

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

**Освітньо-професійна програма: «Розробка
програмного забезпечення»**

Група: 4РП-08

Дипломний проєкт

**здобувача освіти денної форми навчання
РП.08.25.000.ДП**

***ХОДНЕВИЧА
МАКСИМА ОЛЕГОВИЧА***

**м. Одеса
2025 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-08

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

Розробка 2D-гри у жанрі Hack and slash з елементами RPG

Проектний матеріал складається з пояснювальної записки на 72 сторінках та графічного (презентаційного) матеріалу на 12 аркушах (слайдах)

Дипломник _____ (Ходневич М.О.)

Керівник _____ (Джабраїлов Д.В.)

Консультанти:

з економічного розділу _____ (Канський М.Ю.)

з розділу охорони праці та техніки безпеки _____ (Чорновол Н.І.)

з нормоконтролю _____ (Петрашова В.І.)

старший консультант _____ (Кривченко Ю.В.)

До захисту допущений

Голова циклової комісії _____ (Кривченко Ю.В.)

Завідувач відділення _____ (Краснокутська К.Г.)

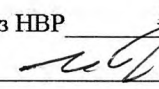
Захист «26» 06 2025 р. Протокол ЕК № 2

Оцінка ЕК 4/80

Секретар ЕК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та III
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:
Заст. дир. з НВР Беркань І.В.

« 12 » 08 2025 р.

ЗАВДАННЯ

на дипломний проєкт

Здобувачеві освіти Ходневичу Максиму Олеговичу
(прізвище, ім'я, по батькові)

1. Тема проєкту Розробка 2D-гри у жанрі Hack and slash з елементами RPG

затверджена наказом по коледжу від «14» листопада 2024р. № 246

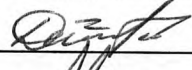
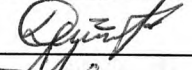


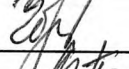
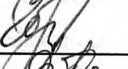
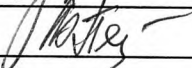
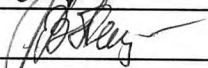
2. Термін здачі закінченого проєкту _____

3. Вихідні данні до проєкту Використання програмного рушія Unity; Використання мови програмування C# та бібліотек Unity для розробки гри; Реалізація основних ігрових механік гри в жанрі Hack and slash; Реалізація різних рівнів для гри; Реалізація генерація навколишнього середовища для рівнів; Реалізувати декілька видів ворогів; Реалізувати систему розвитку персонажу гравця; Реалізація інтерфейсу гравця.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)
Аналіз систем RPG в сучасних іграх; Обрання засобів розробки; Формування ігрового концепту гри; проєктування основних елементів гри; проєктування системи RPG для розробляємої гри; Реалізація спроєктованих елементів гри; Тестування гри.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)
Аналіз існуючих RPG-систем; Обрання засобів розробки; Концепція ігрового процесу; Основні елементи гри; Реалізація рівнів та генерації довкілля; Реалізація ворогів; проєктування RPG-системи гри; Реалізація RPG-системи; Тестування гри; Скріншоти розробленого проєкту.

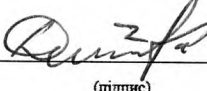
6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Джабраїлов Д.В.		
Економічний розділ	Канський М.Ю.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання 02.06.2025


Керівник

Джабраїлов Д.В.


(підпис)

Завдання прийняв до виконання


Ходневич М.О.


(підпис)

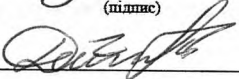
КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Вступ. Постановка мети та задач проектування	14.05.2025	Виконав
2	Проведення аналізу RPG-систем у сучасних іграх	16.05.2025	Виконав
3	Обрання програмного забезпечення для розробки	17.05.2025	Виконав
4	Формування концепту ігрового процесу гри	20.05.2025	Виконав
5	проектування основних елементів гри та RPG-системи	22.05.2025	Виконав
6	проектування графічної частини	28.05.2025	Виконав
7	Реалізація елементів гри	01.06.2025	Виконав
8	Інтеграція інтерфейсів та графіки	03.06.2025	Виконав
9	Відлагодження модулів та ігрових механік	06.06.2025	Виконав
10	Тестування працездатності елементів гри	10.06.2025	Виконав
11	Виправлення виявлених помилок	11.06.2025	Виконав
12	Аналіз результатів, підготовка слайдів презентації	12.06.2025	Виконав
13	Економічні розрахунки та питання з охорони праці	13.06.2025	Виконав
14	Підготовка графічної частини проекту	14.06.2025	Виконав
15	Підготовка проекту до захисту та тестування гри	16.06.2025	Виконав

Дипломник


(підпис)

Керівник


(підпис)

ЗМІСТ

Вступ.....	7
1 Основний розділ.....	8
1.1 Аналіз систем RPG в сучасних іграх.....	8
1.1.1 Загальний аналіз систем RPG в сучасних іграх.....	8
1.1.2 Аналіз системи RPG в грі Dead Cells.....	10
1.1.3 Аналіз системи RPG в Moonlighter.....	11
1.1.4 Аналіз системи RPG в Ender Lilies: Quietus of the Knights.....	12
1.1.5 Результати аналізу.....	13
1.2 Обрання засобів розробки.....	14
1.2.1 Вибір ігрового рушія як основної платформи розробки.....	14
1.2.2 Реалізація ігрової логіки на мові програмування C#.....	15
1.2.3 Інтегроване середовище розробки Visual Studio.....	15
1.2.4 Графічне оформлення проєкту засобами Adobe Photoshop.....	15
1.2.5 Загальна оцінка ефективності обраного інструментарію.....	16
1.3 Формування ігрового концепту гри.....	16
1.4 Проєктування основних елементів гри.....	19
1.5 проєктування системи RPG.....	24
1.6 Реалізація спроєктованих елементів гри.....	30
1.7 Тестування гри.....	48
2 Економічний розділ.....	50
2.1 Резюме.....	50
2.2 Розрахунок ціни програмного продукту нормативним методом.....	50
2.2.1 Визначення трудомісткості розробки програмного забезпечення ...	50
2.2.2 Розрахунок ціни програмного продукту.....	53
3 Розділ охорони праці та техніки безпеки.....	55
3.1 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці програмного комплексу.....	55
3.2 Гігієнічні вимоги до виробничого середовища.....	55
3.2.1 Приміщення.....	55

					<i>РП 08. 25 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

3.2.2 Освітлення.....	56
3.2.3 Рівень шуму	56
3.2.4 Мікроклімат	56
3.2.5 Електробезпека	56
3.3 Вимоги до організації робочого місця працівника.....	57
3.4 Пожежна безпека.....	58
3.5 Організація евакуації працівників під час надзвичайних ситуацій	59
Висновки.....	60
Перелік використаних інформаційних джерел.....	61
Додаток А. Лістинг коду основних модулів гри мовою C#.....	62
Додаток Б. Слайди мультимедійної презентації.....	67

ВСТУП

Ігрова індустрія є однією з найприбутковіших у сфері ІТ. Із розвитком персональних комп'ютерів, смартфонів та швидкісного Інтернету кількість гравців у всьому світі стрімко зростає. Це спричиняє активний попит на нові ігрові проєкти та підвищує актуальність професії розробника відеоігор.

Окрему нішу займає інді-розробка – створення ігор невеликими командами або навіть поодинокими розробниками. Завдяки доступним інструментам, численним гайдам та відкритим навчальним матеріалам в Інтернеті, навіть початківці мають можливість реалізувати повноцінні ігрові проєкти.

Найбільш популярним серед інді-розробників є рушій Unity, який надає широкий функціонал для створення 2D та 3D ігор. Його перевагами є інтуїтивно зрозумілий інтерфейс, підтримка різних платформ, можливість розширення за допомогою плагінів та наявність великої спільноти. Саме тому для реалізації цього дипломного проєкту було обрано Unity у поєднанні з мовою програмування C# та графічним редактором Adobe Photoshop.

Темою даного дипломного проєкту є розробка 2D-гри у жанрі Hack and slash з елементами RPG. Такий вибір обумовлений як популярністю цього жанру серед гравців, так і його технічною придатністю для реалізації в рамках індивідуального проєкту. Проєкт дозволяє на практиці застосувати набуті знання з розробки програмного забезпечення, дизайну ігрових механік, роботи з графікою, а також розширити компетенції у сфері геймдизайну.

Особливістю цієї гри стане унікальна механіка подвійного здоров'я (персонажа та рюкзака), прокачка характеристик за зібрані монети, випадкова генерація рівнів та ворогів, а також стилізована графіка, створена вручну. Усе це має забезпечити глибину геймплею, реіграбельність та впізнаваний візуальний стиль.

У результаті реалізації цього проєкту буде створено повноцінну гру, що продемонструє здатність до самостійної роботи над усіма етапами створення програмного продукту – від ідеї до реалізації та тестування.

					РП 08. 25 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

1 ОСНОВНИЙ РОЗДІЛ

1.1 Аналіз систем RPG в сучасних іграх

1.1.1 Загальний аналіз систем RPG в сучасних іграх

RPG є одним із найпопулярніших жанрів у сучасній ігровій індустрії. Особливістю RPG є те, що в них гравець сам вирішує, як буде розвиватися персонаж, а саме: як він буде прокачувати персонажа, протидіяти різним ворогам та йти по сюжету. Системи RPG також можуть бути інтегровані в різні жанри, зокрема в hack and slash, що робить гру набагато цікавішою та більш захопливою.

Однією з найголовніших особливостей RPG є система прокачки персонажа. У класичних RPG часто використовувалася система досвіду. При підвищенні рівня персонажа в міру проходження гри він міг покращувати свої базові характеристики, використовуючи свій досвід, такі як урон, здоров'я або інтелект, чи вивчати якісь нові навички та здібності. Але також існують інші системи. Наприклад, система RPG, в якій ти те, що ти носиш. У ній здібності, навички та базові характеристики персонажа залежать від того, що на ньому надіто в конкретний момент гри, і гравець сам обирає, яку броню чи зброю залишити для себе після випадіння її з боса або які предмети йому потрібно зараз створити.

Бойові системи є серцем будь-якої RPG, особливо у темі «Розробка 2D-гри у жанрі Hack and slash з елементами RPG», де акцент буде робитися на динамічний екшен. У моєму проєкті бій відбувається в реальному часі зі швидкою реакцією гравця, що і робитиме ігровий процес динамічним і захоплюючим.

Сучасні системи RPG характеризуються можливістю нелінійного проходження. Сюжет гри може розвиватися у відкритому світі, де гравець сам обирає, куди йому йти та як саме проходити цю гру. Така свобода вибору також доповнюється побічними завданнями та різноманітністю опрацьованих локацій. Також сюжет гри може бути нелінійним, і від вибору персонажа може залежати, на яку саме кінцівку він пройде гру. Ще сучасні системи RPG дають можливість взаємодії з неігровими персонажами або NPC, з якими можна побудувати різноманітні відносини, і саме це створює ілюзію живого світу. Для моєї гри

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

унікальність кожного нового проходження буде в тому, що локації будуть генеруватися випадково, а гравець сам буде обирати, як прокачувати свого персонажа.

Важливою частиною системи RPG є баланс. А саме — спроможність розробника збалансувати гру і систему RPG, а якщо точніше, то систему прокачування персонажа так, щоб при цьому персонаж не був надто сильним і через декілька прокачаних властивостей або базових характеристик не вбивав усіх ворогів за один удар і при цьому сам не був би безсмертним, і водночас не був занадто слабким і не в змозі нікого перемогти, через що гра здавалася б непрохідною. У моєму проєкті вся увага буде акцентована на монетках і магазині. З монстрів випадатимуть монетки, які потім витратимуться в магазині для прокачування персонажа, а саме його здоров'я чи урону — вже на вибір гравця. У цій системі важливо зберегти баланс, щоб монеток вистачало на прокачування персонажа після проходження рівнів, або щоб прокачування не додавало надто багато урону, і персонаж не вбивав монстрів після першого прокачування з одного удару.

При розробці ігор з системою RPG система розвитку або прокачування персонажа має залишатися гнучкою, але водночас достатньо спрощеною, щоб не перевантажувати гравця надмірною кількістю параметрів. Інтерфейс повинен бути інтуїтивно зрозумілим, щоб гравець міг добре орієнтуватися у вміннях, екіпіровці та характеристиках персонажа. Бойова система потребує балансу між тактичністю та динамікою, щоб гравець сам вирішував, як саме йому розправлятися з ворогами. Крім того, важливу роль відіграє система здобичі у вигляді монет або нової зброї чи броні та створення предметів, яка підтримує інтерес до гри, даючи змогу відчувати прогрес та нагороду за просування по сюжету.

Системи RPG у сучасних іграх можуть добре інтегруватися в інші ігрові жанри. Особливо це помітно у hack and slash, де швидкий, динамічний бій доповнюється елементами розвитку персонажа. Система RPG, яка робить кожне проходження гри унікальним, змушує гравця пройти гру ще раз, щоб вийти на іншу кінцівку або спробувати іншу прокачку персонажа, і це завжди буде для

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

нього цікавим досвідом, а для розробника впровадження системи RPG у свою гру стане також цікавим досвідом.

1.1.2 Аналіз системи RPG в грі Dead Cells

Почнемо аналіз з гри Dead Cells, яка є 2D-платформером у жанрі roguelite з елементами hack and slash та розроблена студією Motion Twin. Гра реалізує низку RPG-механік, які підходять для динамічного геймплею. У грі реалізовано тимчасову прогресію всередині кожного проходження. Коли гравець досліджує рівень, він знаходить scrolls of power, які дозволяють збільшити один із трьох параметрів: жорстокість, тактику та виживання. Жорстокість впливає на шкоду від зброї ближнього бою та пасток. Тактика покращує шкоду від зброї дальнього бою та активних умінь. Виживання підвищує кількість здоров'я та ефективність щитів. Ця система дозволяє гравцеві гнучко прокачувати свого персонажа в кожному забігу.

Також елементами RPG у Dead Cells є можливість у кожному проходженні знайти чи придбати зброю, щити, активні вміння та амулети, які мають різні рівні рідкості. Окрему роль відіграє система мутацій, а саме пасивних умінь, які відкриваються поступово та обираються на старті й у спеціальних зонах.

Окрім тимчасової прокачки під час забігу, Dead Cells також має систему перманентного прогресу, що реалізована через збір клітин, які випадають після перемоги над ворогами. Ці клітини можна витратити на розблокування нових видів зброї та предметів, покращення шансів випадіння якіснішого спорядження, розширення кількості та ефективності лікувальних фляг. Завдяки цій системі гравець відчуває постійний прогрес, навіть якщо забіг був невдалим.

У Dead Cells система RPG гармонійно інтегрована в динамічну бойову модель. Прокачка персонажа збалансована, а вибір спорядження та мутацій робить кожен забіг унікальним. Переглянути скріншот гри Dead Cells можна на рисунку 1.1:

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10



Рисунок 1.1. Скріншот гри Dead Cells

1.1.3 Аналіз системи RPG в Moonlighter

Moonlighter – це 2D-гра, яка поєднує в собі жанри екшн-RPG та економічного симулятора. Гра має яскравий візуальний стиль, натхненний піксель-артом, що створює атмосферу старої школи, поєднану з сучасною динамікою. Прогрес у Moonlighter досягається через покращення спорядження, крафт нової зброї та броні. Кожен тип зброї має свою унікальну механіку бою, що дозволяє гравцеві мати свій бойовий стиль, який йому до вподоби. Наприклад, одні гравці можуть віддати перевагу швидким атакам кинджалами, тоді як інші – повільним, але потужним ударам великим мечем. Також важливим RPG-елементом у Moonlighter є його економічна система, в якій гравець може продавати трофеї з підземель у власній лавці, встановлюючи ціни відповідно до попиту, що робить персонажа не тільки бійцем, а й підприємцем. Цей аспект гри вимагає стратегічного мислення та здатності адаптуватися до змін ринку.

Також у грі є крафт, зачарування спорядження, купівля зілля та розвиток міста. Місто, у свою чергу, розвивається разом із гравцем, відкриваючи нових персонажів, магазини та функціонал, що робить кожне повернення з підземель по-своєму унікальним. Система RPG Moonlighter базується на ресурсному

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

менеджменті, персоналізації спорядження та виборі стилю гри, що є доволі нестандартною реалізацією системи RPG. Скріншот гри зображений на рисунку 1.2:



Рисунок 1.2. Скріншот гри Moonlighter

1.1.4 Аналіз системи RPG в грі Ender Lilies: Quietus of the Knights

Ender Lilies: Quietus of the Knights – це 2D action-RPG платформер з глибокою атмосферою та історією. Гравець керує дівчинкою Лілі, яка сама нікого не атакує, а викликає духів вмерлих лицарів, кожен із яких має свої унікальні здібності. Це є основною механікою системи RPG гри.

Прогресувати у грі можна шляхом отримання нових духів, яких можна комбінувати у слотах активних та пасивних навичок. Одні духи призначені для ближнього або дальнього бою, інші – щоб лікувати або виконувати різні допоміжні функції. Також кожного духа можна покращувати, що робить їх більш ефективними.

Ще Лілі поступово у грі підвищує власний рівень та покращує різні характеристики. Додатково гравець може знайти реліквії, які дають пасивні ефекти, наприклад, збільшення швидкості відновлення здоров'я.

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

Ender Lilies: Quietus of the Knights має традиційний рівневий прогрес, що є більш класичним підходом до реалізації системи RPG. Але при всьому цьому можна кастомізувати бойовий стиль під себе та створити власну бойову стратегію. Скріншот гри можна побачити на рисунку 1.3:



Рисунок 1.3. Скріншот гри Ender Lilies: Quietus of the Knights

1.1.5 Результати аналізу

В усіх проаналізованих іграх були різні підходи до реалізації систем RPG. У Dead Cells RPG-механіка має тимчасовий характер, бо прокачування персонажа відбувається тільки під час забігу, де гравець обирає персонажу спорядження та мутації. Також відбувається відкриття нових можливостей між забігами, що створює баланс між динамікою й прогресом. Moonlighter відходить від класичної реалізації системи RPG і робить акцент на прогрес через економіку, крафт та управління магазином, де розвиток визначається якістю спорядження та ефективністю використання ресурсів. Ender Lilies: Quietus of the Knights використовує більш традиційний підхід до реалізації системи RPG, де у персонажа підвищується рівень і гравець може покращувати навички персонажа. Усі три гри ілюструють різні успішно інтегровані системи RPG і можуть слугувати орієнтиром при створенні власної системи.

