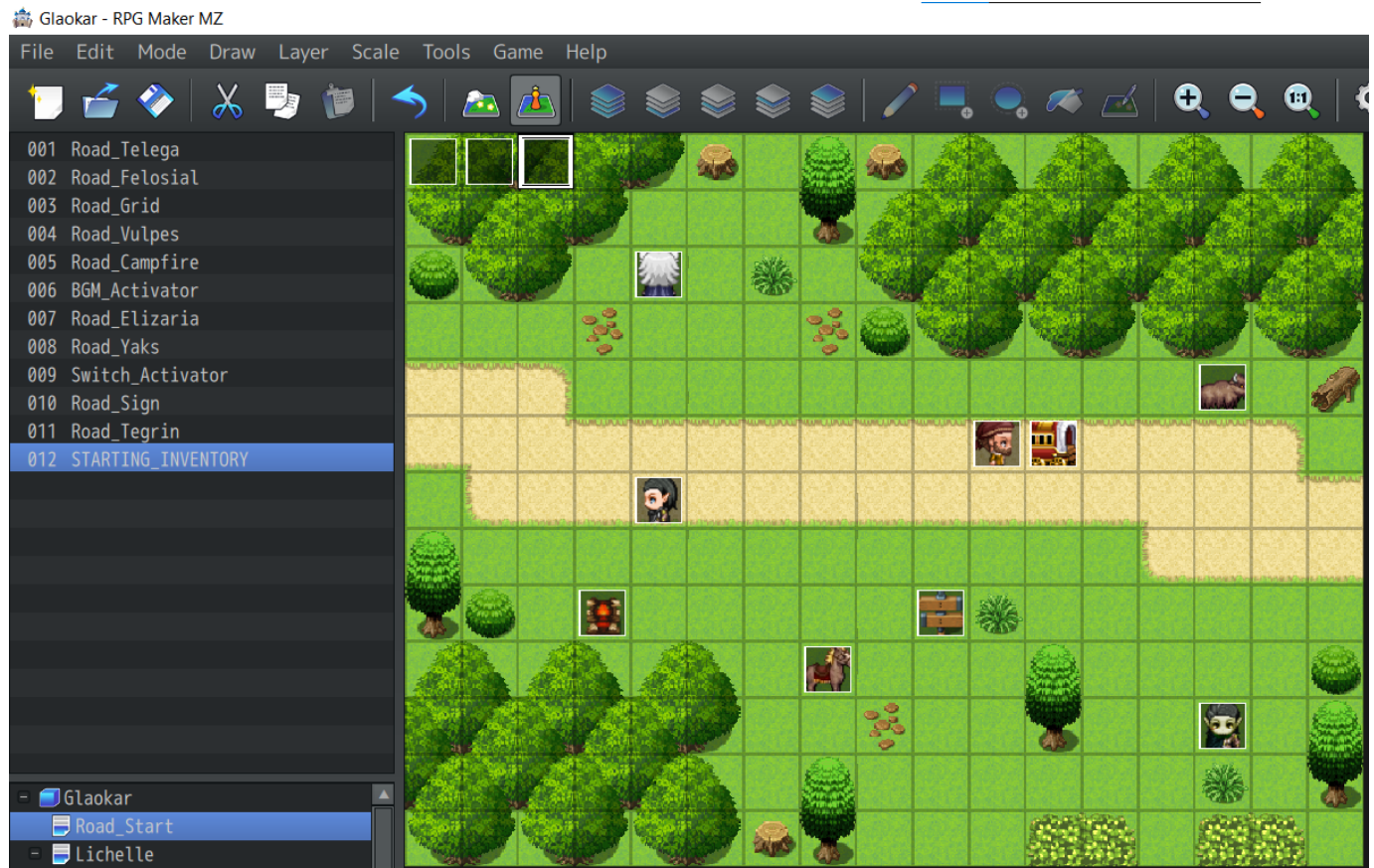


Рисунок 3.10 – Стартова локація та її events, як їх бачить розробник

Рано чи пізно гравець активує діалог з NPC під іменем Tegrin, що на цій карті позначений під подією з номером 011. Після короткої розмови гравцю буде видане завдання провзаємодіяти вже з візком, тобто подією під номером 001.

Отримавши сюжетний предмет, який просив Tegrin, та передавши той йому, гравцю видають ще одне завдання – повідомити решту компанії про завершення відпочинку, через ініціацію діалогу з іншими персонажами, якщо той ще цього не робив.

Провівши розмову з кожним із них та при бажанні послухавши сцену біля візки, гравець звітує Tegrin та отримує направлення до свого коня, тобто події під номером 007, після чого він переноситься на наступну локацію.

Містечко, основна локація демо-версії

При переході гравця зустрічає невелика кат-сцена, в якій дається короткий опис містечка, а також натяк на головну точку інтересу. Після неї екран проявляється, та управління переходить назад до гравця. Частина персонажів заходить у будівлю поряд, спонукаючи прослідкувати за ними.

Хоча є можливість вільно переміщуватись по всій локації, в даний момент майже всі двері закриті, тому рано чи пізно герой все одно зайде в потрібну будівлю. Початкову сцену після переходу на нову локацію можна побачити на рис. 3.11.



Рисунок 3.11 – Початкова сцена другої локації

Імітація переміщення NPC зроблена через прості команди руху, з включенням режиму Through перед дотиканням дверей, та видаленням події вже після цього.

При першому заході відбувається ще одна кат-сцена, яка доповнює характер одного з майбутніх членів команди. А також показує, що кожен з них так само займається своїми ділами і теж взаємодіє з оточенням, а не тільки відповідає персонажу гравця. Перший поверх заїжджого двору та початок наступної кат-сцени можна побачити на рис. 3.12.



Рисунок 3.12 – Кат-сцена при першому вході в таверну

Для зручної роботи з декількома подіями, які повинні програватись одна за одною, або зникати після певної взаємодії, було використано команду скрипту, що на третьому листі. Наприклад, `$gameSelfSwitches.setValue([2, 15, 'B'], true)`, де 2 = номер мапи, 15 = номер події, B = символ власного перемикача, а true = включити попередньо зазначений локальний перемикач (щоб виключити, треба false).

Як і минула кат-сцена, ця також зроблена за допомогою комбінації Автозапускаємих (Autorun) та Паралельних (Parallel) подій, які після свого виконання перемикаються на сторінку з Видаленням події (Erase event), аби вони не запускались при повторному відвідуванні.

Тут гравець вперше отримує доступ до повноцінної діалогової системи та магазину, де може витратити отримані раніше гроші та дізнатись про недавні події. Оскільки між точкою взаємодії з подією є одна непроходима клітинка, був використаний скрипт `$gameMap.event(6).start();`, де 6 це номер події, яку потрібно активувати, щоб почати діалог. Діалогове вікно можна побачити на рис. 3.13.



Рисунок 3.13 – Діалогове вікно з можливістю перейти до магазину

Саме вікно зроблено через декілька циклів з наборами виборів всередині – перший цикл для діалогу в цілому, аби той не переривався після обирання одного з варіантів, другий для пункту Слухів, де є розділення на кілька окремих тем розмови.

Після того як гравець активує подію умовного відпочинку на другому поверсі локації, відновивши можливо витрачені використання навичок, та спуститься вниз, його зустрине один з персонажів-напарників. Запропонувавши свою компанію, він вступить до команди і візуально відобразатиметься на карті, супроводжуючи гравця, як видно на рис. 3.14. Разом з напарником також додається певний набір предметів, імітуючи його інвентар.