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

Unity також підтримує кросплатформеність розробки проєктів, що дає змогу експортувати збірку не тільки під Windows, а й під Android, iOS або WebGL. Це дає проєкту можливість отримати більшу аудиторію на різних платформах. Ще Unity є дуже гнучким рушієм розробки, що дозволяє додавати сторонні бібліотеки, системи частинок, а також візуальні ефекти, які легко інтегруються в проєкт.

1.2.2 Реалізація ігрової логіки на мові програмування C#

Основною мовою програмування для логіки гри було обрано C#, бо вона є основною мовою програмування в рушії Unity. Вона має функціональні можливості для створення ігрової механіки будь-якої складності.

Також через те, що вона є популярною по всьому світу, існує дуже багато гайдів та посібників з розробки на цій мові програмування, і якщо виникають якісь проблеми, знайти їх вирішення на різних форумах буде не дуже складно. Це особливо корисно під час індивідуальної та навчальної розробки.

1.2.3 Інтегроване середовище розробки Visual Studio

Для написання коду буде використовуватися середовище Visual Studio, яке є офіційно рекомендованим IDE для Unity. У Visual Studio є повноцінна інтеграція з проєктом, автоматичне розпізнавання класів, методів та змінних, а також надаються зручні інструменти для відлагодження. Visual Studio значно прискорює процес написання коду через підсвічування синтаксису, автозаповнення та підтримку розширень, а також зменшує кількість помилок.

1.2.4 Графічне оформлення проєкту засобами Adobe Photoshop

Для створення графічних елементів було прийнято рішення використати програму Adobe Photoshop. Усі спрайти, задні фони, елементи інтерфейсу та ефекти будуть створені вручну за допомогою цього редактора. Photoshop дозволяє працювати з прозорістю, шарами та масками, що є важливим для створення якісної 2D-графіки. Також Photoshop має високу точність візуального налаштування, що дає змогу досягти єдиного стилю для всієї гри.

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

1.2.5 Загальна оцінка ефективності обраного інструментарію

Комбінація Unity, C#, Adobe Photoshop та Visual Studio створює ефективне та сучасне середовище для розробки 2D-ігор, бо обрані інструменти повністю відповідають вимогам проєкту. Вони дозволять реалізувати бойові механіки, систему RPG, керування персонажем та створити власну графіку. Також допоможуть усунути всі баги та інші проблеми, які можуть виникнути при розробці проєкту. Такий набір програмного забезпечення є оптимальним вибором для розробки власної інді-гри.

1.3 Формування ігрового концепту гри

Після того, як було визначено інструментарій для розробки гри, треба розробити концепцію ігрового процесу проєкту. Розроблюваний проєкт представляє собою 2D-гру у жанрі hack and slash з елементами RPG, орієнтовану на одиночне проходження з поступовим розвитком персонажа та бойовими викликами на рівнях, які гравцю буде потрібно пройти. Основна увага буде зосереджена на динамічних боях та гнучкій системі розвитку персонажа.

Жанром гри є hack and slash, який передбачає акцент на ближньому бою, швидких атаках, ухиленнях, а також на навичках реакції та прийнятті рішень у реальному часі. Також будуть додані RPG-елементи, як-от випадіння монет з монстрів, покращення базових характеристик персонажа у магазині між рівнями та рандомний спавн ворогів і об'єктів на рівнях, що дозволить зробити геймплей більш глибоким, персоналізованим і стратегічним.

Гравець керує персонажем, який буде лінійно йти через низку рівнів, наповнених ворогами. Персонажем буде лицар Фурі-кішка з фіолетовою шерстю та рюкзаком за плечима. Головною метою гравця є доставити корону до принцеси-кішки у своєму рюкзаку, пройшовши всі рівні та вбивши всіх ворогів. На рисунку 1.5 можна побачити перший концепт-арт головного персонажа гри, дизайн якого потім буде змінено:

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

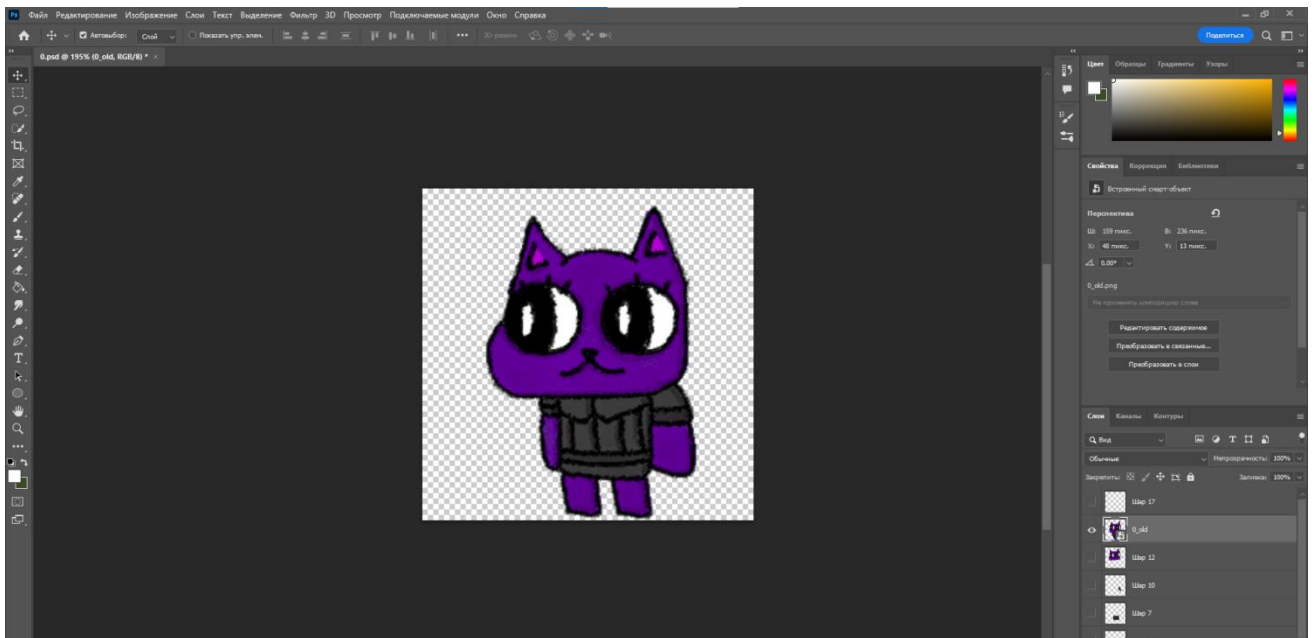


Рисунок 1.5. Перший концепт-арт головного персонажа гри в програмі Adobe Phoshop

Тематичним стилем гри буде фентезі, а всі спрайти будуть намальовані вручну у програмі Photoshop вугляним пензлем, що підкреслює приналежність гри до інді-розробки. Натхнення для малювання спрайтів буде братися з гри Cult of the Lamb. Графіка цієї гри виконана у стилі, який нагадує мультяшні ігри. У грі використовується візуальна естетика, що нагадує ігри на кшталт Don't Starve, але у яскравішому та стилізованішому виконанні.

Унікальним елементом гри будуть дві полоски здоров'я: одна відповідає за персонажа, а інша за рюкзак. Саме ця частина геймплею буде виділяти гру серед інших інді-проектів на ринку відеоігор.

Так як гравцю важливо буде зберегти не тільки власне життя в цілості та сохрнаності, а й цілісність корони у рюкзаку, то у персонажа буде два колайдери: один спереду, що відповідає за здоров'я персонажа, а другий ззаду за здоров'я рюкзака. У персонажа та рюкзака буде по дві одиниці здоров'я, причому здоров'я рюкзака матиме синій колір, а здоров'я персонажа червоний. Якщо закінчиться те чи інше здоров'я, гравець програє.

Під час проходження гри та вбивства ворогів з них будуть випадати монети, які можна буде витратити у магазині. Локація магазину буде спавнитися після декількох пройдених рівнів. В магазині монети можна буде витратити на

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

збільшення урону меча або збільшення здоров'я на вибір: рюкзака чи самого персонажа. Вибір прокачки персонажа під час проходження гри дасть змогу гравцю безпосередньо впливати на свій стиль геймплею. На рисунку 1.6 можна побачити, як саме це буде виглядати на екрані для гравця або загальний концепт ігрового інтерфейсу гри:

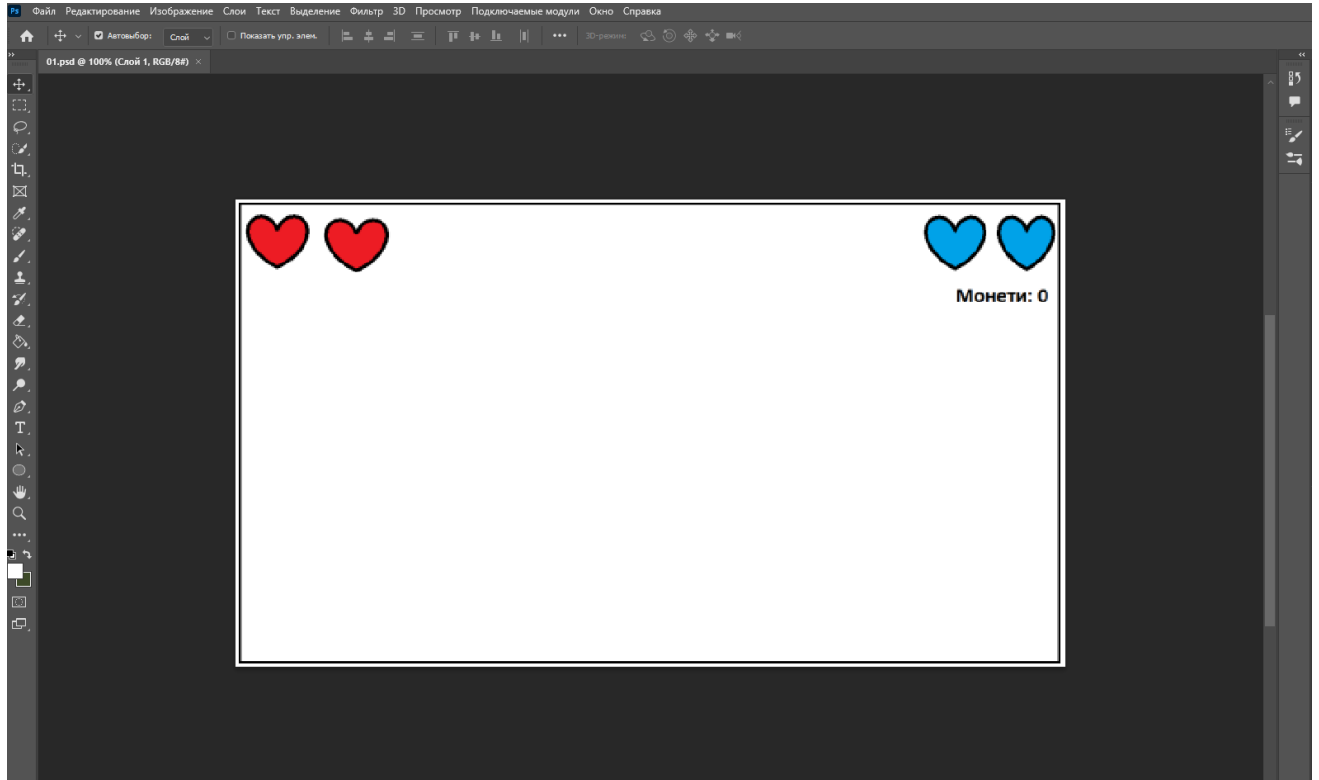


Рисунок 1.6. Загальний концепт ігрового інтерфейсу гри в програмі Adobe Phoshop

У сучасному ігровому середовищі дедалі важливішою стає не лише технічна складова, а й візуальна унікальність. Атмосфера, створена художнім стилем, кольоровою палітрою та анімаціями, здатна не просто передати настрій гри, а й визначити її ідентичність. У проєкті застосовується стилізований вугільний візуал, який підкреслює інді-характер розробки та викликає в гравця емоційний відгук.

У візуальній частині особливу увагу приділено деталям: від блиску монет до дрижання трави під час проходження. Такі нюанси на перший погляд здаються незначущими, але саме вони формують глибину ігрового досвіду. Гравець відчуває, що світ гри живий і реагує на його дії, що сприяє повнішому зануренню в процес.

Особливо важливо створити консистентний стиль, який не лише прикрашає гру, а й допомагає гравцеві краще орієнтуватися. Наприклад, вороги мають чітко відмінні силуети, завдяки чому їх можна впізнати навіть у динамічних сценах бою. Це покращує геймплей та знижує фрустрацію від випадкових поразок.

Концепція ігри базується на поєднанні динамічного бойового геймплею з елементами RPG, що реалізуються через прокачку персонажа. Унікальні механіки, як окреме здоров'я рюкзака та персонажа, а також художній стиль, надають проєкту впізнаваності та емоційної виразності. Такий підхід дозволяє створити цілісну візуально привабливу гру, орієнтовану на інді-аудиторію.

1.4 Проєктування основних елементів гри

Після визначення ігрового концепту гри потрібно спроектувати основні елементи, її механіки та інші особливості. Після цього можна буде сформулювати загальні ігрові концепти окремих основних елементів гри та вже за ними почати розробку самої гри.

Проєктування є дуже важливим етапом, що закладає фундамент майбутньої ігрової логіки, динаміки та загального враження від проєкту.

Пересування персонажа буде відбуватися за допомогою клавіш WASD, а на пробіл ривок, за допомогою якого гравець зможе ухилитися від атак та швидше пересуватись. Атака персонажа буде здійснюватися лівою кнопкою миші за допомогою меча.

Таке управління є інтуїтивно зрозумілим і знайомим більшості гравців, що дозволяє швидко адаптуватися до ігрового процесу без потреби в додатковому навчанні.

Інтерфейс користувача в інді-іграх виконує не лише інформативну функцію, а й є важливою частиною художнього висловлення. Добре продуманий UI дозволяє гравцеві швидко зрозуміти, які дії він може виконати, які ресурси має та які виклики стоять перед ним. Водночас, інтерфейс не повинен перевантажувати екран і відволікати від самого геймплею.

У проєкті був застосований підхід мінімалізму: усі елементи на екрані розташовані інтуїтивно, мають унікальне, але зрозуміле візуальне оформлення.

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

Наприклад, серця персонажа та рюкзака різного кольору і розташовані так, щоб гравець не плутався під час бою.

Це рішення спрямоване на зменшення візуального навантаження та підвищення концентрації на дії, що відбувається у грі.

Крім цього, у грі будуть реалізовані підказки, що з'являються лише в момент потреби, наприклад, коли гравець наближається до об'єкта, з яким можна взаємодіяти. Це підвищує інтуїтивність взаємодії без потреби в окремому навчальному екрані, що може відштовхнути недосвідченого гравця.

Такий підхід дозволяє зробити гру доступною як для новачків, так і для досвідчених користувачів, які цінують динамічність та чистоту інтерфейсу.

Гра буде починатися з початкової локації, де й заспавиться персонаж. Пройшовши її, персонаж потрапить на навчальну локацію, де його чекатиме неігровий персонаж старий кіт з одним оком. З ним можна буде поговорити, і в діалозі він пояснить гравцеві сюжет і сенс гри, а також як саме грати. Він розповість про те, що пересування здійснюється клавішами WASD, а на пробіл можна виконати ривок, завдяки якому можна ухилятися від атак монстрів, та що на ліву кнопку миші атака.

Використання діалогового NPC на початку гри дозволяє органічно поєднати елементи сюжету з навчанням, не порушуючи загальну атмосферу гри.

Потім він почне розповідати, що зверху є тренувальна локація, де можна відточити свої навички, а також розповість про сенс забігу гравця, яким є перемога над усіма ворогами та доставка корони принцесі. Також він пояснить механіку сердець і те, що у головного героя є дві смужки здоров'я: одна відповідає за здоров'я рюкзака, а інша за здоров'я персонажа. Потім він побажає удачі та завершить свій монолог.

Подібне подання інформації допомагає гравцеві краще зануритися в світ гри, підкріплюючи мотивацію і цілісність наративу.. На рисунку 1.7 представлена блок-схема діалогу з NPC:

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20



Рисунок 1.7 Блок-схема діалогу з NPC

Зм.	Арк.	№ докум.	Підпис	Дата

Якщо піти нагору, то там буде тренувальна локація, в якій розміщуватимуться манекени, і гравець зможе відточити свої навички перед тим, як вступати в бій з монстрами. Це безпечна зона, де гравець зможе випробувати різні типи атак, навчитись ухилятися та краще зрозуміти механіку бою. Тренування у цій локації дозволяє гравцеві поступово адаптуватися до ігрового процесу, не боячись за втрату здоров'я чи ресурсів. Якщо з навчальної локації гравець піде не нагору, а праворуч, то там вже буде перша бойова локація, а після її проходження буде друга. Коли гравець пройде дві бойові локації, на нього чекатиме магазин, де він зможе прокачатися за допомогою монет, які випали з ворогів на перших двох бойових локаціях.

Ця система нагородження стимулює гравця ретельно досліджувати рівні та боротися з усіма ворогами, аби накопичити якомога більше ресурсів. Цей підхід мотивує до активної гри та нагороджує старанного гравця, підтримуючи його інтерес до подальшого проходження. Потім він зможе йти далі і знову проходити дві бойові локації. Далі знову магазин, а після нього ще одна фінальна бойова локація.

Кожна бойова локація у грі буде унікальною, тому що на кожній бойовій локації буде випадковий спавн об'єктів. Об'єктами будуть камінь та коробка. Коробка ламатиметься від двох ударів меча персонажа, а камінь від трьох ударів. Інколи ці об'єкти можуть блокувати прохід. Спавн цих об'єктів робитиме не тільки кожну локацію унікальною, а й кожне проходження гри. Це сприяє підвищенню реіграбельності, оскільки гравець постійно стикається з новими ситуаціями, які вимагають різних тактик та рішень. Таким чином, навіть при повторному проходженні гравець не знатиме точної розкладки об'єктів, що забезпечить елемент несподіванки.