Рисунок 3.14 – Поява першого напарника в команді

Система супроводу напарників

Налаштування напарників відбувається в БД, на сторінці Персонажів та Системи. На сторінці Персонажів розробник вставляє потрібні зображення на місце аватару персонажа, його спрайта на мапі та під час бою, якщо включений відповідний режим відображення, а також його початкові параметри та спорядження. В Системі можна лише вимкнути їх відображення на мапі як «паровозик» під час переміщення, що слідкує за головним персонажем.

Сам гравець має можливість змінити формацію, розставивши персонажів в потрібному йому порядку. За замовчуванням це впливає лише на відображення порядку партії на мапі та в окремих меню, проте за бажанням можна вказати особливі події та взаємодії, прив'язавши реакції на них до змінної, яка буде відслідковувати ID першого члену групи.

Для створення деяких сцен може знадобитись зміна головного героя на когось іншого, що може бути реалізовано через просте вилучення з команди всіх напарників окрім потрібного персонажа.

За допомогою подібного переключення можна робити окремі сцени з участю різних напарників, а якщо залишити один з бланків персонажа пустим, можна навіть робити сцени з NPC, взагалі без участі персонажів, підконтрольних гравцю.

Відкриття всієї локації

Після отримання першого напарника, перед гравцем нарешті відкривається можливість дослідити всю локацію, нехай і обмежену лише територією містечка та будівель у ньому. Він може відвідати кілька магазинів, таких як кузня або торговий пост, аби купити на отримані гроші необхідні розхідники або нове спорядження.

Може також заглянути до старости або в рудну біржу, щоб взяти додаткові завдання, одне з яких скоріше крючок на потенційне продовження гри, а інше дасть додаткову нагороду в кінці зачистки підземелля, яке буде відвідано далі по сюжету. Частину локацій, які гравець може відвідати, можна побачити на рис. 3.15



Рисунок 3.15 – Деякі локації демо-версії гри

Гравець також може одразу направитись до другої таверни, де застане сцену суперечки між місцевими розбійниками та іншим компаньйоном, і після короткого обміну погрозами вступаючи в першу битву.

Базовий рушій містить помірну кількість готових спрайтів для ворогів, проте вони зазвичай краще виглядають для фронтального варіанту бою, або одиночних сутичок. Оскільки в більшості випадків потрібен специфічний вигляд недругів, більшість розробників використовують власні зображення.

В даному випадку був використаний функціонал вбудованого генератора персонажів, спрайт з якого було вирізано та підігнано під необхідний розмір, аби співпадати з командою гравця. Мінусом такого підходу є відсутність у ворогів анімацій, притаманних спрайтам у керованих персонажів, що може бути вирішено тільки через використання окремих плагінів.

Для відображення персонажів та їхніх анімацій використовуються зображення із папки `\img\sv_actors`, створені в генераторі раніше. Зовнішній вигляд вікна бою можна побачити на рис. 3.16.



Рисунок 3.16 – Вікно бою проти чотирьох бандитів

Параметри ворогів та логіку окремого бою налаштовують у відповідних сторінках БД рушія. На прикладі першого бою, було додано подію, що коли одному з розбійників опустять здоров'я нижче 50%, він спробує здатися, що дасть гравцеві

вибір – пощадити чи проігнорувати. В залежності від вибору зміниться діалог після бою та декілька наступних сцен, що вносить RPG-елементи, та може бути стимулом пройти гру ще раз, обравши інші варіанти розвитку сюжету.

Зовнішній вигляд сторінки ворогу (Enemies) та окремої групи (Troops) можна побачити на рис. 3.17.

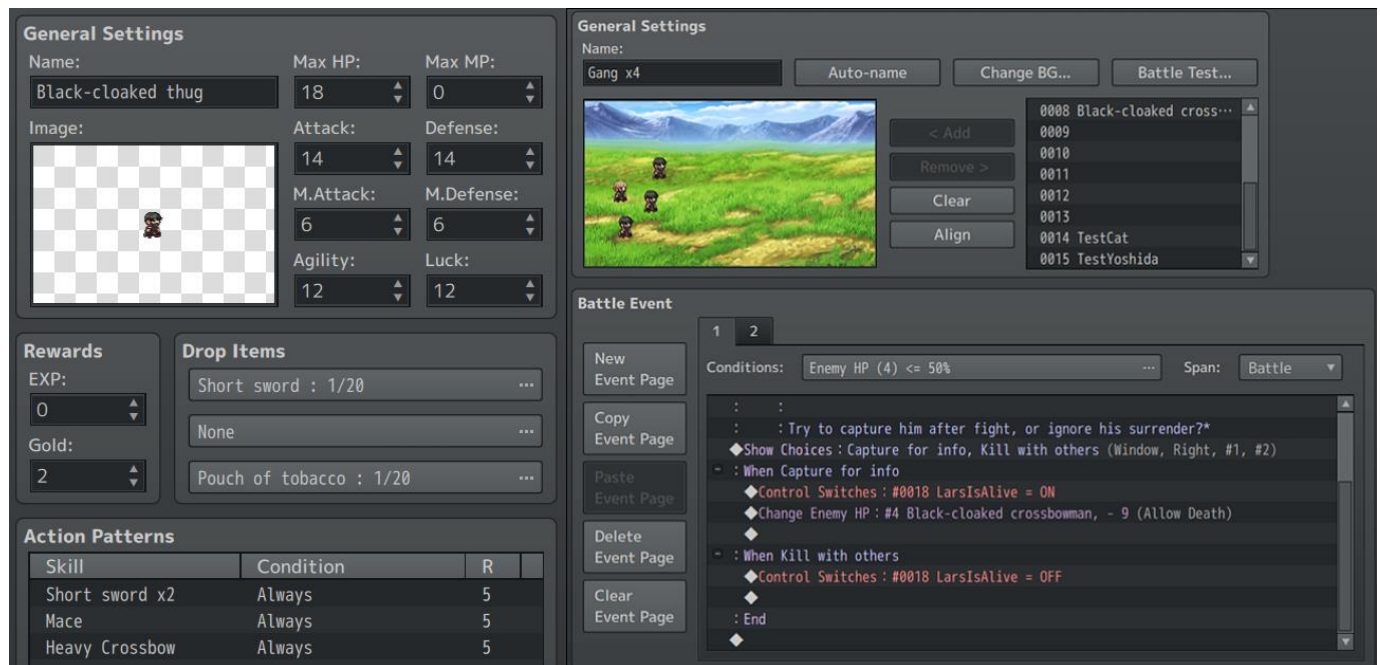


Рисунок 3.17 – Вікна налаштування ворогів та окремих груп

Після завершення бою, та якщо він ще не взяв завдання на знищення лігва бандитів, у гравця буде останній шанс вирішитись що робити в даному сюжетному моменті – допомогти міщанам та спуститись в підземелля, чи проігнорувати конфлікт та спонуки деяких напарників, і відправитись з наймачем далі, передчасно завершивши сюжет демо-версії. Можна сказати тут з'являється перша можливість обрати кінцівку, або зробити важливий сюжетний вибір для майбутньої кампанії, у випадку якщо сюжет був би продовжений за межі демо-версії.

Взявши за основу варіант, що гравець все ж таки піде розбиратись з загрозою, що зависла над містечком, NPC або потенційно пійманий в бою бандит направляє їх на локацію з руїнами маєтку, шукати або відразу підказавши де шукати таємний прохід, як видно на рис. 3.18.



Рисунок 3.18 – Знаходження таємного проходу в підземелля зруйнованого маєтку

Після його знаходження відбувається примусове збереження прогресу, та перехід на сцену підготовки до проникнення, де у гравця є можливість відправити на розвідку одного з напарників, тимчасово змінивши головного героя на іншого.

Початок підземелля

Після переходу в локацію підземелля, гравець в перший раз зустрічає монстра, який може вільно рухатись по мапі. Враховуючи попередньо обговорені плани, гравець може спробувати відразу втекти назад, або спробувати побороти його одному. Сцена зустрічі з переслідуючим монстром можна побачити на рис. 3.19.



Рисунок 3.19 – Перша засада на гравця

Якщо під час бою шанси не будуть на стороні персонажа, в налаштуванні битви була проставлена галочка на дозволену спробу вбїгти, що одразу перенесе персонажа на попередню локацію. Там він зможе розказати про розвідане, спробувати згадати чи дізнатись слабкі сторони монстра, та вже об'єднатись з всією партією.

Перевірка характеристик

Примітним моментом є реалізація однієї з атак монстра, виконаної як перевірка характеристик персонажа, вона ж «saving throw». За принципом як і в формулі зі зброєю, береться його параметр mdef, від нього віднімається 10 та отримане число ділиться на два, округляючи вниз – це дає значення «модифікатору» до кидка віртуального кубика з 20 гранями.

Отриманий модифікатор, наприклад $((16 - 10) / 2) = 3$, додається до випадкового значення від 1 до 20, і вже отриманий результат порівнюється зі складністю самої перевірки, в даному випадку 12 або більше. Результат перевірки може різнитись від

використаної навички чи ситуації – в даному варіанті, якщо гравець успішно проходить перевірку, то можливі пошкодження від атаки діляться на два.

По такому самому принципу можна створити перевірки інших параметрів чи просто випадковий кидок кубика. Через Agility або Luck можна перевіряти на уникнення чи знаходження пасток, Attack можна використовувати як визначення фізичної сили персонажа, Magic attack як його ерудицію та ін. Проте для більш тонкого налаштування може бути легше використати відповідні плагіни.

В рамках підземелля було використано лише декілька перевірок, таких як Luck поряд зі схованим скарбом та Agility, щоб запобігти пошкодженню від падіння у пастки з руйнівною підлогою. Деякі з них було зроблено через подію з активацією через дотик до персонажа, інші через повноцінні скрипти та використання R-тайлів в якості окремих зон, та відображенням результатів кидків віртуального кубика.

Інтерактивність оточення

Серед особливих механік, що були реалізовані в рамках підземелля, можна зазначити «пастки з підлогою, що провалюється» та таємні проходи, як видно на рис. 3.20 та 3.21.

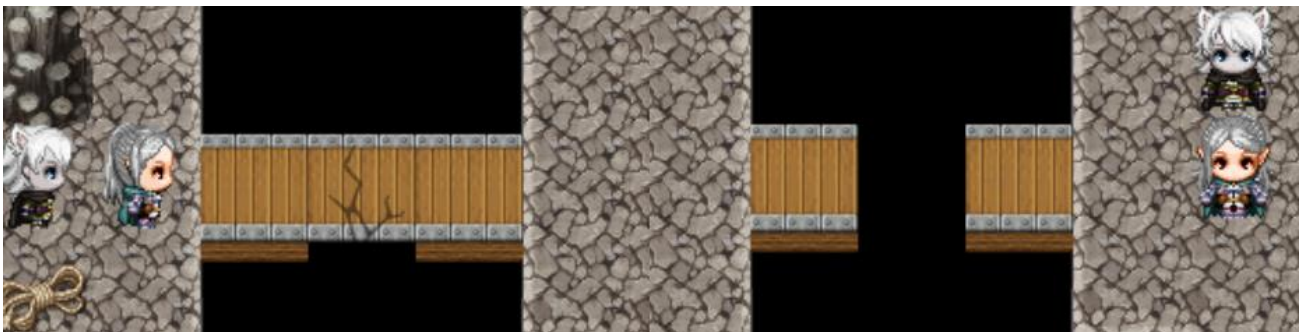


Рисунок 3.20 – Приклад реалізації тайлу, блокуючого переміщення

Для створення подібних «заблокованих місць» на мапі було використано події з двома сторінками, на другій з яких в якості зображення використовується елемент тайлсету, помічений в БД як непроходимий. В даних прикладах це імпортовані тайлсети E та B, де взято тайли Darkness та Entrance.

За схожим принципом реалізовано й створення таємних проходів, де на першій сторінці події стоїть зображення тайлу стіни. При взаємодії над гравцем з'являється хмарка з символом знаку оклику, після чого власний перемикач переводить на другу сторінку, і зображення змінюється на прохід, який в БД помічений як маючий прохідність.



Рисунок 3.21 – Приклад реалізації тайлу, що додає можливість переміщення

Для імітації «туману війни» в таємних проходах було використано скрипт, що перевіряє наявність персонажа гравця на певній зоні, помічений окремим тайлом з набору R, наприклад `$gameMap.regionId($gamePlayer.x, $gamePlayer.y) === 1`.

Для вирішення проблеми перетину тайлів одного з одним, коли стоячи в «проході» персонаж може переміститись на «стіни», можна створити пусті події з пріоритетом «на одному рівні», що будуть блокувати шлях, як видно на рис. 3.22.