Після фінальної, п'ятої бойової локації буде кінцева локація, в якій на гравця очікуватиме кішка-принцеса, чекаючи на свою корону. Віддавши її, вона подякує гравцеві, і, пішовши далі, гра завершиться. Ця сцена завершує ігрову подорож емоційною нотою, яка додає змісту всім попереднім діям гравця. У фіналі буде подяка за гру та кнопка, яка запропонує перепройти гру ще раз. Повторне

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

проходження дозволить випробувати інші стратегії розвитку та взаємодії з ворогами, роблячи кожну нову гру відмінною від попередньої. На рисунку 1.8 представлена блок-схема всіх локацій гри:

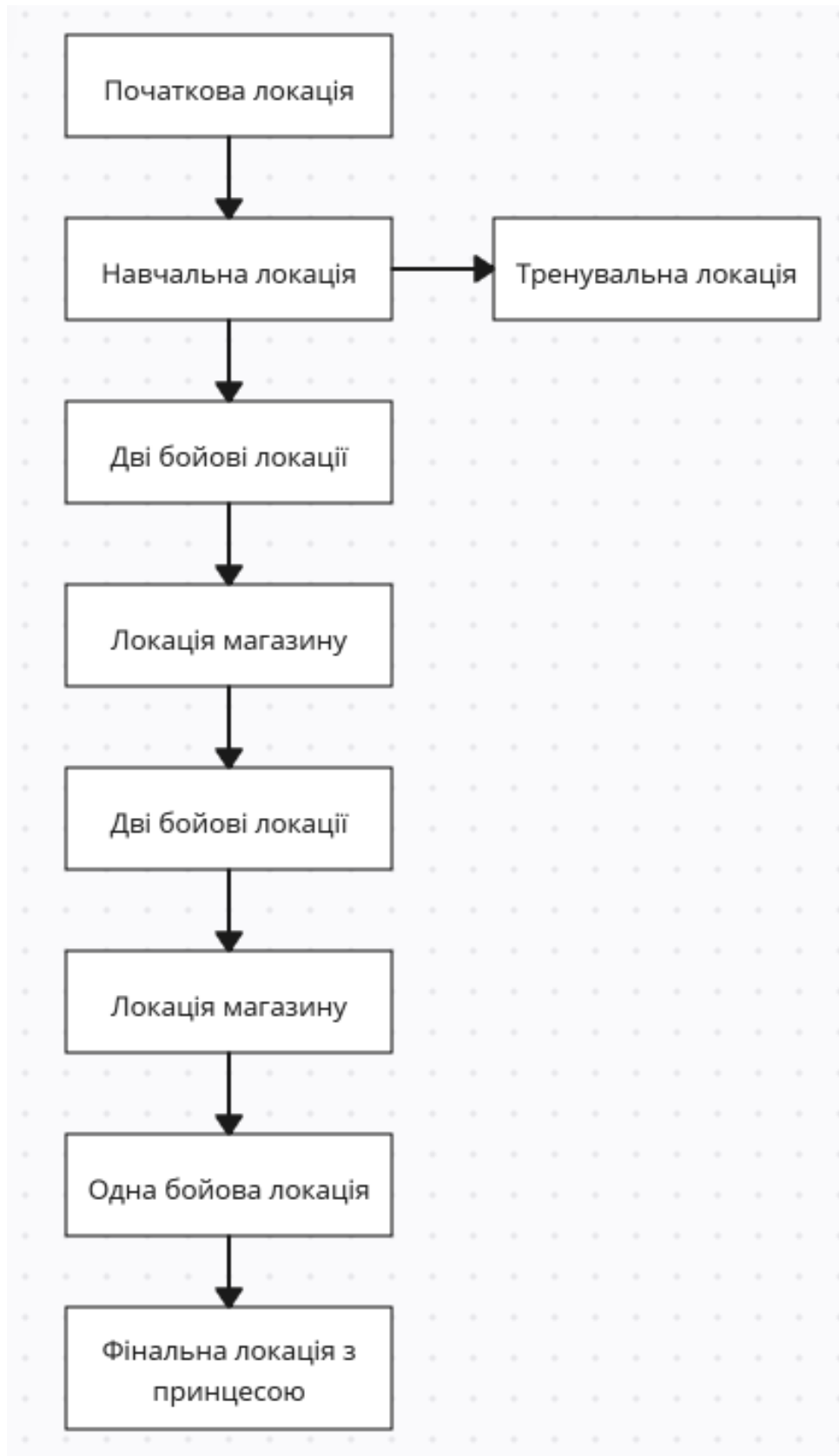


Рисунок 1.8. Блок-схема всіх локацій гри

З кожною пройденою бойовою локацією рівень складності гри зростатиме, тому що на кожній наступній бойовій локації будуть спавнитись сильніші вороги:

- Маленька гусениця: найслабший монстр, який з'являтиметься на всіх локаціях, саме з неї почнеться перша бойова локація;
- Велика гусениця: подібна до маленької, але з більшим запасом здоров'я та швидшим пересуванням;
- Вогняний пень: нерухомий ворог, який атакує вогняними шарами в усі сторони через певні проміжки часу;
- Жаба: ворог із найбільшим запасом здоров'я, який швидко стрибає у бік гравця;
- Летюча миша: ворог, що, навпаки, віддаляється від гравця та запускає вогняний шар у його бік.

Окрім базових механік, в грі буде реалізовано багато інших дрібниць, таких як різні анімації та звуки під час проходження. Наприклад, кожен тип ворога матиме унікальні звукові ефекти смерті та характерні візуальні ефекти, що зробить ігровий процес більш емоційно насиченим. Взагалі, завдяки поєднанню динамічного бою, системи прокачки персонажа, а також продуманої структури рівнів, гра створюватиме захопливий ігровий досвід, що мотивуватиме гравця повертатися до неї знову. Кожен новий забіг відкриватиме можливість протестувати іншу стратегію, завдяки чому навіть повторне проходження залишатиметься цікавим та свіжим.

1.5 проєктування системи RPG

Система RPG, яка буде розроблятися в 2D-грі у жанрі hack and slash, виконуватиме функцію розвитку персонажа, що дозволяє гравцеві адаптувати стиль проходження, відчути поступовий прогрес та забезпечити унікальність кожного нового забігу. Вона є дуже важливим компонентом геймплею, що підтримує інтерес до гри. Система прокачки створена таким чином, щоб давати відчуття нагороди за кожне прогрес у грі, що стимулює продовжувати гру далі.

Після початкової сцени і перед проходженням рівнів з ворогами гравець буде зустрічати неігрового персонажа, що виконує роль наставника. Він пояснює, що

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

від нас потребується, ігрові механіки та управління. Такий підхід додає інтерактивного навчання, щоб запобігти окремим інструкційним екранам. Крім цього, NPC-наставник виконує важливу сюжетну функцію, підкреслюючи атмосферу гри та допомагаючи гравцеві зануритися у вигаданий світ.

Після діалогу з NPC гравець буде потрапляти на послідовні рівні, де йому потрібно вбивати ворогів та збирати з них випавші монети. Вони випадають у випадковій кількості та будуть основною валютою у грі. Також ці монети будуть основним елементом системи RPG, оскільки саме вони дозволять гравцеві проводити покращення свого персонажа. Кожна зібрана монета має значення, що мотивує вступати в бій навіть зі всіма ворогами.

Після проходження кожних двох рівнів буде з'являтися магазин. Магазин – це локація, у якій гравець може витратити монетки на один із трьох типів покращень:

- Збільшення урону від меча: вороги будуть вмирати швидше, що буде змінювати темп бою;
- Збільшення здоров'я персонажа: підвищує витривалість гравця у бою;
- Збільшення здоров'я рюкзака: підвищує міцність рюкзака в бою;

Кожне з цих покращень змінює стиль гри, дозволяючи гравцеві підлаштуватися під власні уподобання – грати агресивніше, витримувати більше шкоди або зосередитись на збереженні ресурсу.

Цикл «два рівні, а потім знову магазин» повторюється двічі, що забезпечує ритмічність процесу та дозволяє поступово нарощувати силу персонажа. Така структура дозволяє зберігати темп геймплею без надмірного ускладнення, залишаючи систему RPG інтуїтивною та зрозумілою.

Ігрова сесія складається з фіксованої кількості рівнів. Після кожної пари оновлення характеристик, а на завершальному рівні фінальний виклик. Після його подолання гравець доставляє корону принцесі, завершуючи гру. Таким чином, система RPG є не лише геймплейною складовою, а й сюжетною опорою, що формує цілісний ігровий досвід. На рисунку 1.9 можна побачити загальний цикл RPG-прогресу в грі:

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25



Рисунок 1.9. Блок-схема загального циклу RPG-прогресу в грі

Зм.	Арк.	№ докум.	Підпис	Дата

У сучасних ігрових проєктах, особливо інді-сегменту, все частіше зустрічається підхід, коли навіть невеликі елементи дизайну мають значення. Це може бути виражено у вигляді дрібних візуальних деталей, які підсилюють загальну атмосферу гри, або у формі інтерактивності, яка створює додаткові рівні занурення гравця.

Особливу роль у таких іграх відіграє система зворотного зв'язку. Гравець повинен постійно отримувати підтвердження своїх дій: звук удару, спалах на екрані чи дрібна вібрація екрана під час зіткнення. Усе це формує відчуття присутності та контролю.

Ще одним важливим аспектом є геймдизайн, орієнтований на повторне проходження. Розробники часто додають так звану "глибину", яка є можливістю відкрити нові деталі сюжету, альтернативні шляхи або приховані елементи при наступному проходженні. Таким чином, навіть після першого завершення гри гравець має причину повертатися.

У межах розроблюваного проєкту велика увага приділяється саме варіативності досвіду гравця. Наприклад, кожне рішення в магазині має наслідки. Гравець, який зробив акцент на здоров'я, може легше пережити фінальну локацію, тоді як інший, який поклався лише на урон, знищуватиме ворогів швидше, але ризикуватиме своїм життям.

Система монет не лише служить ресурсом, а й виконує функцію ігрового мотора. Вона спонукає гравця активно взаємодіяти з оточенням, атакувати ворогів, шукати шляхи до успіху. Збір монет є не просто фоновою механікою, а структурним елементом, що веде гравця вперед.

Спробу зберегти баланс у цій RPG-системі можна розглядати як ключовий виклик. Надто щедра система винагород може швидко зробити гру надто легкою, тоді як занадто скупі нагороди можуть зменшити мотивацію продовжувати гру. Саме тому тестування і балансування є таким важливим етапом.

Різні типи ворогів не тільки додають складності, а й слугують навчальним інструментом. Вони навчають гравця новим формам небезпеки, адаптації до нових умов і змушують змінювати стратегію. Наприклад, летюча миша, яка атакує

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

дистанційно, вимагає іншого підходу, ніж вогняний пень чи гусениця.

Окремо варто відзначити візуальні ефекти. Наприклад ефекти крові, при вбивстві ворогів. Це особливо важливо для 2D-проектів, де саме графічна стилізація може компенсувати технічні обмеження.

Таким чином, гра, яка з першого погляду здається простою, насправді містить велику кількість деталей, кожна з яких працює на формування цілісного досвіду. Гравець не просто проходить рівні, а проживає гру, занурюючись у її світ через механіки, візуал і звук.

Розвиток персонажа у грі базуватиметься не на лінійній прокачці через систему досвіду або очки навичок. Він ґрунтуватиметься на вмінні гравця управляти здобутими ресурсами і правильно їх витратити для себе, щоб пройти гру. Наприклад, гравець може вибрати агресивний стиль гри та прокачувати тільки урон, або обережний – нарощувати здоров'я, або третій варіант, який є комбінованим, де гравець качає і те, і те. Це дає змогу створити персоналізовану стратегію проходження, яка відповідає індивідуальному стилю гри кожного користувача. Рішення, прийняті під час прокачки, матимуть безпосередній вплив на подальший перебіг боїв, що додає глибини тактичному плануванню.

Ще однією особливістю проєктованої RPG-системи є випадкова генерація об'єктів та ворогів на рівнях. На рівнях будуть спавнитися коробки та камені. Коробка ламатиметься після двох ударів, а камінь — після трьох. На кожному рівні кількість ворогів, їх типи та кількість об'єктів завжди буде різною. І при кожному новому проходженні гри на тому ж самому рівні, який гравець проходив раніше, все також знову буде різним. Це забезпечує унікальність кожного проходження, навіть якщо гравець обирає ті самі шляхи прокачування. Завдяки цьому система RPG не є статичною та набуває динаміки. Ніколи неможливо буде передбачити точне розміщення об'єктів, ворогів і які саме це будуть вороги. Такий підхід формує елемент реіграбельності, коли гра кожного разу відчувається по-новому. Гравцеві потрібно не тільки правильно розвивати персонажа, а й швидко адаптуватися до нових ситуацій.

Вибір прокачки в магазині буде відбуватися через простий, зручний

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

інтерфейс: гравець бачить три варіанти апгрейдів і вартість кожного. Якщо монет недостатньо, поліпшення стає недоступним. Кожне підсилення зберігається на період одного проходження, а при закінченні гри, якщо гравець захоче пройти гру знову, всі поліпшення скидаються. Це дозволяє підтримувати баланс складності, забезпечуючи рівні умови для всіх нових спроб гри. Таким чином, кожне проходження стає окремим викликом, де доводиться знову доводити свою майстерність і приймати нові стратегічні рішення. На рисунку 1.10 можна побачити блок-схему алгоритму вибору прокачки в магазині:

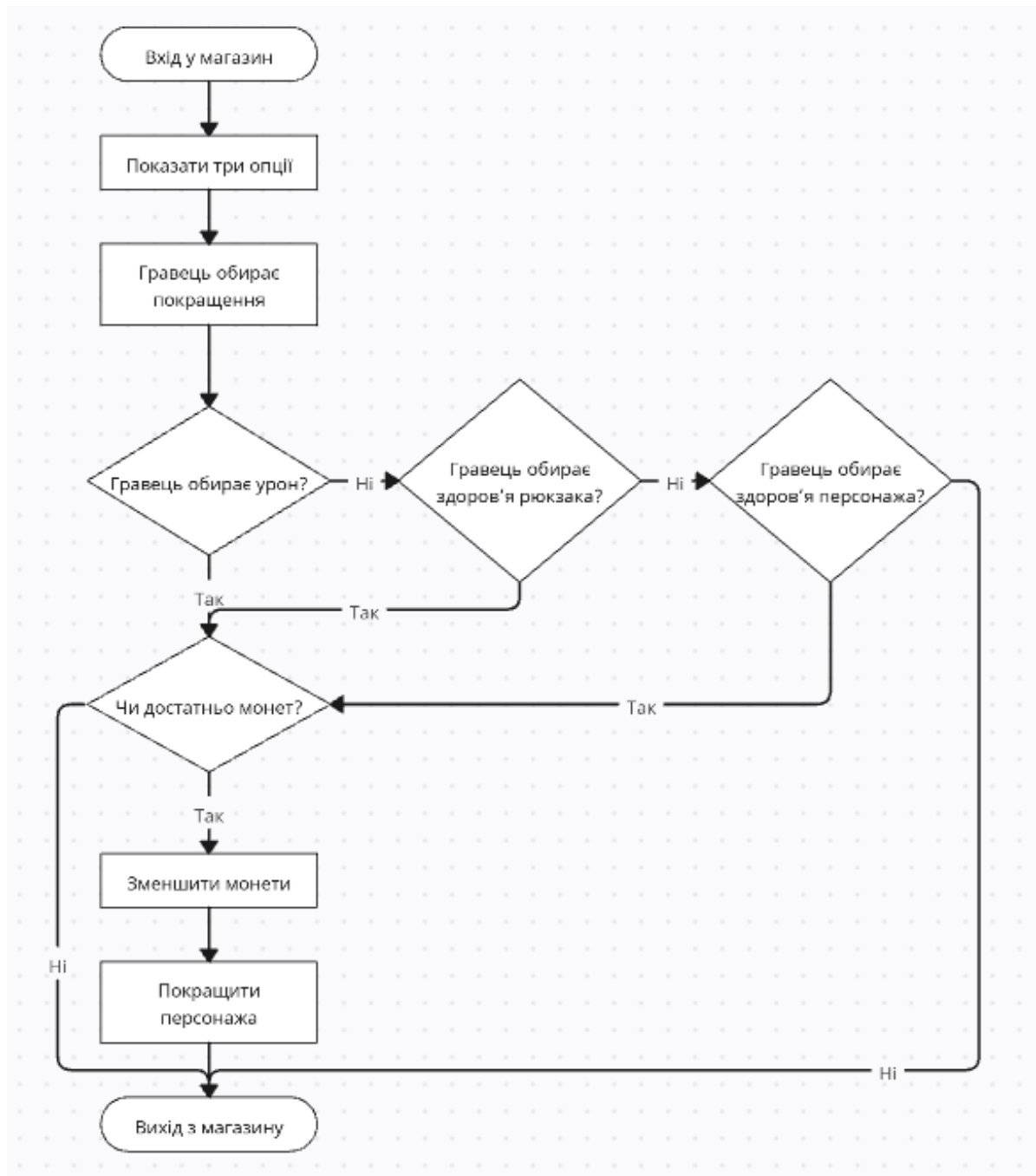


Рисунок 1.10. Блок-схема алгоритму вибору прокачки в магазині

1.6 Реалізація спроектованих елементів гри

Після всіх проектувань потрібно приступити до розробки самої гри. Для створення проєкту використовуються відповідний редактор Adobe Photoshop та програма UnityHub, в якій розробка буде проходити на Unity 2022.3.42f1. Перше, що потрібно зробити – це створити новий проєкт. На рисунку 1.11 можна побачити вікно створення проєкту у UnityHub, де також можна вибрати шаблони створення проєктів під себе:

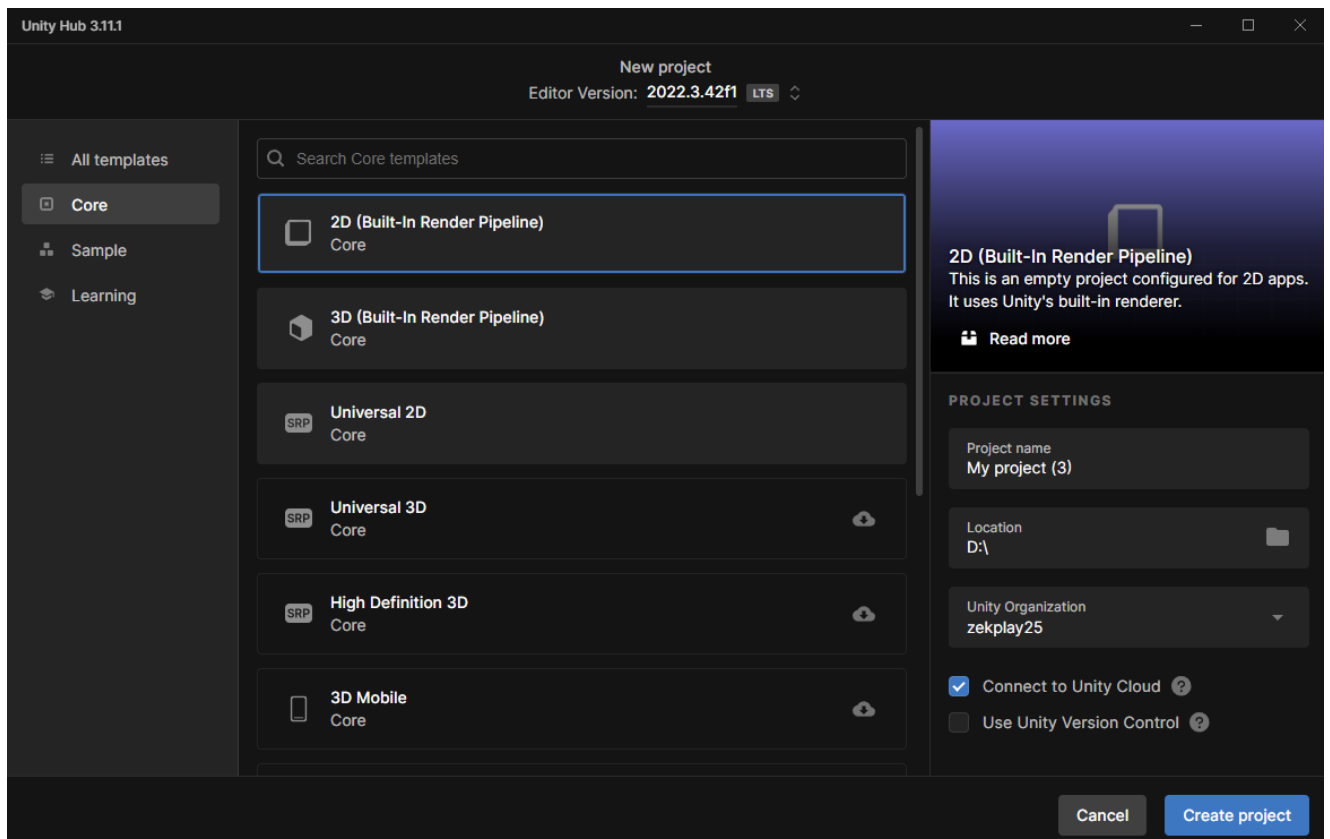


Рисунок 1.11. Вікно створення проєкту у UnityHub

Після створення нового проєкту в UnityHub буде тільки пуста сцена з одним об'єктом – камерою, через яку гравець бачитиме все, що відбувається на екрані.

Наступним кроком у розробці проєкту є розміщення об'єктів на сцені. А для того, щоб розмістити об'єкти на сцені, їх потрібно створити, а саме намалювати спрайти, тому переходимо в програму Photoshop, де відбуватиметься візуалізація всіх наступних об'єктів гри. Adobe Photoshop – це програма з багатою функціональністю, яка ідеально підходить для створення графіки завдяки

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

різноманіттю пензлів.

При створенні файлу в програмі Photoshop можна вибрати шаблони розмірів файлів, орієнтацію та кольоровий режим, що робить подальше створення зображень дуже зручним. На рисунку 1.12 можна побачити меню створення файлу в Adobe Photoshop:

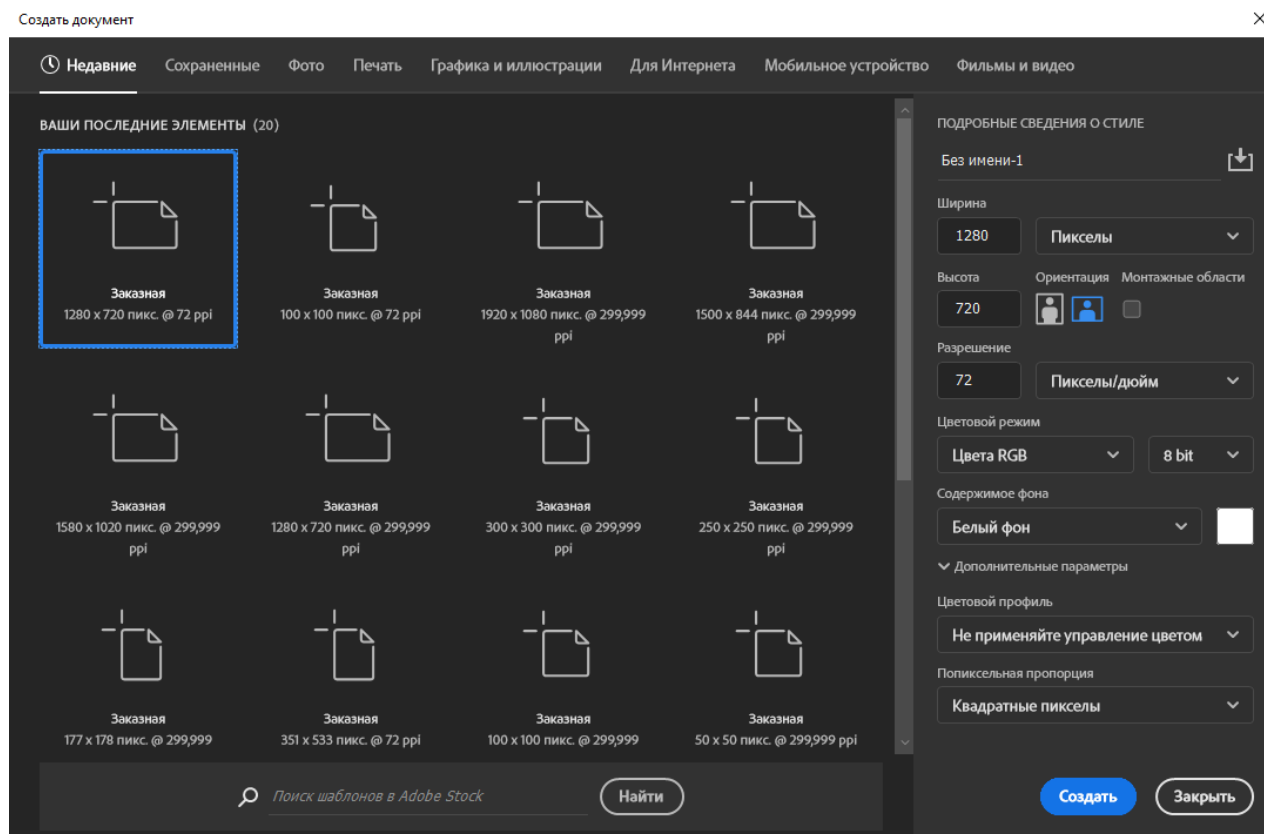


Рисунок 1.12. Меню створення файлу в Adobe Photoshop

Абсолютно всі спрайти гри будуть намальовані в цій програмі, бо вона має зручний функціонал та саме той набір інструментів, потрібний для графіки цього проекту. При створенні спрайтів гри був використаний «Вугільний пензель», який є дуже красивим та естетичним для малюнків не надто професійного художника. За допомогою цього пензля створюється відчуття, що малюнок на екрані був намальований дитячою крейдою. Також для досягнення кращої текстури був використаний інструмент «Палець» для розтушовування покрасу різноманітних об'єктів. Крім того, було підібрано яскраві кольори, щоб гра мала більш веселу стилістику та атмосферу. На рисунку 1.13 можна побачити процес малювання тренувальної локації гри:

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

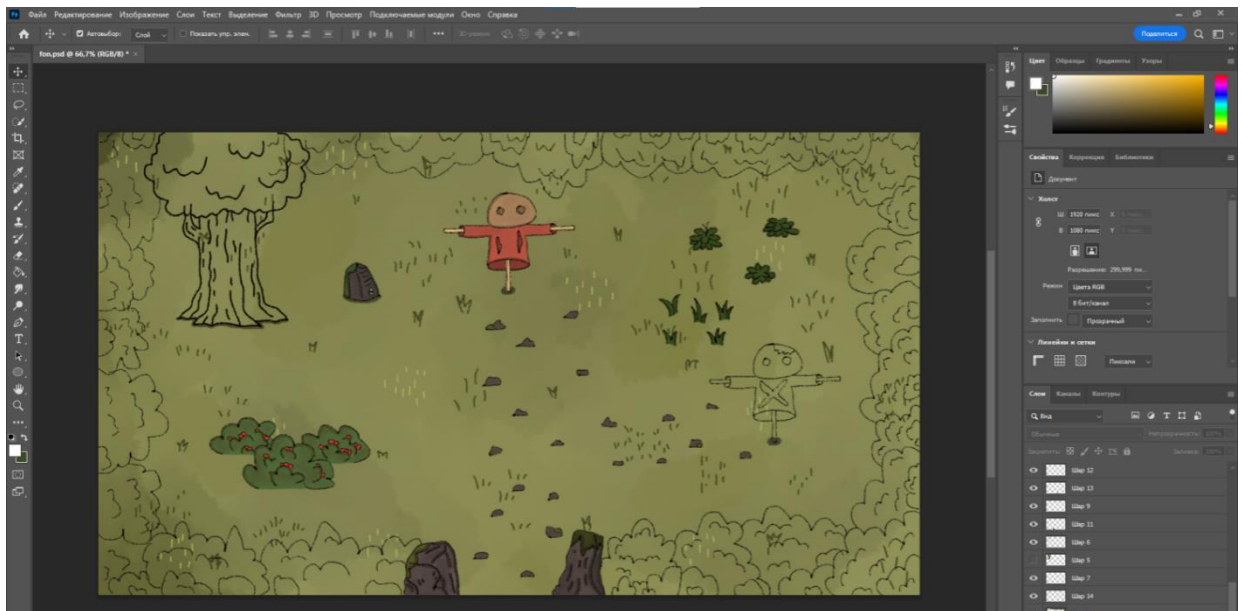


Рисунок 1.13. Процес малювання тренувальної локації гри

Також на основі початкового дизайну головного персонажа гри на рисунку 1.7 було намальовано новий спрайт дизайну персонажа, який є більш красивим та краще вписується у повноцінну графіку гри. Його можна побачити на рисунку 1.14 в інтерфейсі програми Photoshop:

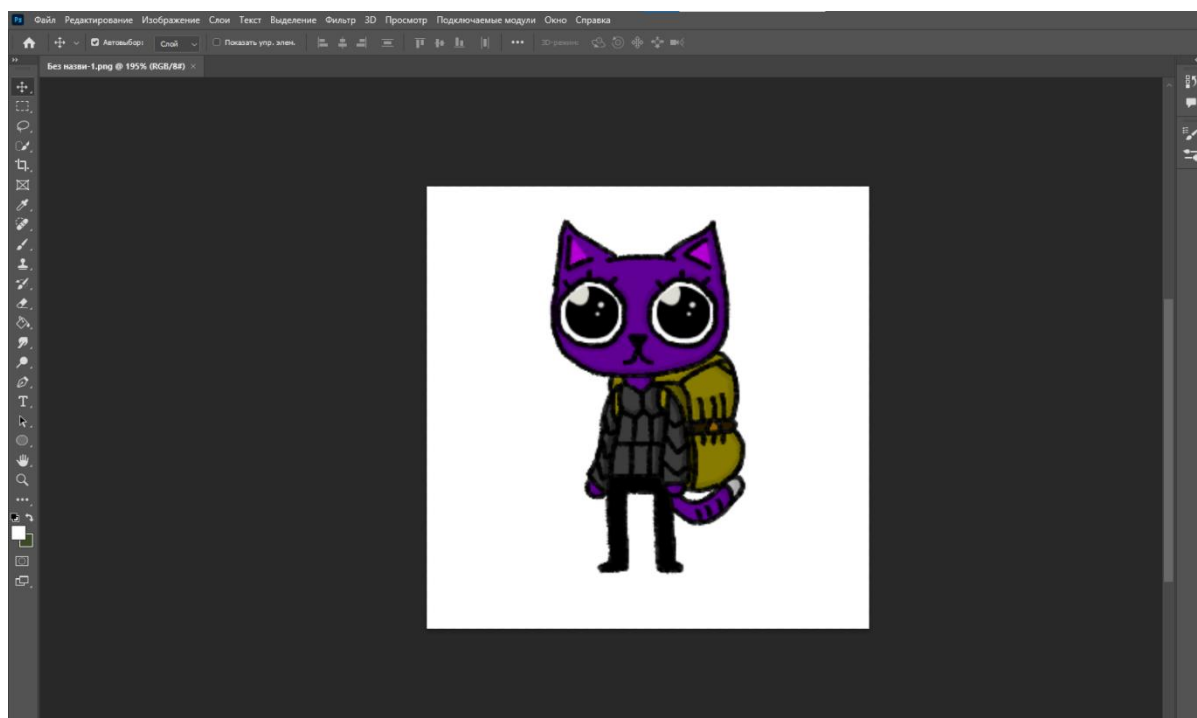


Рисунок 1.14. Новий дизайн головного персонажа гри, намальований у програмі Adobe Photoshop

Після того, як всі спрайти гри і візуальна частина були готові, потрібно

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

перейти до самої розробки. Створюємо в Unity папку Sprites та імпортуємо всі спрайти туди, а потім перетягуємо на сцену початкову локацію та персонажа. У початковій локації додаємо та налаштовуємо розміри колайдерів для того, щоб гравець не міг вийти за рамки локації під час гри. Це дозволяє чітко обмежити ігровий простір, уникнути помилок під час геймплею та зберегти логіку рівнів.

Після цього створюємо новий скрипт у тільки що створеній папці Scripts та називаємо його PlayerMovement. Скрипт PlayerMovement.cs реалізовує базову логіку руху та інші взаємодії ігрового персонажа в середовищі Unity. Він забезпечує обробку атаки, руху, ривка, отримання ушкоджень, інтерфейсу сердець гравця та рюкзака, а також збір і підрахунок монет. Цей скрипт є центральним елементом поведінки головного героя, оскільки саме через нього здійснюється більшість взаємодій між гравцем та ігровим світом.

На початку гри персонаж встановлюється у стартову позицію, яка визначається статичним класом SpawnManager. Потім ініціалізуються компоненти Rigidbody2D, AudioSource, Collider2D, що відповідають за фізику та звуки, які видаватиме персонаж, а також елементи інтерфейсу гравця, такі як серця та рахунок монет. Завдяки цим компонентам забезпечується правильне функціонування персонажа у фізичному середовищі гри, включаючи звук і візуальні ефекти.

Рух реалізується через перевірку введення з клавіатури. Якщо натиснуто клавіші руху WASD, то позиція персонажа оновлюється відповідно до напрямку. Окремо враховується орієнтація спрайта залежно від того, чи гравець іде вліво чи вправо. Код, у якому це відбувається, наведений нижче:

```
if (movement.x < 0) transform.localScale = new Vector3(1, 1, 1);  
else if (movement.x > 0) transform.localScale = new Vector3(-1, 1, 1);
```

Якщо натискається клавіша пробіл, активується короткий ривок, при якому персонаж переміщується у напрямку руху. Це дозволяє ухилитися від противників під час бою.

Атака активується натисканням лівої кнопки миші. У момент атаки на

короткий час вмикається колайдер меча. Код, у якому це відбувається, наведений нижче:

```
swordCollider.enabled = true;
```

Після того, як анімація меча програлася, колайдер вимикається.

Система отримання шкоди побудована на двох незалежних шкалах здоров'я: одна для персонажа, а інша для рюкзака. Коли ворог атакує колайдер, який знаходиться позаду персонажа на рюкзаку, віднімається синє серце, а якщо колайдер, який знаходиться спереду віднімається червоне серце. Також, якщо одна зі шкал здоров'я закінчується, запускається анімація смерті, екран темніє, вимикаються вороги, зникають усі індикатори на екрані, і через кілька секунд з'являється кнопка перезапуску.