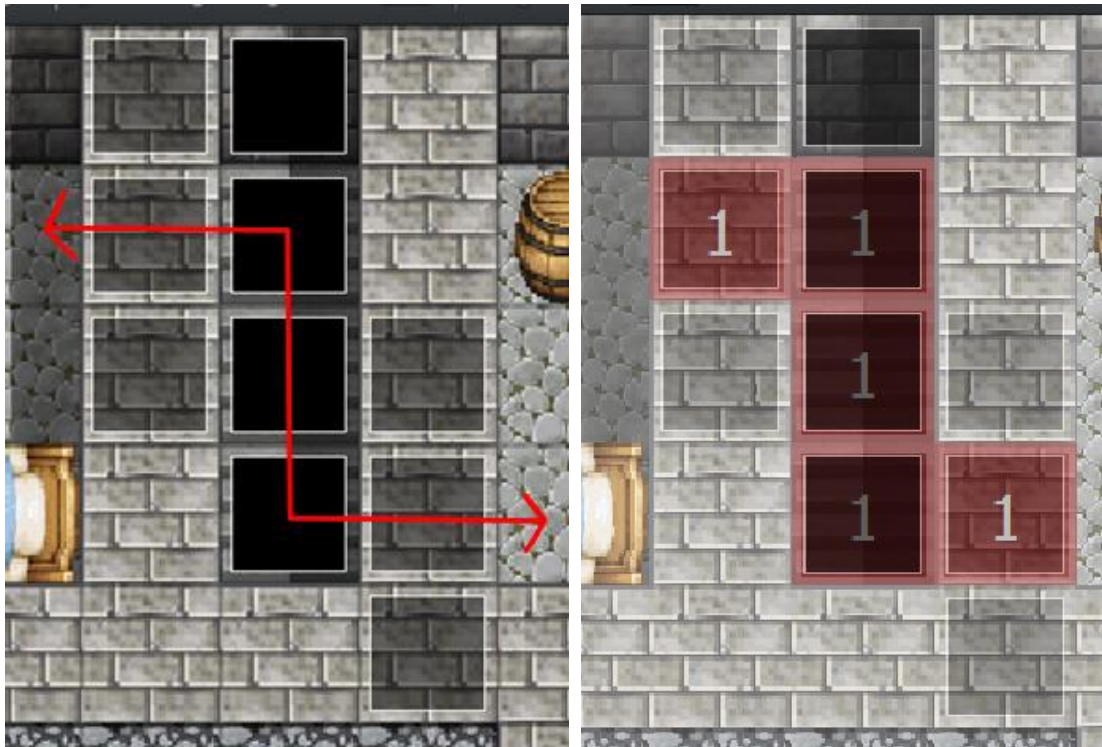


Рисунок 3.22 – Блокування шляху через невидимі події та використання зон

Метод можливо не самий оптимальний, проте дуже легкий в плані його втілення, без використання особливих тайлів чи важкої логіки всередині.

Дослідження підземелля, або етап «dungeon crawler»

Розібравшись із загрозою на вході, гравцеві відкривається дослідження всього підземелля. Його організацію як окремого елемента історії та гри можна побачити в табл. 3.2.

Таблиця 3.2 – Організація підземелля

Елемент	Вміст
Структура	1) центральний маршрут; 2) необов'язкові кімнати; 3) секрети.
Наповнення	1) чергування боїв; 2) діалоги та реакції напарників; 3) пастки та прихований скарб.

Як видно з таблиці вище, гравець може оглядати контейнери заради пошуку предметів, взаємодіяти з певним оточенням, знаходити таємні проходи і вступати в сутички з бандитами, що залишилися в підземеллі. Загальний вигляд мапи можна побачити на рис. 3.23.



Рисунок 3.23 – Мапа підземелля (без розставлених ворогів та взаємодій)

Воно вважається завершеним, коли гравець відвідає дві кімнати зліва-зверху, а також переможе всіх супротивників, що знаходяться на рівні.

3.4 Аналіз отриманих результатів

В рамках демо-версії вдалось реалізувати декілька крупних локацій та помірну кількість менших, між якими гравець може переміщуватись. Також в певній мірі переробити принцип впливу характеристик зброї, та самі формули бою.

Також було створено власні класи персонажів, з їхніми унікальними навичками та значеннями параметрів. Реалізовано систему лімітованих використань окремих навичок, які «перезаряджаються» після певних подій, таких як відпочинок.

За допомогою вбудованого в рушій генератора було перенесено чи створено зображення ключових персонажів історії, що використовуються під час діалогів та бою, в якості анімованих спрайтів. Схожим чином опрацьовано принцип додавання власних спрайтів для подій та ворогів, в рамках використання графічних редакторів для малювання та підгонки зображень під потрібний розмір.

Використовуючи вбудовану систему подій, вдалось майже повністю перенести та реалізувати всі заплановані діалоги та взаємодії – як між напарниками та NPC, так і між окремими елементами оточення, як предмети чи особливі місця на мапі.

Проблеми та обмеження

Серед основних проблем, що виникли під час роботи, можна зазначити складність налаштування подій, які повинні виконуватись одна після одної, в особливості створення окремих сцен з об'єктами і персонажами, що пересуваються.

Це може стати особливо незручно, коли якась локація має бути певним «хабом» чи базою, де гравець часто перебуватиме чи буде з'являтися, при бажанні наповнити її окремими сценами чи подіями, зав'язаними на певних тригерах.

Іншим обмеженням базового рушія можна назвати тотальну фронтальну направленість всіх графічних елементів в сторону до гравця, що особливо помітно під час наповнення інтер'єру – ліжка, стули, полички, двері та інші елементи декору не мають бокового відображення. У розробника є можливість їх перевернути власноруч та повторно імпортувати до потрібного набору тайлів, проте в багатьох випадках стилістично зображення буде дуже сильно вибиватись серед оточення.

Проте слід зазначити, що майже всі проблеми можна вирішити через використання плагінів чи власних наборів графіки, знайдених в мережі чи створених власноруч.

Тріал версія

Функціонально пробна версія ідентична до повної ліцензії – різниця лише в обмеженні по часу та правах на публікацію гри.

В рамках пробної версії проєкти можна створювати лише для себе – ліцензійна угода (EULA) забороняє розповсюджувати (як платно, **так і безкоштовно**) та продавати ігри, створені у пробній версії. Проте весь прогрес (мапи, БД, зображення та ін.) зберігається, і при бажанні можна продовжити розробку, купивши повну версію.

Перспективи розширення

Під час створення демо-версії було отримано натхнення на багато різних елементів настільно-рольових ігор, які можна спробувати перенести в рамках системи рушія. Можливість використання в формулах власних скриптів дуже сильно поширює можливості створення та застосування окремих механік, таких як параметри чи взагалі окремі загальні події та змінні.

Простота редактору мап та подій дозволяє при необхідній кількості часу та зусиль розширити масштаби гри настільки, наскільки буде потрібно для історії. Вбудовані можливості дозволяють робити як інтенсивні на бої ігри, так і більш сюжетно-орієнтовані, де системи бою замінені на сцені втікання, пошук предметів чи вирішення головоломок.

Висновки до розділу 3

В рамках розділу було описано структуру та окремі елементи демонстраційної версії ігрового застосунку на RPG Maker MZ. Було створено набір взаємопов'язаних локацій, подій та сценаріїв, які забезпечують лінійний та послідовний розвиток сюжету, а також взаємодію гравця з оточенням.

Також було успішно організовано діалогову систему, квестову систему (через перевірки потрібних змінних чи значень) та вплив рішень гравця на певні події. В деякій мірі перероблено бойові механіки, а саме змінено формули пошкоджень і

висоту параметрів. Для цього також створено власні класи персонажів, з унікальними навичками для кожного з них, систему навичок з обмеженою кількістю використань та механіки перевірок характеристик.

Ще було приділено увагу до організації ресурсів проєкту, загальній структурі подій та використанню плагінів, що можуть розширити стандартний функціонал рушія.

У результаті було отримано повноцінний каркас для потенційного подальшого масштабування гри, додавання нових механік, локацій та подій. При необхідності отриманий досвід можна в подальшому використовувати для створення ігор в інших жанрах та напрямках.

4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

Після завершення розробки та тестування гри, розробнику залишається лише перейти по File --> Deployment, та обрати необхідну платформу і шлях збереження, як видно на рис. 4.1.