Також у коді візуально реалізована індикація ушкоджень. При ударі від ворога персонаж відштовхується, програється анімація отримання шкоди, а потім він починає блимати і стає тимчасово невразливим. Це досягається за допомогою коротких пауз і зміни прозорості спрайта. Код, у якому це відбувається, наведений нижче:

```
spriteRenderer.color = new Color(originalColor.r, originalColor.g, originalColor.b, 0.8f);
```

Ще персонаж має можливість підбирати монети, які випадатимуть із монстрів. При зборі оновлюється лічильник у правому верхньому куті екрана. Збір монети супроводжується звуком, а сам об'єкт монети знищується. Код, у якому це відбувається, наведений нижче:

```
coinCount++;  
Destroy(other.gameObject);
```

Таким чином, скрипт управління об'єднує в собі кілька ключових ігрових механік, створюючи динамічний та зрозумілий процес управління персонажем. На рисунку 1.15 можна побачити, як візуально виглядає інтерфейс у грі:

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

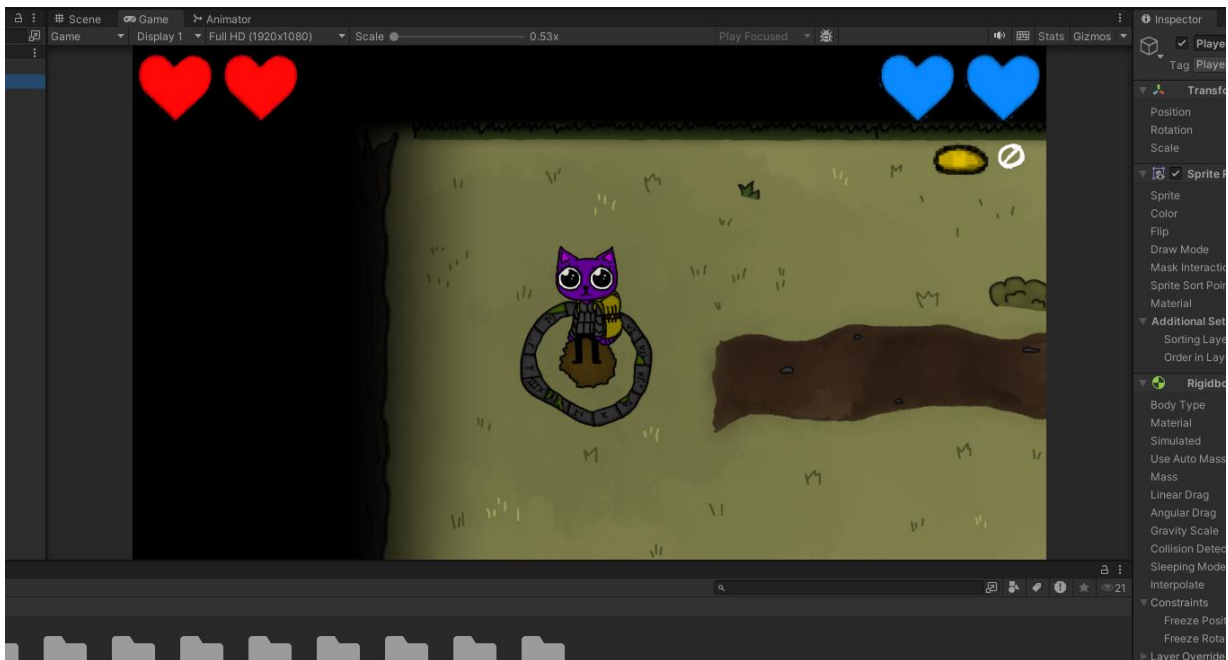


Рисунок 1.15 Візуальний інтерфейс гри

Також було реалізовано скрипт слідкування камери за персонажем. Після запуску гри камера одразу фокусується на персонажі, а надалі в методі `FixedUpdate()` камера плавно пересувається за гравцем із заданою швидкістю, використовуючи функцію `Vector3.Lerp` для згладжування руху. Завдяки цьому переміщення камери виглядає природно і не ривчасто.

Ще у гравця є меч, у якого також присутній скрипт. У цьому скрипті реалізується механіка нанесення шкоди ворогам при зіткненні меча з ними. Коли меч через `OnTriggerEnter2D` стикається з об'єктом, що має тег "Enemy", він перевіряє, чи реалізує цей об'єкт інтерфейс `IDamageable`. Якщо так, то викликається метод `TakeDamage`, передаючи кількість шкоди та напрямок удару. При цьому інтерфейс `IDamageable` дозволяє різним ворогам мати власну реалізацію прийому пошкоджень.

Пройшовши вперед по початковій локації, персонаж потрапляє на навчальну локацію через "перехід", який є просто пустим об'єктом з колайдером. При зіткненні гравця з цим об'єктом спавниться нова локація, стара видаляється, і сам персонаж переноситься на початок нової локації. Таких об'єктів всього дев'ять на сцені, і деякі з них, окрім того, що видаляють локацію, на якій в даний

момент знаходиться гравець, також видаляють об'єкти, які випадково спавняються на бойових локаціях.

Потрапляючи на навчальну локацію, гравець бачить дорогу вгору та дорогу вліво, де стоїть неігровий персонаж. Якщо підійти до нього, з'являється елемент канвасу — малюнок клавші “Е”, і при натисканні на цю клавшу викликається метод `PlayAnimation()` у скрипті `TriggerAnimationAndHide`, який програє анімацію натискання цієї клавші. На рисунку 1.16 можна побачити, як виглядає неігровий персонаж та навчальна локація:

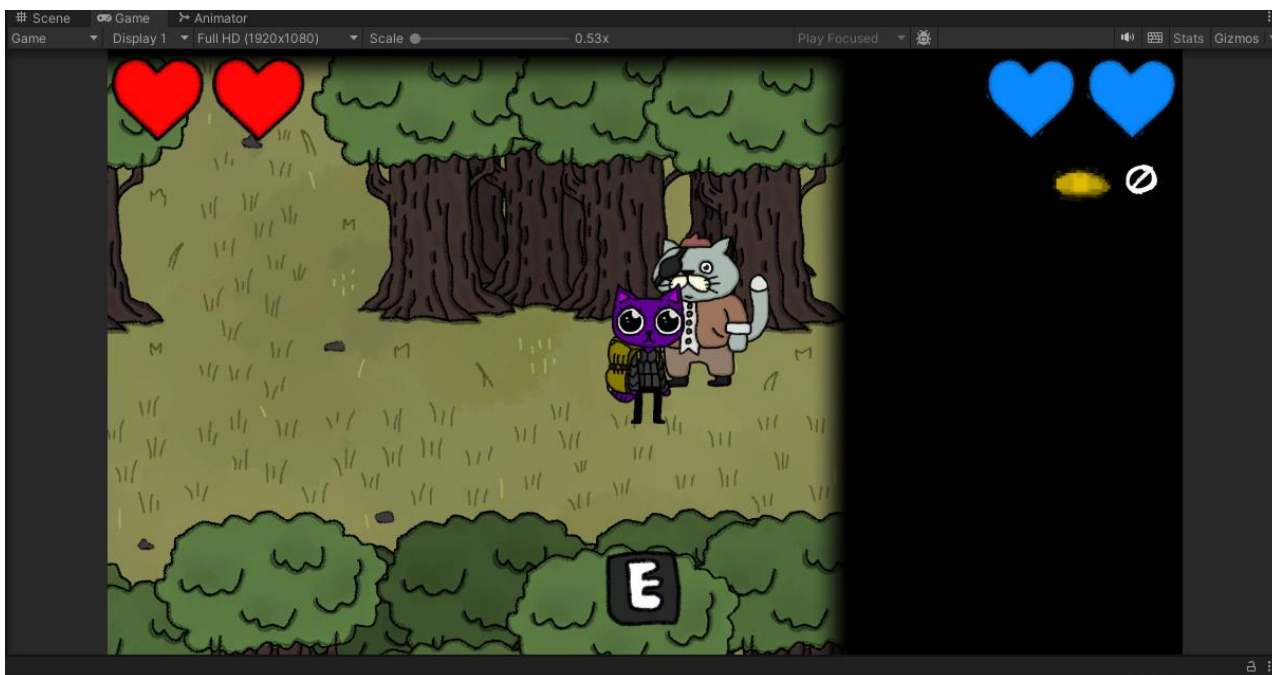


Рисунок 1.16. Неігровий персонаж та навчальна локація

Після натискання на клавшу Е поряд з неігровим персонажем починає працювати скрипт `GuiderScript`, який відповідає за взаємодію між гравцем і навчальним NPC у грі. Цей NPC виконує роль своєрідного провідника у світ гри, знайомлячи гравця з базовими механіками та загальним контекстом завдання. Метод `OnAnimationComplete()` викликається після завершення анімації значка Е. Цей метод вимикає індикатор клавші, активує два нових елементи канвасу, які включають зображення NPC з зацикленою анімацією говоріння, та запускає ефект «друкарської машинки» через метод `StartTypingEffect()` зі скрипта `TypewriterEffect`.

Завдяки цьому підходу діалог подається гравцеві поступово, створюючи ефект живої розмови, а не миттєвого виведення тексту. У скрипті `TypewriterEffect` також міститься сам текст діалогу, який буде відображатися на екрані. Крім того, під час діалогу зупиняється анімація гравця, а також відключається компонент `PlayerMovement`. Це потрібно для того, щоб уникнути випадкового руху або бойових дій у момент, коли гравець читає пояснення. Таким чином, увага гравця повністю фокусується на інформації. Текстом діалогу є в кодї:

```
private string[] messages = {  
    "  
    \"Привіт, я бачу, ти збираєшся доставити корону принцеси через цей небезпечний ліс.\",  
    \"Тож дозволь дати тобі кілька порад, щоб ти не вмерла одразу.\",  
    \"По-перше, у тебе є два види сердечок: Сині та Червоні. \",  
    \"Червоні відповідають за твоє здоров'я. Якщо вони опустяться до нуля – ти помреш.\",  
    \"Сині відповідають за твій рюкзак, який в тебе на спині зараз. \",  
    \"Якщо вони опустяться до нуля – ти втрапиш корону, і далі жити не буде сенсу.\",  
    \"Також при вбивстві монстрів з них випадатимуть золоті монетки. \",  
    \"Їх ти зможеш обміняти у мого брата-близнюка, торговця, по дорозі до замку на свої покращення.\",  
    \"Ходити можна за допомогою клавіш WASD, атакувати – ПКМ.\",  
    \"А ривок на пробіл. Але, ймовірно, ти вже і так все це знаєш.\",  
    \"Бажаю удачі у виживанні!\",  
    "  
};
```

Сам діалог можна пропускати, клікаючи лівою кнопкою миші, і після завершення діалогу всі елементи канвасу на екрані зникають, а гравець знову може рухатися. Завдяки цьому плавному переходу між діалогом і геймплеєм забезпечується безперервність ігрового досвіду без різких змін у темпі. На рисунку 1.17 можна побачити процес діалогу з NPC:

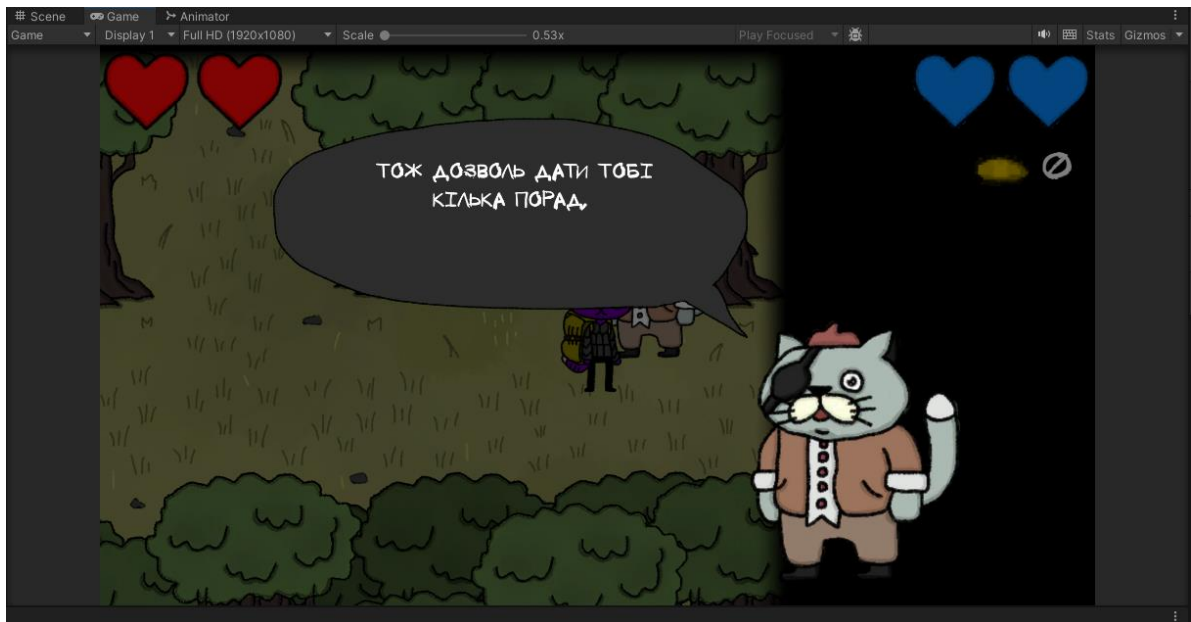


Рисунок 1.17. Процес діалогу з NPC

Якщо персонаж піде наверх, то його там буде чекати тренувальна локація, в якій є два манекени. Ці манекени мають свій скрипт EnemyDummy і при ударі мечем в них програється анімація та звук удару по ним. Також на локації є кущи, в лівому нижньому куту і якщо на до них доторкнутись, то в них також програється анімація та звук шелесту кустів. Ця локація предназначена для того щоб гравець міг розібратись в управлінні та потренуватись перед битвою. На рисунку 1.18 можна побачити як саме виглядає тренувальна локація:

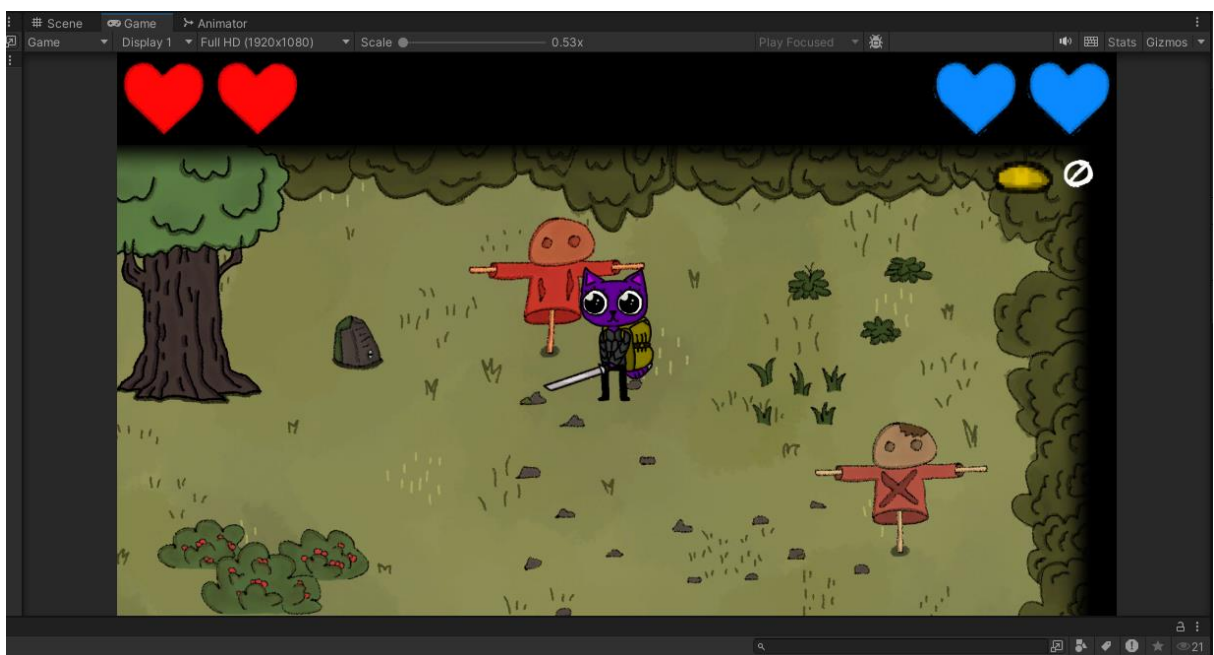


Рисунок 1.18. Тренувальна локація

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

Після того, як були розроблені всі локації, які зустрічаються на початку гри, потрібно почати розробку ворогів. Почнеться розробка з маленької гусениці. Після створення скрипта `SmallCaterpillar` перше, що треба зробити — це рух до гравця. Гусениця періодично змінює напрямок руху, обираючи нову точку біля гравця за допомогою випадкових зсувів. Код, у якому це відбувається, наведений нижче:

```
targetPosition = new Vector2(player.position.x + randomX, player.position.y + randomY);
```

Потім потрібно зробити індикатор здоров'я, який буде прихованим, поки гусениця не отримає першу шкоду від гравця. Він буде оновлюватися кожен кадр, слідкувати за гусеницею і при отриманні шкоди зменшувати візуально здоров'я гусениці. Шкоду вона буде отримувати при зіткненні з мечем, після чого буде відніматися здоров'я, програватися звук отримання урону, запускатися анімація отримання урону, а потім відштовхуватися від персонажа і отримувати тимчасову невразливість.

Коли здоров'я гусениці впаде до нуля, усі колайдери вимикаються, і коли звук її смерті повністю відтвориться, спавниться новий об'єкт, який є її мертвим тілом, та монети навколо.

Також скрипт реалізує інтерфейси `IDamageable` (отримання шкоди) та `IStoppable` (знищення при зупинці гри). Код, у якому це відбувається, наведений нижче:

```
public void StopMovement()
{
    Destroy(gameObject);
}
```

Після того, як скрипт маленької гусениці був розроблений, він також був накинутий на спрайт простої гусениці. Відмінності між цими двома ворогами полягають у тому, що у гусениці більший колайдер, більше здоров'я та вона швидша, ніж маленька гусениця.

В принципі, скрипт `SmallCaterpillar` буде використовуватися як шаблон для

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

написання скриптів подальших ворогів. Це дуже зручно, і все, що потрібно буде потім зробити – це змінити, додати або видалити деякі частини коду у створеному шаблоні.

Наступним ворогом, який буде розроблятися, є жаба, яка матиме найбільшу кількість здоров'я у грі. За розмірами вона така ж, як і гусениця, але рухається інакше. Жаба періодично виконує стрибок у напрямку до гравця. Код, у якому це відбувається, наведений нижче:

```
Vector2 playerPosition = player.position;
Vector2 direction = (playerPosition - (Vector2)transform.position).normalized;
if (direction.x > 0)
{
    transform.localScale = new Vector3(-1, 1, 1);
}
else if (direction.x < 0)
{
    transform.localScale = new Vector3(1, 1, 1);
}
float jumpDistance = 3f;
Vector2 targetPosition = (Vector2)transform.position + direction *
jumpDistance;
```

Також у грі повинен бути ворог, що стоїть на одному місці і атакує дистанційно. Цим ворогом у цьому проєкті стане вогняний пень. При спавні він залишається на місці та кожні п'ять секунд випускає навколо себе чотири вогняні сфери, які також мають тег “Enemy” і при зіткненні з персонажем нанесуть йому урон і після цього зникнуть.

Також на бойовій локації буде тег “Wall”, і по краях локації будуть колайдери. Вони виконують функцію не лише для того, щоб гравець не вийшов за межі локації, а й щоб при зіткненні з цими колайдерами вогняний шар зникав. Ця додаткова логіка допомагає уникнути хаосу на краях карти та зберігає чіткість ігрового простору.