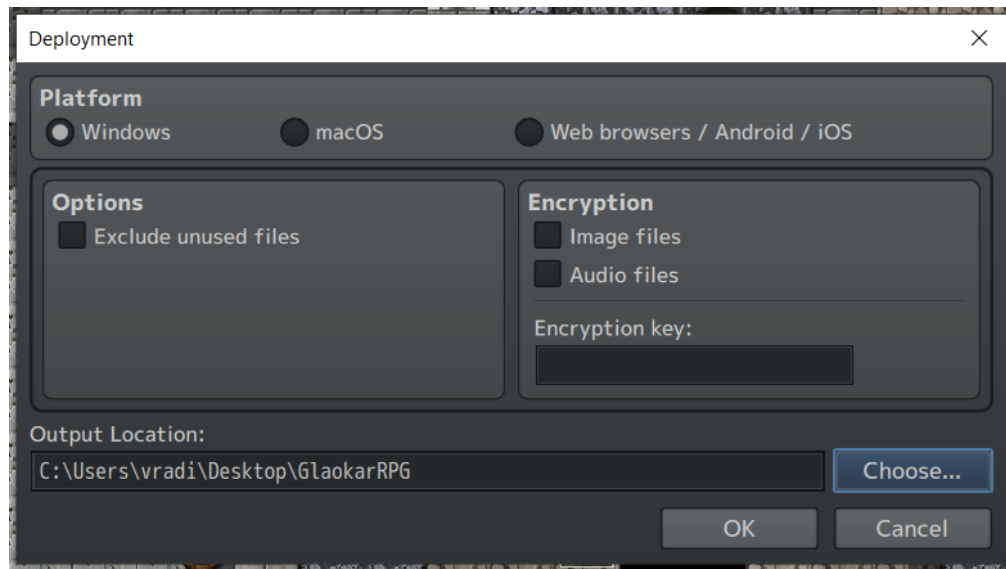


Рисунок 4.1 – Меню експорту готової гри

Після завершення процесу, у обраній директорії з'являться необхідні файли та один файл з розширенням .exe, який і буде запускати саму гру. Слід зазначити що іконка, яку розробник встановлює в img/icon відображається лише під час запуску гри, а не на самому виконуваному файлі – той бере значок компасу або дракона, прив'язаний до рушія, і для його зміни треба використовувати сторонні програми.

4.1 Керівництво користувача

Ігровий інтерфейс, як було зазначено в попередніх розділах, залишився майже без змін, зберігши базовий вигляд від рушія. Серед функцій, які увімкнені за замовчуванням, можна зазначити сенсорний режим (Touch UI), що додає можливість взаємодії з деякими елементами інтерфейсу за допомогою миші, а не тільки клавіатури.

Системні вимоги

Оскільки в базовій комплектації рушії призначений для розробки 2D-ігор, які можуть запускатись на телефоні або навіть браузері, для них зазвичай потрібні лише мінімальні вимоги. Серед яких можна перерахувати наступні:

- наявність девайсу з встановленим Windows, macOS, Android або iOS;
- клавіатура або миша (режим Touch UI);
- роздільна здатність екрану від 816x624 пікселів;
- на прикладі створеної демо-версії, вільне місце на диску від ~280 до ~360

Мбайт (якщо при експорті не виключати невикористані файли).

Управління

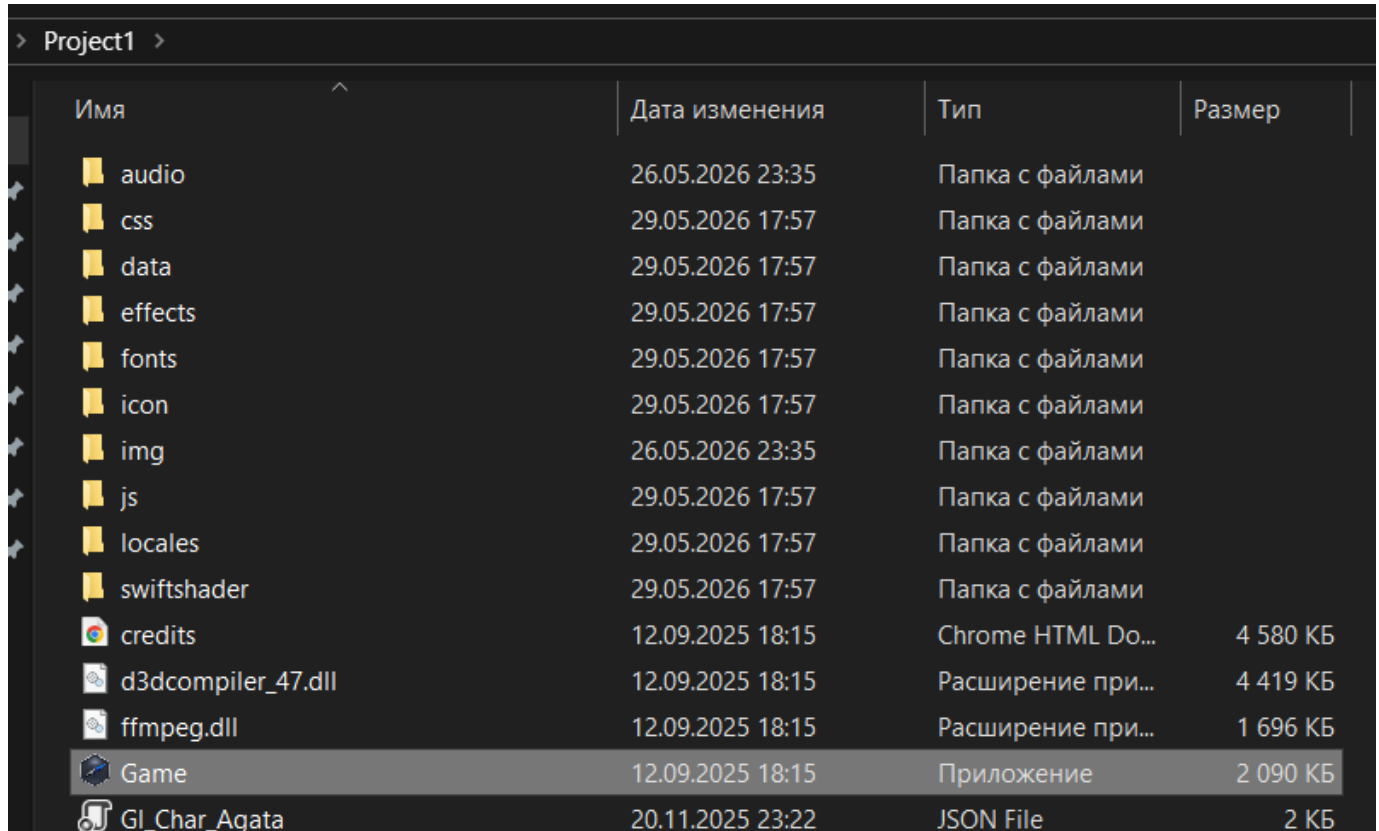
За замовчуванням, розробник не має можливості зазначити які клавіші будуть відповідати за які елементи UI, хоча це можна зробити через застосування плагінів. Відповідно сам гравець також не може переназначити управління, якщо розробник не задав такої можливості через плагіни. Проте сам рушій має доволі гнучке керування, описане в табл. 4.1, на прикладі ПК з клавіатурою та мишкою.

Таблиця 4.1. – Розкладка клавіатури

Елемент управління	Як його реалізовано
Переміщення по мапі	1) стрілочки; 2) натискання миші на необхідну клітинку; 3) WASD, додане за допомогою окремого плагіну .
Взаємодія з подіями	1) клавіші Enter, Z або Space; 2) натискання на подію лівою кнопкою миші.
Взаємодія з меню	1) відкривання/закривання меню через клавіші Esc, X або Numpad 0; 2) підтвердження або перехід далі через клавіші взаємодії.
Тільки для розробника	1) F9 для відкриття вікна з Перемикачами та Змінними; 2) рух із затиснутим ALT для проходження всіх тайлів.

Запуск гри

Після того, як гру було експортовано у потрібну директорію, її можна легко додати у архів для полегшення перенесення та розповсюдження. І вже архівовану гру можна пробувати поширювати на відповідних ресурсах, або відсилати знайомим чи колегам для тестування або збору відгуків. При відкритті чи розархівуванні папки зі грою, отримувач побачить її вміст, як видно на рис. 4.2.



Имя	Дата изменения	Тип	Размер
audio	26.05.2026 23:35	Папка с файлами	
css	29.05.2026 17:57	Папка с файлами	
data	29.05.2026 17:57	Папка с файлами	
effects	29.05.2026 17:57	Папка с файлами	
fonts	29.05.2026 17:57	Папка с файлами	
icon	29.05.2026 17:57	Папка с файлами	
img	26.05.2026 23:35	Папка с файлами	
js	29.05.2026 17:57	Папка с файлами	
locales	29.05.2026 17:57	Папка с файлами	
swiftshader	29.05.2026 17:57	Папка с файлами	
credits	12.09.2025 18:15	Chrome HTML Do...	4 580 КБ
d3dcompiler_47.dll	12.09.2025 18:15	Расширение при...	4 419 КБ
ffmpeg.dll	12.09.2025 18:15	Расширение при...	1 696 КБ
Game	12.09.2025 18:15	Приложение	2 090 КБ
Gl_Char_Agata	20.11.2025 23:22	JSON File	2 КБ

Рисунок 4.2 – Вміст експортованої папки з грою та її виконуваний файл

За замовчуванням аудіо та графічні файли експортуються з відкритим до них доступом, тобто будь хто може взяти створені ассети. Якщо розробник забажає, він перед експортом може проставити окремі пункти на шифрування цих файлів з власноруч заданим ключем шифрування (encryptionKey).

В мережі можна знайти неофіційні програми для розшифрування файлів, наприклад якщо необхідно відновити втрачений проєкт, проте їх використання залишається на розсуд користувача.

Запустивши гру, гравця зустрічає титульний екран з зображенням та музикою, яку встановив розробник в налаштуваннях БД. В базовому наборі у гравця є лише три можливі опції – почати нову гру, завантажити збережену з обраного сейв-файлу, та відкрити налаштування.

В налаштуваннях можна змінити рівень гучності окремих аудіо-елементів (BGM, BGS, ME та SE), увімкнути такі корисні функції автобіг (завжди переміщуватись з підвищеною швидкістю, а не тільки при затисканні Shift) та запам'ятовування останніх обраних команд (щоб не треба було кожен раз наводитись на останню використану опцію в меню). Сам титульний екран демо-версії можна побачити на рис. 4.3

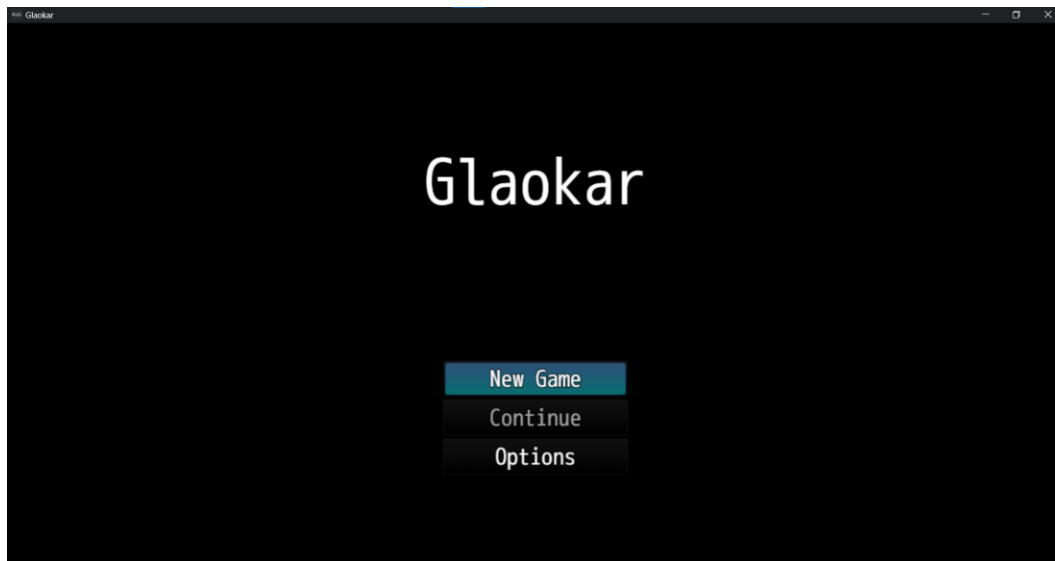


Рисунок 4.3 – Титульний екран демо-версії

Рушій має невеликий набір готових зображень та обрамлень, проте їх стилістика може не завжди відповідати використаній у грі. Плюс, використання базових асетів (без будь-яких змін) у серйозних проектах зазвичай не схвалюється, нехай офіційно і не забороняється.

В рамках цієї кімнати розробника під час роботи було опрацьовано елементи, перелічені в табл. 4.2.

Таблиця 4.2. – Використання DEV-кімнати

Елемент гри	Що саме було опрацьовано
1	2
Персонажі	<ol style="list-style-type: none">1) класи персонажів, їх можливості та обмеження;2) навички класів, балансування їх використання;3) параметри броні та формули зброї;4) використання предметів та ресурсів;5) опрацювання принципів роботи існуючих та створених власноруч станів.
Бій	<ol style="list-style-type: none">1) загальний принцип бойової системи (доступні дії, порядок ходів, використання навичок);2) перевірка працездатності формул для зброї та окремих навичок;3) балансування параметрів ворогів;4) корегування кількості сутичок на поточний етап розвитку персонажів;5) масштабування зображень ворогів для вікна бою.
Події	<ol style="list-style-type: none">1) взаємодія гравця з багатопаровими подіями, розгалуження умов в рамках однієї події;2) перевірка роботи загальних подій (тимчасові бонуси, заряди навичок та затемнення екрану, що імітує вечір);3) отримання/вилучення предметів та валюти, механіка торгівлі;4) перевірка пошуку шляху в рамках логіки рушію (ворог як подія, що рухається);5) створення сцен без участі гравця.

Завершення таблиці 4.2.

Оточення та локації	<ol style="list-style-type: none">1) пастки (як події, що реагують на знаходження гравця на конкретному тайлі мапи);2) створення дверей як розділювачів кімнат всередині однієї локації, та як перешкод які можна обійти різними засобами;3) форматування окремих блоків тексту (декілька абзаців у вступі до другої локації);4) збереження забракованих подій та взаємодій на майбутнє.
----------------------------	---

Наявність подібних місць, де розробники можуть вільно перевіряти елементи системи, це звичайна практика навіть в повноцінній студійній розробці ігор. Це «закулісся» зазвичай також має швидкий доступ до всіх предметів у грі, рідше місця з ворогами відключеним ШІ та іншими об'єктами, релевантними до конкретної гри. У проєктах на RPG Maker у них можуть бути розташовані тригери окремих сюжетних виборів чи конкретних сцен.

Збереження прогресу

Базовий рушій RPG Maker MZ містить вбудовану систему збереження та завантаження прогресу. Розробник може увімкнути в налаштуваннях системи автозбереження, які будуть автоматично зберігати прогрес після завершення бою або при переходах між локаціями. Єдиною незручністю може стати можливість завантажити збережений файл лише з титульного екрана – під час гри можна зберігатись, проте щоб завантажитись доведеться виходити.