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

Останнім монстром у цій грі стане летюча миша. Вона додає швидкість та несподіванку у бої, змушуючи гравця швидко реагувати. Її унікальність полягає в тому, що вона замість того, щоб летіти на гравця, навпаки відлітає від нього якомога далі, щоб потім почати спавнити біля себе кожні три секунди вогняний шар, який при спавні буде визначати, де в момент його спавну знаходиться гравець, а потім летіти в ту сторону, де знаходився гравець у момент спавну. Також летюча миша зможе пролітати над об'єктами, які спавнитимуться на бойовій локації. Це можна налаштувати через Layer Overrides у колайдері в інспекторі.

Розроблені вороги вирізняються своїм різноманіттям, індивідуальністю та унікальністю поведінки. Кожен з них має власний набір характеристик, анімацій, звуків і механік, що не тільки урізноманітнює ігровий процес, а й створює динамічне та захопливе середовище, яке стимулює гравця до пошуку нових стратегій у бою та підтримує інтерес на всіх етапах проходження. На рисунку 1.19 можна побачити, як саме виглядають усі вороги, присутні у проєкті:



Рисунок 1.19. Всі вороги у грі: маленька гусениця, гусениця, жаба, вогняний пінь, летюча миша

Після того, як усі вороги у грі були створені, можна приступати до розробки першої бойової локації, яка буде йти відразу після навчальної локації. Головним є реалізувати процедурну генерацію об'єктів та ворогів на локації. Процедурна генерація є сучасним підходом у геймдизайні, який дозволяє зробити кожну сесію проходження унікальною, що значно збагачує досвід гравця та підвищує реіграбельність. Цей підхід дозволяє зробити кожне проходження унікальним, що значно підвищує загальну реіграбельність гри. Оскільки вона квадратна, це буде робитися з використанням сітки. Спочатку сітку треба сформувати, і у методи

GenerateGrid() у кодї GridSpawner формується сітка точок, у яких можуть бути розміщені об'єкти. Код, у якому це відбувається, наведений нижче:

```
Vector2 cellPosition = new Vector2(startX + x * cellSize, startY + y * cellSize);
gridCells.Add(cellPosition);
availableIndices.Add(gridCells.Count - 1);
```

Потім метод SpawnObjects() випадково обирає позиції на сітці та розміщує об'єкти з масиву prefabs, що дозволяє створювати варіативні рівні, у яких об'єкти щоразу з'являються в інших місцях. Це створює ефект динамічного світу, де гравець не може передбачити наперед усі перешкоди та розташування елементів. Код, у якому це відбувається, наведений нижче:

```
GameObject prefab = prefabs[Random.Range(0, prefabs.Length)];
Instantiate(prefab, gridCells[cellIndex], Quaternion.identity);
```

Кожен прохід через бойову зону не повторюється, а постійно змінюється, що підтримує інтерес гравця протягом тривалого часу. Після цього, за заданими параметрами в інспекторі, а саме які вороги повинні з'явитися та між якими числами обирати випадкову кількість ворогів, які повинні з'явитися, вони спавняються у вільних від об'єктів клітинках, які не знаходяться надто близько до гравця. Код, у якому це відбувається, наведений нижче:

```
if (Vector2.Distance(player.position, cellPos) < minDistanceToPlayer)
{
    availableIndices.RemoveAt(randIndex);
    continue;
}
```

Це запобігає появі ворогів безпосередньо біля гравця та створює безпечну зону при старті рівня, щоб гравець не втрачав здоров'я одразу при появі на бойовій локації. Наявність такої буферної зони також забезпечує гравцю час на адаптацію до нової ситуації, що особливо важливо для збереження балансу гри. На рисунку 1.20 можна побачити, як виглядає бойова локація у грі:



Рисунок 1.20. Бойова локація у грі

Коли бойова локація була створена, потрібно перейти до створення локації магазину. Локація магазину складається з лавки, за якою стоїть той же неігровий персонаж, що і на навчальній локації. За сюжетом і діалогом з неігровим персонажем на навчальній локації можна зрозуміти, що це його брат-близнюк. Спереду, на самій лавці, лежать покращення. На об'єктах покращення знаходиться скрипт ShopObject. Якщо гравець наближається на задану відстань, об'єкт візуально виділяється: збільшується у розмірах, підсвічується «ореолом», і з'являється ціна покращення. Код, який це реалізує, наведений нижче:

```
transform.localScale = Vector3.Lerp(transform.localScale, shouldHighlight ?
enlargedScale : originalScale, Time.deltaTime * lerpSpeed);
currentColor.a = Mathf.Lerp(currentColor.a, targetAlpha, Time.deltaTime *
lerpSpeed);
uiGroup.alpha = Mathf.Lerp(uiGroup.alpha, targetAlpha, Time.deltaTime *
lerpSpeed);
```

При натисканні клавіші E, якщо гравець знаходиться поруч, спочатку перевіряється, чи вистачає монет на покупку покращення. Якщо гравцю не вистачає монет на покращення, то він чує звук відмови від неігрового персонажа-продавця від угоди. А якщо монет вистачає, то здійснюється віднімання вартості предмета від наявної у гравця кількості монет, відбувається вибране покращення,

а сам об'єкт покращення зникає зі звуком покупки. Локацію магазину у грі можна побачити на рисунку 1.21:

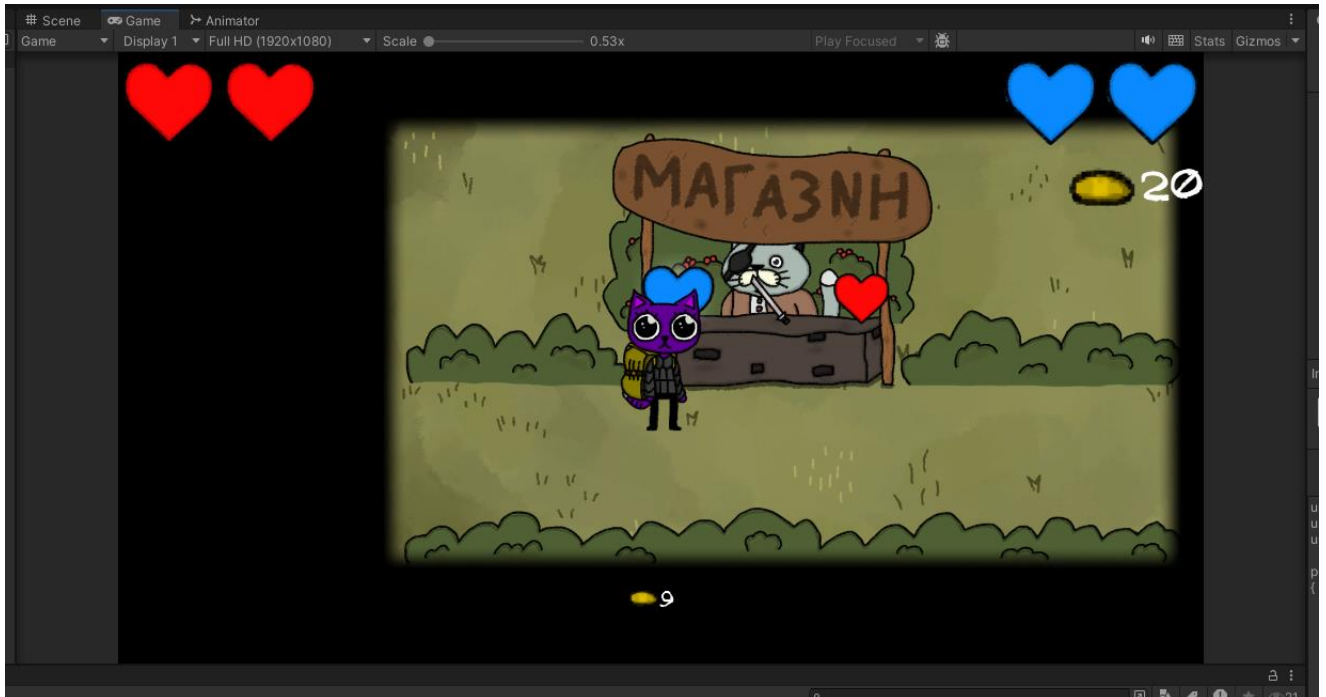


Рисунок 1.21. Локація магазину у грі

Одним із ключових елементів у розробці гри є проектування рівнів. Добре спроектовані рівні не тільки забезпечують геймплейну складність, але й утримують увагу гравця, викликаючи відчуття досягнення та бажання рухатись далі. Це досягається за допомогою балансування між напругою й відпочинком, сюрпризами й передбачуваністю.

Наприклад, після важкої бойової локації гравець потрапляє до магазину. Цей момент є психологічним розвантаженням. Подібний підхід запобігає перевтомі та підтримує мотивацію проходити гру далі. Також рандомізований спавн ворогів формує відчуття новизни при кожному проходженні, навіть якщо це той самий рівень.

Важливу роль відіграє також структура простору. Наявність вузьких проходів, відкритих арен і місць для ухилення завдяки випадковому спавну об'єктів на бойових локаціях дозволяє гравцеві реалізувати тактичні рішення, а не просто натискати кнопки. Таке проектування рівнів формує відчуття контролю та справедливості, що напряду впливає на задоволення від гри.

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

Після розробки локації магазину потрібно розставити всі локації в правильному порядку та налаштувати кожну локацію. Спочатку буде йти перша бойова локація, на якій буде спавнитися від двох до п'яти маленьких гусениць. Цей рівень виконує роль вступного випробування, дозволяючи гравцеві адаптуватися до бойової системи та відчути ритм гри. Потім на другій бойовій локації буде спавнитися від нуля до трьох маленьких гусениць та завжди буде спавнитися одна гусениця. який підхід поступово підвищує складність, не перевантажуючи гравця, але водночас тримає його у тонусі. Після цього буде йти локація магазину, де ціни спеціально зроблені такими, щоб гравець міг завжди купити хоча б одне покращення для себе. Це рішення дозволяє уникнути ситуацій, коли гравець відчуває безсилля через нестачу монет та гарантує, що кожен гравець відчує поступовий прогрес і мотивацію рухатися далі.

Далі буде йти третя бойова локація, де буде спавнитися від одного до двох вогняних пнів, від одної до двох гусениць та від нуля до одної жаби. А на четвертій бойовій локації буде з'являтися від одної до двох летючих миш, від нуля до одної гусениці, обов'язково буде з'являтися одна жаба та від нуля до трьох маленьких гусениць. Таким чином, із кожною наступною локацією гравець стикається з більш складними комбінаціями ворогів, що підтримує напругу та динаміку геймплею. Потім буде йти знову локація магазину, де гравець повинен буде вибрати покращення, щоб вступити у фінальний бій на п'ятій бойовій локації. На другій локації магазину, також як і на першій, ціни сформовані спеціально так, щоб гравець міг купити хоча б одне покращення. Це дозволяє зберігати баланс перед вирішальним етапом гри та підвищує ймовірність успішного завершення проходження.

На п'ятій бойовій локації з'являються абсолютно всі монстри гри в кількості від одного до двох, тобто в найскладнішому варіанті проходження гри їх може бути десять. Це фінальне випробування вимагає від гравця максимальної концентрації та вміння використовувати всі здобуті раніше навички й прокачування.

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

Однією з характерних рис успішних інді-ігор є висока реіграбельність, а тобто здатність гри залишатися цікавою навіть після першого проходження. Саме тому повторне проходження проєкту повинне бути не просто повторенням тих самих подій, а можливістю випробувати нову тактику, інший стиль гри або ж дослідити невідомі раніше нюанси.

Глибина повторного проходження досягається завдяки системі випадкових елементів, як-от спавн ворогів чи об'єктів, а також змінюваним можливостям прокачки. Це дозволяє створити у гравця враження, що кожен новий забіг є унікальним. Таким чином, навіть знайомі локації сприймаються по-іншому, що зберігає інтерес до гри в довготривалій перспективі.

Крім того, можливість випробувати різні стилі гри сприяє глибшому зануренню: хтось гратиме агресивно, інші обережніше. Це розширює цільову аудиторію гри та дозволяє кожному гравцеві знайти свій підхід до перемоги.

Усе це робить повторне проходження не додатковим бонусом, а повноцінною частиною ігрового досвіду, на яку варто звертати увагу ще на етапі проєктування.

Тепер, коли головна частина гри готова, можна зайнятися її кінцем. Заключний етап гри є важливим для формування загального враження і залишає гравцю приємний післясмак. На фінальній локації гравець зустрінеться з принцесою біля стіни замку, яка буде там стояти і очікувати на гравця. Якщо до неї підійти та натиснути E, то вона отримає свою корону і віддячить за це гравця. Цей момент виступає логічною та емоційною кульмінацією гри, яка надає сенс усім попереднім зусиллям гравця. Фінал – це своєрідна нагорода, яка символізує завершення шляху, наповненого викликами, боями та прийнятими рішеннями. Фінальна сцена має на меті залишити позитивне враження та завершити ігровий досвід на вдячній ноті. Саме ці останні хвилини здатні вплинути на загальне сприйняття гри та залишити приємне післясмак у свідомості гравця. На рисунку 1.22 можна побачити, як виглядає фінальна локація гри:

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

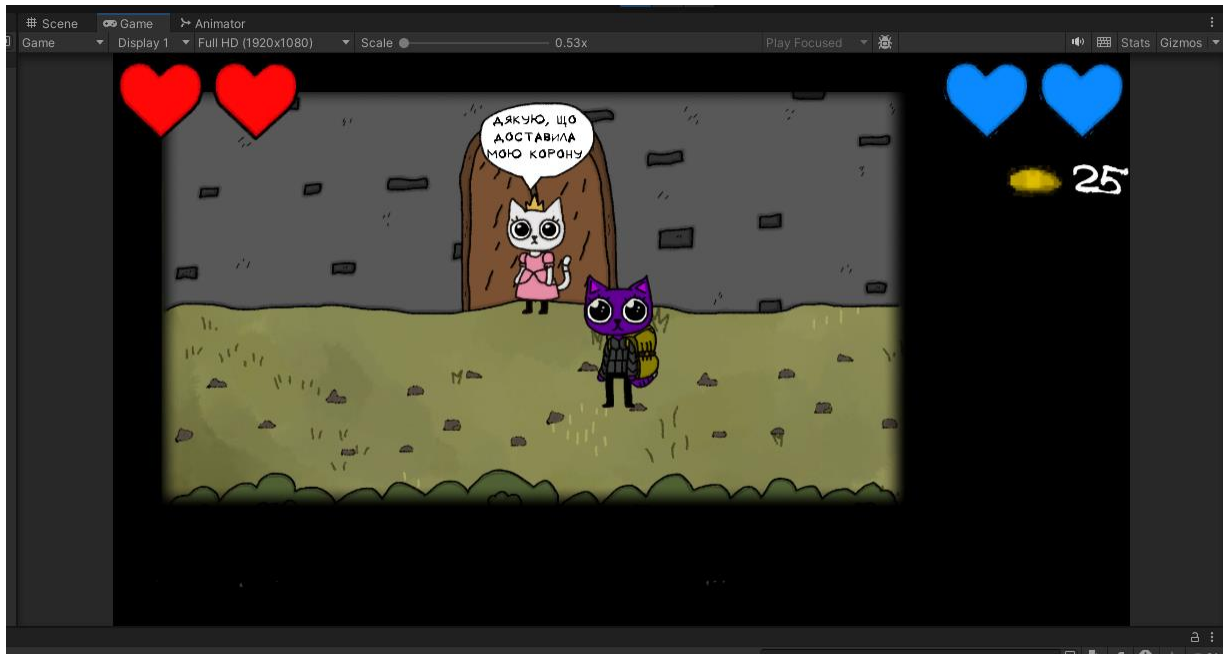


Рисунок 1.22. Фінальна локація гри

Щоб повністю пройти гру, потрібно піти далі, і коли персонаж торкається останнього переходу, всі об'єкти зі сцени видаляються, включаючи головного персонажа, і висвічується текст із подякою за гру, іменем, групою та нікнеймом автора і кнопкою “Перепройти знову”. При натисканні цієї кнопки гра буде перезапущена, і гравець знову з’явиться на початковій локації. Кінець гри можна побачити на рисунку 1.23:

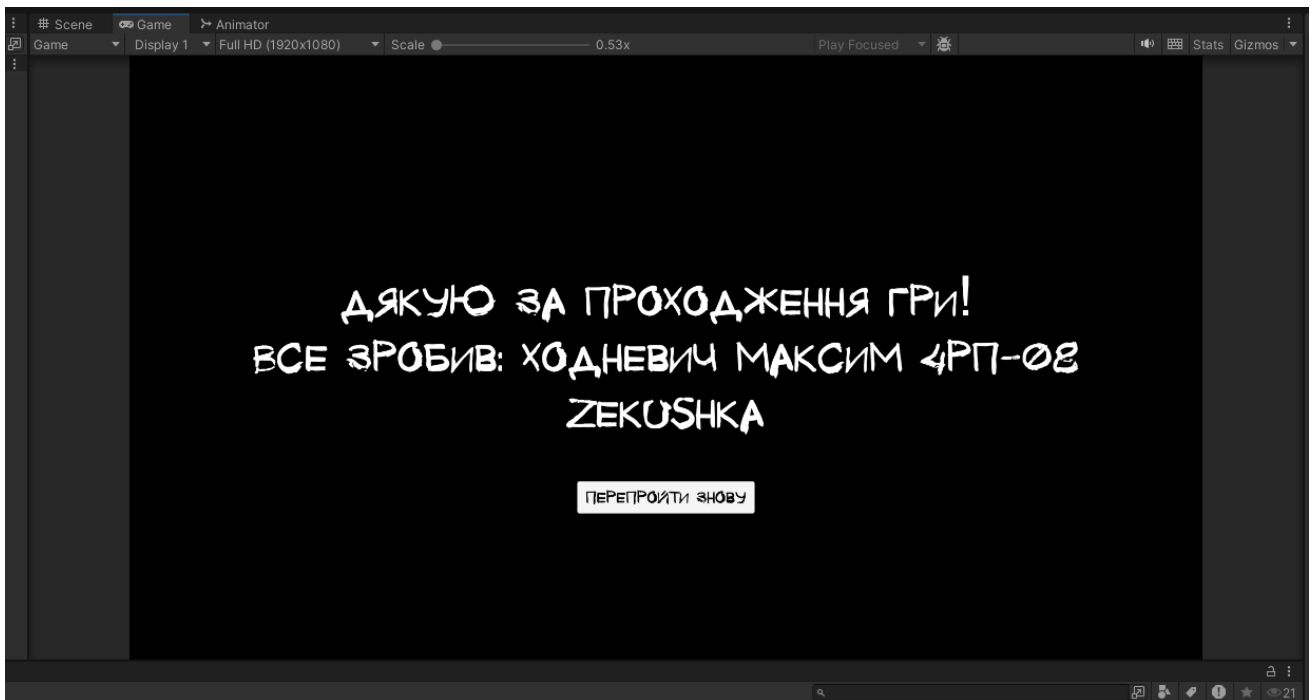


Рисунок 1.23. Кінець гри

					РП 08. 25 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

Загалом реалізовані компоненти відповідають початковій концепції гри та забезпечують стабільний ігровий процес із елементами RPG, що робить проєкт завершеним і готовим до подальшого тестування або розширення. Усі ключові механіки, включаючи бойову систему, прокачку, генерацію рівнів та інтерфейс, працюють узгоджено та без критичних помилок. Це дозволяє не лише демонструвати гру як готовий продукт, а й закласти основу для її майбутнього вдосконалення, наприклад, додавання нових рівнів, ворогів або яось розвивати сюжет.