Під час збереження система, як їй і подобає, записує весь поточний стан гри, включаючи:

- поточну локацію, координати та напрямлення персонажа;
- склад команди, їх параметри, навички, предмети, валюту та статуси;
- значення всіх Змінних та Перемикачів, а також стани Подій.

За замовчуванням рушій показує 19 слотів для ручних збережень та 1 зарезервований під автозбереження, проте їх кількість та елементи відображення можна легко змінити за рахунок плагінів. Кожен слот відображає свою назву, склад команди та час проходження, як видно на рис. 4.5.

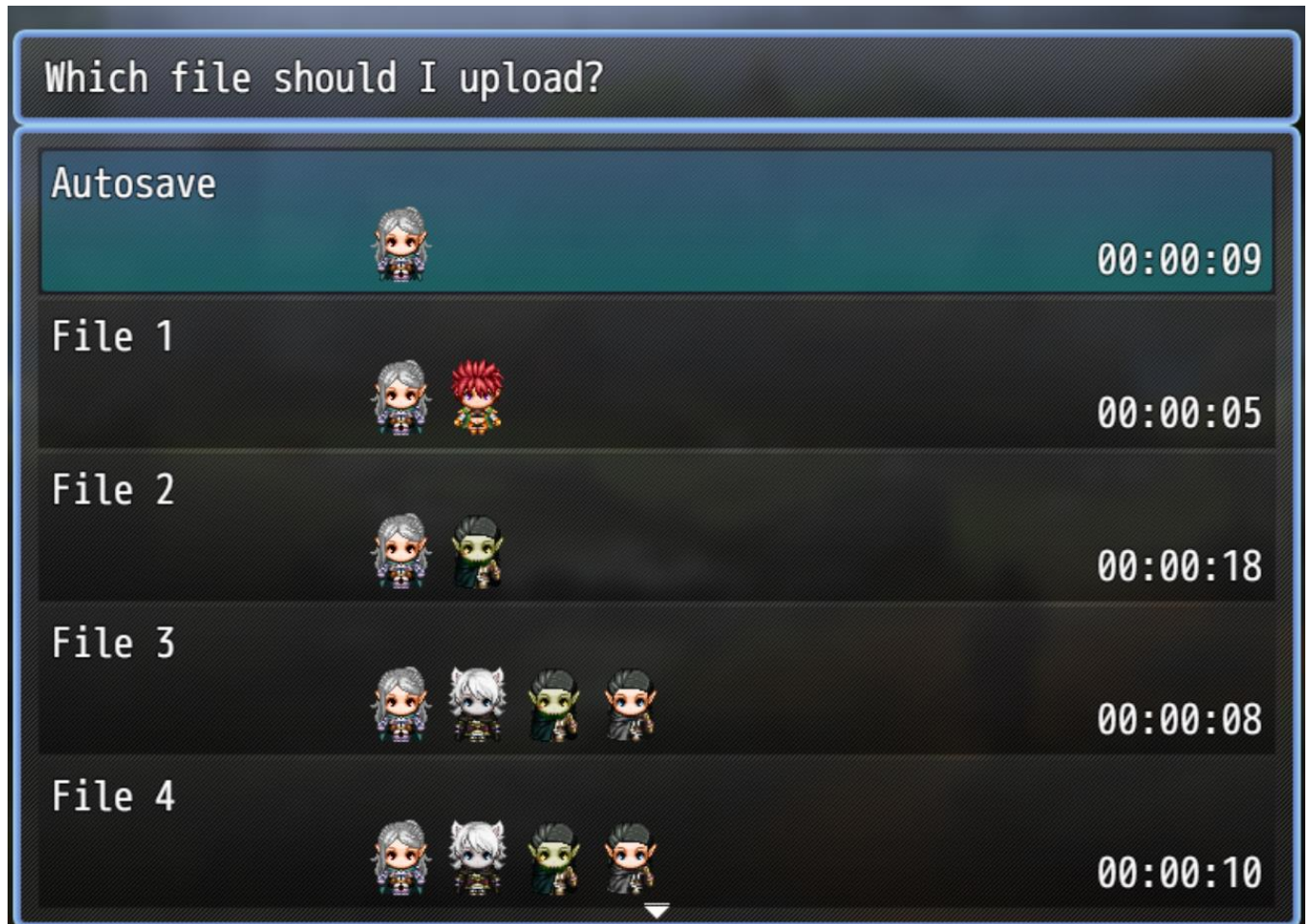


Рисунок 4.5 – Створені під час розробки та тестування сейв-файли

Під час дослідження локацій гравець може вільно зберігати прогрес у будь який момент, коли не відкрите діалогове вікно події. Окрім ручних та автозбережень, після певних сюжетних моментів відбувається примусове збереження прогресу.

Подібний підхід дуже лояльний до помилок гравця та загальної «важкості» гри, тому деякі розробники обмежують подібні можливості або ускладнюють механіку в

цілому, через збереження тільки у конкретних об'єктів, використання обмеженої кількості спеціальних предметів, їхню комбінацію чи інші підходи.

Самі файли зі збереженнями створюються рушієм локально на пристрої користувача. У випадку ПК, це окрема папка /saves у кореневій директорії експортованого проєкту. Це дозволяє переносити збереження між різними копіями гри або легко створювати їх резервні копії.

Тестування демо-версії

В рамках самого тестування та виправлення можливих помилок, весь контент демо-версії було опрацьовано як у версії з редактора, так і в експортованій «готовій». Всі локації, події та взаємодії, зазначені в третьому розділі, працюють коректно та згідно запланованим ідеям:

- переміщення між мапами зв'язано правильно, кожен вихід та вхід з локації веде до логічного в рамках локацій місця;
- порядок діалогів та сцен відтворюється як потрібно, «скрипти» не ламаються, а використані події підчищаються коли необхідно;
- бойові сцени мають помірне число «елементів рандому», на які гравець опосередковано може вплинути (1/20 або 5% шанс нанести x3 пошкодження при кожній атаці у персонажів та ворогів);
- знаходження секретів та збирання предметів на продаж чи використання (зброя, броня, спорядження, скарби) викликають приємні відчуття, навіть якщо було відомо їх розміщення.

Міnorні помилки граматики та анімацій також знайдено та виправлено, і в цілому було отримано ідеї для покращення чи зміни деяких елементів гри, переглянувши її з точки зору гравця, та в рамках повного проходження підряд.

В середньому проходження демо-версії зайняло приблизно 50 хвилин, якщо не пропускати діалоги та взаємодіяти з усіма об'єктами та NPC. Повноцінні проєкти можуть мати від декількох до більше ніж 30 годин геймплею, в залежності від основного жанру та волі розробника.

Висновки до розділу 4

В рамках розділу було продемонстровано експортування готової демо-версії на та можливі налаштування цього процесу. Було описано системні вимоги та розписано управління гри, примітивши що керування може бути змінено лише через застосуванні плагінів.

Розібрано як відбувається експорт гри та що відбувається при її запуску. Оглянуто зовнішній вигляд титульного екрану, доступні на ньому налаштування та кнопки.

Було описано принцип роботи DEV-кімнати та перелічено елементи системи, для перевірки яких її було використано. Розібрано систему збережень прогресу, її розміщення та можливості реалізації – вручну, тільки після боїв та переходів між мапами, тільки в сюжетних точках, чи окремими спеціальними методами.