1.7 Тестування гри

Тестування є важливим етапом розробки 2D-гри у жанрі hack and slash з елементами RPG, оскільки дозволяє виявити як візуальні, так і логічні помилки. Візуальні недоліки, як-от некоректне розміщення об'єктів чи анімаційні збої, зазвичай помітні одразу під час проходження гри. Натомість програмні помилки можуть бути непомітними, але впливати на геймплей: від некоректної роботи механік до блокування прогресу гравця. Саме тому тестування слід проводити не одноразово, а на кожному етапі розробки, поступово перевіряючи всі ключові аспекти проєкту.

У середовищі Unity тестування проводиться безпосередньо в редакторі за допомогою режиму «Play», що дозволяє моделювати потрібні ситуації на сцені. Цей режим є зручним і дає змогу оперативно вносити зміни без необхідності створення повноцінної збірки гри. Під час тестування було перевірено взаємодію гравця з ворогами і NPC, системою шкоди, підбором монет, покупкою покращень у магазині та роботою інтерфейсу. Також виконано базове балансування швидкості руху, частоти атак, коштовності покращень у магазині і складності боїв. Критичних помилок у роботі основних механік не виявлено. Однак для повної впевненості в якості гри доцільно також проводити зовнішнє тестування з участю незалежних гравців. На рисунку 1.24 можна побачити процес тестування гри:

					<i>РП 08. 25 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

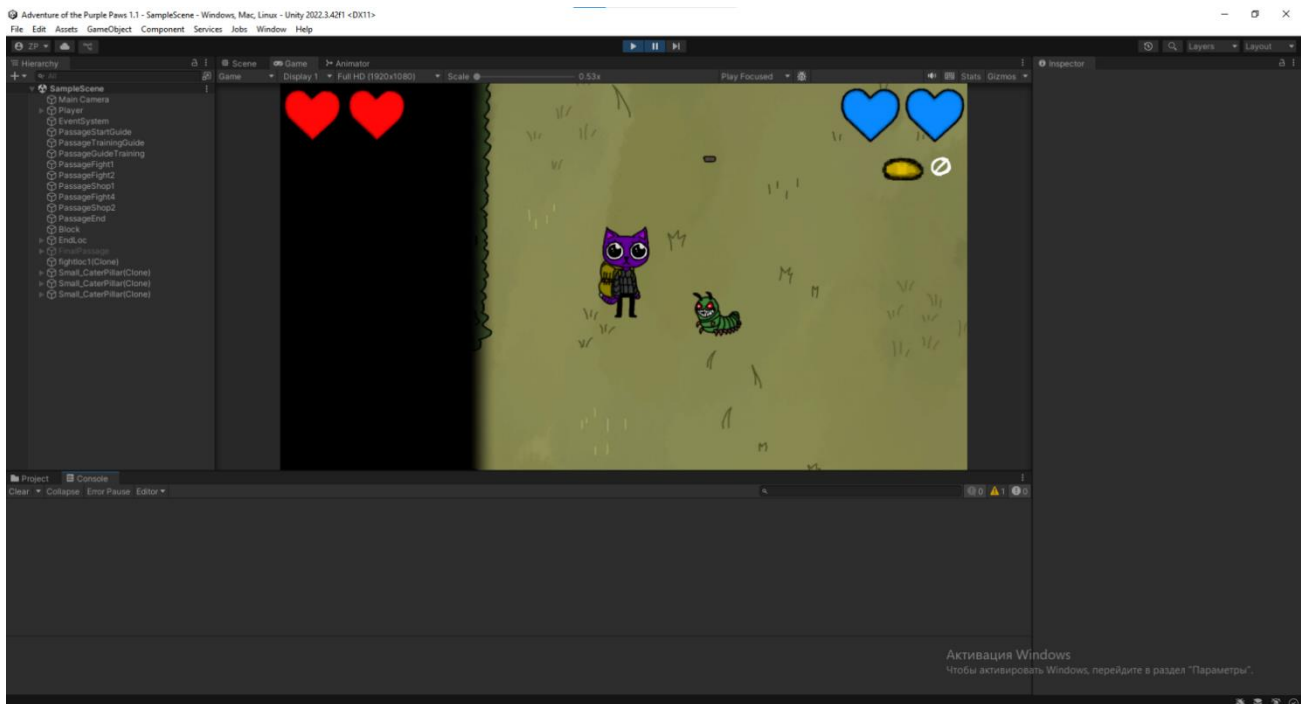


Рисунок 1.24. Процес тестування гри

У процесі тестування розробленої гри було перевірено коректність роботи всіх основних механік: руху персонажа, бойової системи, системи RPG, генерації рівнів, взаємодії з об'єктами, а також функціонування інтерфейсу. Під час тестування виявлено низку незначних помилок, таких як некоректне відображення колайдерів у певних ситуаціях та зникнення монет під час збору. Усі ці помилки були успішно усунуті шляхом внесення змін у скрипти та налаштування об'єктів у сцені.

Результати тестування показали, що гра працює стабільно, не має критичних багів і забезпечує плавний ігровий процес. Інтерфейс є інтуїтивно зрозумілим, механіки взаємодіють між собою логічно й послідовно. Окрема увага приділялась перевірці балансу системи прокачування персонажа, і після кількох ітерацій налаштування вдалося досягти оптимального співвідношення складності й винагороди. Завдяки випадковій генерації ворогів і об'єктів кожне нове проходження гри є унікальним, що підвищує реіграбельність.

Таким чином, тестування підтвердило працездатність усіх реалізованих компонентів гри та відповідність початковому задуму. Гра готова до подальшого використання, демонстрації або можливого вдосконалення у майбутньому.

					<i>РП 08. 25 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

У цьому дипломному проєкті було реалізовано та створено 2D-гру жанру Hack and Slash із включенням елементів RPG, використовуючи рушій Unity.

Ефективність програмного продукту визначається не лише якістю, а й ефективністю процесу його створення. Якість ПЗ оцінюється за такими критеріями: зручність для користувача, ефективність використання ресурсів, а також відповідність вимогам до програмного забезпечення.

Для оцінки витрат часу на створення цього програмного продукту проведемо відповідні розрахунки.

2.2 Розрахунок ціни програмного продукту нормативним методом

2.2.1 Визначення трудомісткості розробки програмного забезпечення

Тривалість створення ПЗ визначається обсягом роботи, складністю завдань, рівнем підготовки виконавців і встановленими строками, які залежать від ринкових умов.

У таблиці 2.1 наведені приклади аналогічних програм, які мають подібні функції до нашої розробки.

Для обраного варіанта, який виділений сірим кольором, визначено трудомісткість на основі відповідних норм.

Таблиця 2.1. Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_o , умов. машинних командах.
1. ПП СУБД	2500 – 9800
2. Комплексні системи ведення БД	950 – 7430
3. ПП введення інформації	1060 – 5750
4. ПП оптимізації розрахунків	1300 – 4200
5. ПП автоматизації засобів по каталогу	680 – 7000
6. ПП автоматизованих розрахунків	1300 – 8600
7. ПП загальної математики і ПП імітаційного моделювання	7800 – 8800
8. ПП організації обчислювального процесу	13000 – 10200

Вибравши аналог ПП, що містить V_0 в умовних машинних командах, трудомісткості визначати на основі табл.2.2

Таблиця 2.2. Трудомісткість

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262
4.00	283

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7\div 0,8$): $T_{ар} = 244 \times 0,8 = 195,2$ (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{ТЗ} = T^a p \times L_1 \times K_H \quad (2.1)$$

$$T_{ТП} = T^a p \times L_2 \times K_H \quad (2.2)$$

$$T_{РП} = T^a p \times L_3 \times K_H \times K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага і-го етапу розробки (див. табл. 2.2.);

K_H – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.3.);

K_T – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.4.).

Таблиця 2.3. Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L_1)	0,15	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення поправочного коефіцієнта,
що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K_n
А	Принципово нові ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПП маючий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Тому що розробка системи є ПП, що має аналоги програмних продуктів, то код ступеня новизни для мого ПП – В, а значення коефіцієнта $K_n=0,7$. По таблиці 2.5, знаючи код ступеня новизни, тепер можна визначити значення питомих коефіцієнтів трудомісткості: $L_1=0,12$; $L_2=0,11$; $L_3=0,61$;

Таблиця 2.5. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, %	Значення K_T
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

У розробленому програмному продукті використовується від 40 до 60 відсотків існуючих функцій, це значить, що $K_T=0,7$.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ}=T_a \cdot L_1 \cdot K_n = 195,2 \times 0,12 \times 0,7 = 16,39 \text{ (люд/годин)} \quad (2.4)$$

Трудомісткість розробки технічного проєкту

$$T_{ТП}=T_a \cdot L_2 \cdot K_n = 195,2 \times 0,11 \times 0,7 = 15,03 \text{ (люд/годин)} \quad (2.5)$$

Трудомісткість розробки робочого проєкту

$$T_{РП}=T_a \cdot L_3 \cdot K_n \cdot K_T = 195,2 \times 0,61 \times 0,7 \times 0,7 = 58,12 \text{ (люд/годин)} \quad (2.6)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: - технічне завдання $N_{ТЗ}=2$ (стр), - розробка ТП $N_{ТП}=28$ (стр), - розробка робочого проєкту $N_{РП}=37$ (стр), - пояснювальна записка відповідно $N_{ПЗ}=30$ (стр).

Розрахунок зведений у таблицю 2.6

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин.		
	1	2	3
1.ТЗ	$T_{ТЗ}=16,39$	$T_{КК}=0,7*N_{ТЗ}=0,7*2=1,4$	$T_{НК}=0,15*N_{ТЗ}=0,15*2=0,30$
2.Розробка ТП	$T_{ТП}=15,03$	$T_{КК}=0,7*N_{ТП}=0,7*28=19,6$	$T_{НК}=0,15*N_{ТП}=0,15*28=4,2$
3.Розробка РП	$T_{РП}=58,12$	$T_{КК}=0,7*N_{РП}=0,7*37=25,9$	$T_{НК}=0,15*N_{РП}=0,15*37=5,55$
4.Розробка ПЗ	$T_{ПЗ}=1,5**N_{ПЗ}=1,5*30=45$	$T_{КК}=0,7*N_{ТЗ}=0,7*30=21$	$T_{НК}=0,15*N_{ПЗ}=0,15*30=4,5$
Усього, в т.ч.:	216,99		
- на розробку	$\Sigma T_p=134,54$		
- контроль керівника		$\Sigma T_{КК}=67,9$	
- нормоконтроль			$\Sigma T_{НК}=14,55$

2.2.2 Розрахунок ціни програмного продукту

Щоб розрахувати загальну ціну програмного продукту, було враховано зарплати, витрати на матеріали, вартості машино-годин, а також витрати на розробку.

Погодинна ставка, обрана для цього проекту – 100 грн, оскільки виконавці заслуговують на гідну оплату праці.

Таблиця 2.7. Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудоміст. робіт, години	Погодинна тар. ставка, грн.	Розрахунок, грн.
1.Розробка ПП	134,54	100.00	13 454.00
2.Контроль керівника	67,9	100.00	6790.00
3.Нормоконт-роль	14,55	100.00	1455.00
Усього	-	-	$\Sigma Z_o=21\ 699.00$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 2.8

Таблиця 2.8. Розрахунок матеріальних витрат на розробку ПЗ

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	100	1.50	150.00
Разом	-	-	-	$B_{Mi}=150.00$
Транспортно – заготівельні Витрати (10%)				$B_{тр_з}=0,1 \times B_{M1}=0,1 \times 150=15.00$
Усього				$B_M=B_{Mi}+B_{тр_з}=165.00$

					РП 08. 25 002. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.9.

Таблиця 2.9. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	165.00	V_M (див. табл. 2.7)
2. Основна заробітна плата	21 699.00	Z_o (див. табл. 2.6)
3. Додаткова заробітна плата	2169.90	$Z_d = 0,1 \times Z_o = 21\ 699.00 \times 0,1$
4. Відрахування до єдиного фонду соціального внеску	5250.16	$V_{с.в.} = 0,22 \times (Z_o + Z_d) = 0,22 \times (21\ 699.00 + 2169.90)$
5. Накладні витрати	8 679.60	$V_{нак.} = 0,4 \times Z_o = 0,4 \times 21\ 699.00$
6. Повна собівартість	37 963.66	$C_{пов} = V_M + Z_o + Z_d + V_{с.в.} + V_{нак.} = 165.00 + 21\ 699.00 + 2\ 169.90 + 5\ 250.16 + 8\ 679.60$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$П = (C_{п} * P) / 100 \quad (2.7)$$

Де p – плановий рівень рентабельності (10-20%).

$$П = (37\ 963.66 * 10) / 100 = 3\ 796.37 \text{ грн.}$$

Договірна ціна визначається по формулі:

$$Ц_{нір} = C_{п} + П \quad (2.8)$$

$$Ц_{нір} = 37\ 963.66 + 3\ 796.37 = 41\ 760.03 \text{ грн.}$$

Звідси ціна реалізації становить:

$$Ц_p = Ц_{нір} + ПДВ \quad (2.9)$$

$$Ц_p = 41\ 760.03 + 41\ 760.03 * 0,2 = 50\ 112.04 \text{ грн.}$$

Розроблений ігровий проект є економічно обґрунтованим, а його реалізація забезпечує оптимальне співвідношення ігрової функціональності, витрат на створення та простоти впровадження для індивідуального користування або обмеженого тиражування. У процесі розробки враховувалися ключові аспекти доцільності витрат, що позитивно впливає на загальну рентабельність продукту.

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

Безпечні умови праці – це запорука здоров'я та працездатності працівників, що впливає на ефективність роботи та прибуток підприємства. Дотримання норм законодавства, стандартів і технічних вимог забезпечує належний рівень безпеки. У цьому розділі розглянуто аспекти охорони праці програміста під час розробки 2D-гри у жанрі Hack and slash з елементами RPG у Unity.

3.1 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці програмного комплексу

Фактор є небезпечним, якщо в певних умовах може спричинити травму чи різке погіршення здоров'я. Якщо ж його дія викликає зниження працездатності або професійні захворювання – це шкідливий фактор, який за тривалого впливу може стати небезпечним.

Під час роботи за комп'ютером на працівника можуть впливати такі фактори:

- Невідповідність параметрів мікроклімату нормам;
- Недостатній рівень освітленості;
- Ураження електрострумом;
- Статична електрика;
- Порушення організації робочого місця тощо.

3.2 Гігієнічні вимоги до виробничого середовища

Згідно з чинними нормами охорони праці, умови на робочому місці оператора ПК повинні сприяти високій продуктивності. Розглянемо основні вимоги.

3.2.1 Приміщення

Рекомендовано розміщувати робочі приміщення з відеодисплейними терміналами з орієнтацією вікон на північ або північний схід. На вікнах мають бути встановлені жалюзі чи щільні штори для повного затемнення. Згідно з ДБН В.2.5-28-2018, приміщення повинні бути забезпечені як природним, так і штучним освітленням.

					РП 08. 25 003. 00 ДП ПЗ	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		55

У таких просторах повинні також бути кімнати для відпочинку, психологічного розвантаження тощо.

Робоче місце кожного користувача повинне мати площу не менше ніж 6 м² і об'єм не менше 20 м³. Стіни мають бути пофарбовані матовими матеріалами, що відповідають санітарним вимогам.

3.2.2 Освітлення

Освітлення в приміщеннях із ПК має бути комбінованим – природне доповнюється штучним. Для основного освітлення рекомендовано лампи ЛД (газорозрядні). Оптимальний рівень освітленості на робочій поверхні має складати від 300 до 500 лк.

3.2.3 Рівень шуму

Для робіт, пов'язаних із концентрацією уваги, допустимий рівень шуму становить 50 дБ. Щоб зменшити вібрації та шум, обладнання слід розміщувати на спеціальні прокладки. Якщо стіни є джерелами шуму — їх варто облицьовувати звукопоглинальними матеріалами.

3.2.4 Мікроклімат

Відхилення від нормативних параметрів мікроклімату негативно впливає на швидкість реакцій і продуктивність праці, а також може підвищити кількість помилок.

У приміщеннях для розробників повинна підтримуватись температура в межах 22–25 °С, відносна вологість – 40–60 %, швидкість руху повітря – 0,1–0,2 м/с.

Для підтримки цих показників слід забезпечити ефективну вентиляцію, опалення та кондиціонування повітря, яке має проходити очищення від пилу й шкідливих домішок.

3.2.5 Електробезпека

Проходячи через організм людини електричний струм робить термічну, електролітичну і біологічну дію.