На етапі тестування було декілька раз пройдено демо-версію гри повністю, від початку до кінця та взаємодіючи зі всім доступним оточенням та об'єктами. Під час проходження підмічено незначні граматичні та стилістичні помилки, які було виправлено в рамках «перед-релізної» версії.

Після завершення тестування, було вирішено що створена демо-версія відповідає запланованим очікуванням і початковим планам, та при достатньому бажанні може бути розширена до повноцінного фан-проєкту, з використанням певних доробок та полірування.

Серед потенційних перспектив розширення можна перелічити

- додавання нових локацій, ворогів та інтерактивних об'єктів;
- розширення та продовження сюжету;
- нові класи, навички, предмети;
- покращення UI та системи через плагіни.

ВИСНОВКИ

У ході виконання роботи було створено робочу демо-версію ігрового застосунку у жанрі 2D-гри з елементами візуальної новели. Для досягнення поставленої мети було виконано такі завдання, як:

- проаналізовано популярні рушії для розробки ігор, та зроблено їх порівняння в рамках потрібного жанру;
- зроблено огляд рушія RPG Maker MZ та його можливостей для розробки ігор;
- створено основний концепт, локації та персонажів для гри в рамках інструментів рушія;
- здійснена програмна реалізація ігрового застосунку можливостями рушія;
- проведено тестування гри та виправлення незначних візуальних помилок;
- розглянуто можливості подальшого розвитку гри з демо-версії.

У першому розділі КР був проведений розбір жанрів JRPG та ВН, а також розглянуті деякі популярні ігри на рушії RPG Maker. Було проведено порівняння відомих рушіїв за критеріями, необхідними для розробки запланованої демо-версії, визначено основні можливості рушія, необхідні для початку роботи.

У другому розділі було проведено широкий огляд загальних можливостей рушія RPG Maker MZ та окремих його систем, таких як малювання локацій, налаштування персонажів, логіки бойової системи та реалізацію системи подій.

У третьому розділі було розглянуто загальну структуру гри, та зроблено детальний огляд можливостей гравця в рамках створеної демо-версії. Опрацьовано застосування плагінів, графічних та аудіо-матеріалів.

У четвертому розділі було проведено експорт готової програмної реалізації та її остаточне тестування. Зроблено внесення необхідних виправлень та розглянуто можливості для подальшого розширення проєкту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. FINAL FANTASY VI. URL: <https://www.nintendo.com/us/store/products/final-fantasy-vi-switch/> (last accessed 26.04.2026)
2. To the Moon. URL: https://store.steampowered.com/app/206440/To_the_Moon/ (last accessed 26.04.2026)
3. Ib. URL: <https://store.steampowered.com/app/1901370/Ib/> (last accessed 26.04.2026)
4. OneShot. URL: <https://store.steampowered.com/app/420530/OneShot/> (last accessed 26.04.2026)
5. OMORI. URL: <https://store.steampowered.com/app/1150690/OMORI/> (last accessed 26.04.2026)
6. Visual Novel Database (VNDB). URL: <https://vndb.org/> (last accessed: 26.04.2026).
7. RPG Maker MZ Official Website. URL: <https://rpgmakerofficial.com/product/mz/> (last accessed: 27.04.2026)
8. Unity Manual. URL: <https://docs.unity3d.com/Manual/> (last accessed: 27.04.2026).
9. Godot Documentation. URL: <https://docs.godotengine.org/en/stable/> (last accessed: 27.04.2026).
10. Ren'Py Documentation. URL: <https://www.renpy.org/doc/html/> (last accessed: 27.04.2026).
11. RPG Maker MZ Documentation. URL: https://rpgmakerofficial.com/product/MZ_help-en/#t=01.html (last accessed: 27.04.2026)
12. whtdragon. Animals and Running Horses (RPG Maker Resources) URL: <https://forums.rpgmakerweb.com/threads/whtdragons-animals-and-running-horses-now-with-more-dragons.53552/> (last accessed: 28.04.2026)

13. Pixabay. Free Images and Sound Effects. URL: <https://pixabay.com/> (last accessed: 28.05.2026).
14. Freesound. URL: <https://freesound.org/> (last accessed: 28.05.2026).
15. JRPG (Japanese Role-Playing Game). URL: https://en.wikipedia.org/wiki/History_of_Eastern_role-playing_video_games#Japanese_role-playing_gamesc (last accessed: 29.04.2026)
16. RPG Maker MZ Terms of Use. URL: <https://www.rpgmakerweb.com/eula> (last accessed: 25.05.2026).
17. RPG Maker MZ Database. URL: https://rpgmakerofficial.com/product/MZ_help-en/01_08.html (last accessed: 05.06.2026).
18. RPG Maker MZ Events. URL: https://rpgmakerofficial.com/product/MZ_help-en/01_09.html (last accessed: 06.06.2026).
19. Aleem, S., Capretz, L.F. & Ahmed, F. Game development software engineering process life cycle: a systematic review. J Softw Eng Res Dev 4, 6 (2016). URL: <https://link.springer.com/article/10.1186/s40411-016-0032-7> (last accessed: 08.06.2026).
20. JavaScript Guide. Mozilla Developer Network. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> (last accessed: 11.06.2026).
21. JavaScript Reference. Mozilla Developer Network. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference> (last accessed: 11.06.2026).
22. Plugin. RPG Maker Wiki. URL: <https://rpgmaker.fandom.com/wiki/Plugin> (last accessed: 12.06.2026).
23. Isbister K. How Games Move Us: Emotion by Design. MIT Press, 2016. URL: <https://mitpress.mit.edu/9780262534536/how-games-move-us/> (last accessed: 12.06.2026).
24. Kenney Game Assets. URL: <https://kenney.nl/assets> (last accessed: 13.06.2026).

25. Salen K., Zimmerman E. Rules of Play: Game Design Fundamentals. MIT Press, 2003. URL: <https://mitpress.mit.edu/9780262240451/rules-of-play/> (last accessed: 13.06.2026).

26. D. Callele, E. Neufeld and K. Schneider, "Requirements engineering and the creative process in the video game industry," 13th IEEE International Conference on Requirements Engineering (RE'05), Paris, France, 2005, pp. 240-250, doi: 10.1109/RE.2005.58. URL: <https://ieeexplore.ieee.org/document/1531045> (last accessed: 13.06.2026).

27. Schell, J. (2019). The Art of Game Design: A Book of Lenses, Third Edition (3rd ed.). A K Peters/CRC Press. URL: <https://doi.org/10.1201/b22101> (last accessed: 13.06.2026).

28. Visual Novel. Wikipedia. URL: https://en.wikipedia.org/wiki/Visual_novel (last accessed: 13.06.2026).

29. Interactive Storytelling. Wikipedia. URL: https://en.wikipedia.org/wiki/Interactive_storytelling (last accessed: 13.06.2026).

30. Narrative Design for Games. URL: <https://gamedesignskills.com/game-design/narrative-design/> (last accessed: 13.06.2026).

31. Kanode C. M., Haddad H. M. Software Engineering Challenges in Game Development. 2012 20th International Conference on Software Engineering and Data Engineering. URL: https://www.researchgate.net/publication/262181775_Software_Engineering_Challenges_in_Game_Development (last accessed: 13.06.2026)