Для попередження поразок електричним струмом необхідно:

					РП 08. 25 003. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

- У повному обсязі виконувати правила провадження робіт і правил технічної експлуатації;
- Виключати можливість доступу працівника до частин устаткування, що працює під небезпечною напругою, неізольованим частинам, призначеним для роботи при малій напрузі й не підключеним до захисного заземлення;
- Застосовувати ізоляцію, що служить для захисту від поразки електричним струмом.

Для попередження поразок електричним струмом необхідно:

- У повному обсязі виконувати правила провадження робіт і правил технічної експлуатації;
- Виключати можливість доступу працівника до частин устаткування, що працює під небезпечною напругою, неізольованим частинам, призначеним для роботи при малій напрузі й не підключеним до захисного заземлення;
- Застосовувати ізоляцію, що служить для захисту від поразки електричним струмом.

Заземлені конструкції (наприклад, труби чи батареї), що знаходяться в межах робочих зон, мають бути закриті діелектричними накладками або сітками, щоб унеможливити контакт з небезпечними елементами.

3.3 Вимоги до організації робочого місця працівника

Робочі місця необхідно розташовувати таким чином, щоб у полі зору працівника не знаходилися вікна, освітлювальні прилади та поверхні з відбивальними властивостями.

Відеотермінали слід встановлювати під кутом 90–100 градусів до вікон, так щоб світло надходило збоку. Рекомендується розміщувати робочі місця з ВДТ у глибині приміщення.

Робочий стіл повинен мати регулювання по висоті в межах 680–800 мм, а його ширина має дозволяти виконувати дії в межах моторного поля. Рекомендовані параметри: висота — 725 мм, ширина — 600–1400 мм, глибина — 800–1000 мм.

					РП 08. 25 003. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

Стілець для працівника має бути обладнаний механізмом регулювання висоти сидіння, спинки і кута її нахилу, які повинні налаштовуватись незалежно, легко і надійно фіксуватися.

Екран ВДТ потрібно розміщувати так, щоб забезпечувалося зручне зорове спостереження під вертикальним кутом $+30^\circ$ до нормальної лінії зору.

Клавіатуру варто розташовувати на поверхні стола на відстані 100–300 мм від його краю, зверненого до працівника.

Організація комп'ютерного робочого місця має забезпечувати ергономічну відповідність усіх його елементів і їхнє взаємне розташування згідно з рисунком 3.1:

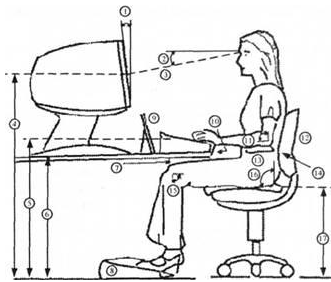


Рис.3.1. Робоче місце і робоча поза користувача комп'ютера

3.4 Пожежна безпека

Пожежна безпека приміщень, що мають електричні мережі, регламентується ГОСТ 12.1.033-81, ГОСТ 12.1.004-85. Робота оператора ЕОМ повинна вестися в приміщенні, що відповідає категорії Д пожежної безпеки.

Пожежна безпека забезпечується:

- системою запобігання пожежі;
- системою протипожежного захисту;
- організаційно-технічними заходами.

Протипожежний захист реалізується шляхом встановлення автоматичної пожежної сигналізації, оснащення приміщення засобами пожежогасіння та організації своєчасної евакуації працівників.

Для гасіння пожеж на ранніх стадіях розвитку та ліквідації незначних вогнищ за допомогою персоналу застосовуються первинні засоби пожежогасіння.

До них належать вуглекислотні й порошкові вогнегасники, інвентар для пожежогасіння (ящики з піском, бочки з водою, покривала з негорючої тканини).

Зображення засобів подано на рисунку 3.2:



Рис.3.2. Первинні засоби пожежогасіння

3.5 Організація евакуації працівників під час надзвичайних ситуацій

У разі надзвичайних ситуацій — пожежі, задимлення, витoku газу або короткого замикання важливо своєчасно евакуювати працівників. Від правильної підготовки залежить їхнє життя та здоров'я.

Для ефективної евакуації необхідно:

- Розробити й розмістити плани евакуації в усіх приміщеннях біля виходів із позначенням маршрутів, засобів пожежогасіння та аварійних виходів;
- Проводити навчальні евакуації щонайменше раз на півроку;
- Обладнати евакуаційні виходи світловими вказівниками з автономним живленням.
- Забезпечити вільний доступ до шляхів евакуації;
- Призначити відповідальних за евакуацію у кожному кабінеті чи на поверсі;
- Встановити систему оповіщення, яка спрацьовує автоматично при пожежі;
- Провести інструктаж персоналу щодо правил поведінки у надзвичайній ситуації.

ВИСНОВКИ

У результаті виконання дипломного проєкту було спроектовано та успішно реалізовано 2D-гру у жанрі hack and slash із елементами RPG на ігровому рушії Unity. Процес розробки супроводжувався ґрунтовним аналізом сучасних RPG-систем, які можна спостерігати у відомих інді-іграх, таких як Dead Cells, Moonlighter та Ender Lilies. Це дозволило виділити найважливіші особливості жанру, а також перейняти найкращі ідеї для створення власної унікальної гри.

Особливу увагу під час роботи було приділено динамічній бойовій системі, яка забезпечує живий і захоплюючий ігровий процес. Водночас реалізована інтуїтивно зрозуміла система прокачки персонажа, що робить розвиток героя гнучким і привабливим для гравця. Також була впроваджена процедурна генерація об'єктів і ворогів, що додає свіжості кожному проходженню гри та підвищує її реіграбельність.

Унікальною особливістю проєкту стала наявність двох шкал здоров'я: для персонажа й його рюкзака, що відкриває нові можливості для тактичного мислення та заглиблює геймплей, створюючи додатковий виклик для гравця.

Для реалізації гри були обрані найефективніші та найзручніші інструменти розробки: рушії Unity, мова програмування C#, середовище розробки Visual Studio для кодування, а також Adobe Photoshop для створення якісної графіки. Завдяки цьому вдалося втілити повноцінний продукт із унікальною атмосферою, механіками і дизайном.

Всі ключові системи гри, включно з бойовою механікою, взаємодією з NPC, прокачкою персонажа через ігровий магазин, були повністю реалізовані та пройшли ретельне тестування. Розроблена RPG-система дозволяє гравцям гнучко формувати свій ігровий досвід, а багаторазове проходження відкриває нові варіанти прокачки та стратегії, що значно підвищує цікавість до гри.

Проєкт підтвердив доцільність використання обраних технічних рішень та інструментів. Отримані результати можуть слугувати базою для подальшого розвитку гри або створення схожих інді-проєктів.

					<i>РП 08. 25 000. 00 ДП ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		60

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Джозеф Хокінг Unity в дії / Джозеф Хокінг., 2018. – 335 с.;
2. Джон Менінг Unity для розробників. Мобільні мультиплатформені ігри / Джон Менінг, Періс Батфілд-Еддісон., 2018. – 352 с.;
3. Алан Торн Мистецтво створення сценаріїв у Unity / Алан Торн., 2016. – 362 с.;
4. Procedural Generation in Game Design / Тарн Адамс, 2017. – 336 с.
5. Шрайбер Дж. Level Up! Посібник з розробки ігор / Джессі Шрайбер. – К.: Вид. група «Основи», 2020. – 384 с.
6. Stroustrup B. A Tour of C++ (Second Edition). – Addison-Wesley, 2018. – 240 p. – ISBN 978-0-13-499783-4.
7. Jason Gregory. Game Engine Architecture / Jason Gregory. – Third Edition. – CRC Press, 2018. – 1240 p. – ISBN978-1-138-03545-4.
8. Charles Humphrey. Learning C# by Developing Games with Unity 2021 / Packt Publishing, 2021. – 794 p. – ISBN 978-1-80020-780-6.
9. Marschner M. Mastering Unity 2D Game Development / Ashley Godbold, Simon Jackson. – Packt Publishing, 2017. – 468 p. – ISBN 978-1-78588-903-5.
10. Бондаренко О.В. Unity для початківців: Основи створення 2D-ігор / О.В. Бондаренко. – Харків: ФОП Панов А.М., 2020. – 192 с.
11. Каганюк О.К. Комп'ютерна схемотехніка: Навчальний посібник. – Луцьк: РРВ Луцького НТУ, 2016. – 236 с.
12. Бібліотека MSDN [Електронний ресурс]. – Режим доступу: URL <http://msdn.microsoft.com/ru-ru/library/default.aspx>.
13. Unity Manual [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/index.html>
14. GitHub [Електронний ресурс]. – Режим доступу: <https://github.com>
15. GameDev.net [Електронний ресурс]. – Режим доступу: <https://www.gamedev.net>

					<i>РП 08. 25 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

ДОДАТОК А. Лістинг коду основних модулів гри мовою С#

Скрипт PlayerMovement.cs

```
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
public class PlayerMovement : MonoBehaviour
{
    public int playerHP = 2;
    public GameObject[] playerHearts;
    public float moveSpeed = 5f;
    public float dashSpeed = 10f;
    public float dashCooldown = 1f;
    public Animator animator;
    public Collider2D swordCollider;
    public AudioClip walkSound, dashSound, attackSound, damageSound, deadSound, coinSound;
    public int coinCount = 0;
    public Text coinCountText;
    private Rigidbody2D rb;
    private Vector2 movement;
    private bool isAttacking = false, isDashing = false, isDead = false;
    private float lastDash;
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
        swordCollider.enabled = false;
        UpdateHearts();
    }
    void Update()
    {
        if (isDead) return;
        movement = new Vector2(Input.GetAxisRaw("Horizontal"), Input.GetAxisRaw("Vertical")).normalized;
        if (!isAttacking && !isDashing) Move();
        if (Input.GetKeyDown(KeyCode.Space) && Time.time >= lastDash + dashCooldown)
        StartCoroutine(Dash());
        if (Input.GetMouseButtonDown(0) && !isAttacking) StartCoroutine(Attack());
        animator.SetBool("isWalking", movement != Vector2.zero);
    }
    void Move()
    {
        rb.MovePosition(rb.position + movement * moveSpeed * Time.fixedDeltaTime);
        transform.localScale = new Vector3(movement.x < 0 ? 1 : -1, 1, 1);
    }
    System.Collections.IEnumerator Dash()
    {
        isDashing = true;
        lastDash = Time.time;
        AudioSource.PlayClipAtPoint(dashSound, transform.position);
        float end = Time.time + 0.2f;
        while (Time.time < end)
        {
            rb.MovePosition(rb.position + movement * dashSpeed * Time.fixedDeltaTime);
            yield return null;
        }
        isDashing = false;
    }
    System.Collections.IEnumerator Attack()
    {
```

```

        isAttacking = true;
        swordCollider.enabled = true;
        animator.SetTrigger("Attack");
        AudioSource.PlayClipAtPoint(attackSound, transform.position);
        yield return new WaitForSeconds(0.3f);
        swordCollider.enabled = false;
        isAttacking = false;
    }
    void OnCollisionEnter2D(Collision2D col)
    {
        if (col.gameObject.CompareTag("Enemy"))
        {
            playerHP--;
            UpdateHearts();
            AudioSource.PlayClipAtPoint(damageSound, transform.position);
            if (playerHP <= 0) Die();
        }
    }
    void Die()
    {
        isDead = true;
        animator.SetTrigger("Death");
        AudioSource.PlayClipAtPoint(deadSound, transform.position);
        Invoke("Restart", 2f);
    }
    void Restart() => SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Coin"))
        {
            coinCount++;
            coinCountText.text = coinCount.ToString();
            AudioSource.PlayClipAtPoint(coinSound, transform.position);
            Destroy(other.gameObject);
        }
    }
    void UpdateHearts()
    {
        for (int i = 0; i < playerHearts.Length; i++)
            playerHearts[i].SetActive(i < playerHP);
    }
}

Скрипт Sword.cs
using UnityEngine;
public class Sword : MonoBehaviour
{
    public int damage = 1;
    void OnTriggerEnter2D(Collider2D col)
    {
        if (col.CompareTag("Enemy"))
            col.GetComponent<IDamageable>()?.TakeDamage(damage, (col.transform.position
transform.position).normalized);
    }
}
public interface IDamageable
{
    void TakeDamage(int d, Vector2 dir);
}

Скрипт SmallCaterpillar.cs
using UnityEngine;

```

```

using System.Collections;
public class SmallCaterpillar : MonoBehaviour
{
    public Transform player;
    public float speed = 2.5f;
    public int maxHealth = 5;
    public GameObject coinPrefab, deathParticlePrefab, spawnAfterDeathPrefab;
    public AudioClip hurtSound, deadSound;
    private int hp;
    private bool inv, bouncing;
    private AudioSource[] audio;
    private Animator anim;
    void Start()
    {
        player = GameObject.FindWithTag("Player")?.transform;
        hp = maxHealth;
        anim = GetComponent<Animator>();
        audio = GetComponents<AudioSource>();
    }
    void Update()
    {
        if (!bouncing && player)
        {
            transform.position = Vector2.MoveTowards(transform.position, player.position, speed * Time.deltaTime);
            anim.SetBool("isRunning", true);
        }
    }
    public void TakeDamage(int dmg, Vector2 dir)
    {
        if (inv) return;
        hp -= dmg;
        audio[1].PlayOneShot(hurtSound);
        anim.SetTrigger("Hurt");
        if (hp <= 0) Die();
        else { inv = true; StartCoroutine(Bounce(dir)); StartCoroutine(ResetInv()); }
    }
    void Die()
    {
        audio[2].PlayOneShot(deadSound);
        for (int i = 0; i < Random.Range(1, 6); i++)
            Instantiate(coinPrefab, transform.position + (Vector3)Random.insideUnitCircle, Quaternion.identity);
        Instantiate(deathParticlePrefab, transform.position, Quaternion.identity);
        var obj = Instantiate(spawnAfterDeathPrefab, transform.position, Quaternion.identity);
        obj.transform.localScale = new Vector3(-transform.localScale.x, 1, 1);
        Destroy(gameObject);
    }
    IEnumerator Bounce(Vector2 dir)
    {
        bouncing = true;
        Vector2 start = transform.position, end = start + dir.normalized * 2f;
        for (float t = 0; t < 0.2f; t += Time.deltaTime)
        {
            transform.position = Vector2.Lerp(start, end, t / 0.2f);
            yield return null;
        }
        yield return new WaitForSeconds(1.5f);
        bouncing = false;
    }
    IEnumerator ResetInv() { yield return new WaitForSeconds(0.5f); inv = false; }
}

```

```

void OnTriggerEnter2D(Collider2D col)
{
    if (col.CompareTag("Sword"))
        TakeDamage(col.GetComponent<Sword>().damage, (transform.position - player.position).normalized);
}
}

```

Скрипт GridSpawner.cs

```

using UnityEngine;
using System.Collections.Generic;
public class GridSpawner : MonoBehaviour
{
    public GameObject[] prefabs;
    public Vector2 center = new(5, 5), size = new(10, 10);
    public float cellSize = 1f, minDist = 2f;
    public int minCount = 0, maxCount = 25;
    public Transform player;
    public EnemySpawnData[] enemies;
    List<Vector2> cells = new();
    List<int> free = new();
    void Start()
    {
        player = GameObject.FindGameObjectWithTag("Player")?.transform;
        for (int x = 0; x < size.x; x++)
            for (int y = 0; y < size.y; y++)
            {
                Vector2 pos = new(center.x - size.x / 2 + x * cellSize, center.y - size.y / 2 + y * cellSize);
                cells.Add(pos); free.Add(cells.Count - 1);
            }
        for (int i = 0; i < Random.Range(minCount, maxCount + 1) && free.Count > 0; i++)
            Spawn(prefabs[Random.Range(0, prefabs.Length)]);
        foreach (var e in enemies)
            for (int i = 0, c = Random.Range(e.minCount, e.maxCount + 1); i < c && free.Count > 0;)
            {
                int id = free[Random.Range(0, free.Count)];
                if (Vector2.Distance(player.position, cells[id]) < minDist) { free.Remove(id); continue; }
                var obj = Instantiate(e.prefab, cells[id], Quaternion.identity);
                var sr = obj.GetComponent<SpriteRenderer>();
                if (sr) sr.sortingOrder = Mathf.RoundToInt(-cells[id].y * 100);
                free.Remove(id); i++;
            }
    }
    void Spawn(GameObject prefab)
    {
        int id = free[Random.Range(0, free.Count)];
        Instantiate(prefab, cells[id], Quaternion.identity);
        free.Remove(id);
    }
}
[System.Serializable]
public class EnemySpawnData
{
    public GameObject prefab;
    public int minCount, maxCount;
}

```

Скрипт ShopObject.cs

```

using UnityEngine;
using UnityEngine.UI;
using System.Collections;
public class ShopObject : MonoBehaviour
{

```

```

public float dist = 5f, lerp = 5f;
public Vector3 scaleBig = new(1.5f, 1.5f, 1.5f);
public Renderer halo;
public CanvasGroup ui;
public Text coinText;
public int price = 5;
Transform player;
PlayerMovement pm;
Sword sword;
Vector3 scaleOrig;
Color haloColor;
public AudioSource buySound, denySound;
void Start()
{
    player = GameObject.FindGameObjectWithTag("Player").transform;
    pm = player.GetComponent<PlayerMovement>();
    sword = GameObject.FindGameObjectWithTag("Sword").GetComponent<Sword>();
    coinText = GameObject.FindGameObjectWithTag("CoinCount").GetComponent<Text>();
    scaleOrig = transform.localScale;
    haloColor = halo.material.color;
    halo.material.color = new(haloColor.r, haloColor.g, haloColor.b, 0);
    ui.alpha = 0;
}
void Update()
{
    float d = Vector3.Distance(player.position, transform.position);
    bool near = d <= dist;
    float t = Time.deltaTime * lerp;
    transform.localScale = Vector3.Lerp(transform.localScale, near ? scaleBig : scaleOrig, t);
    var c = halo.material.color;
    c.a = Mathf.Lerp(c.a, near ? 1 : 0, t);
    halo.material.color = c;
    ui.alpha = Mathf.Lerp(ui.alpha, near ? 1 : 0, t);
    if (near && Input.GetKeyDown(KeyCode.E)) Buy();
}
void Buy()
{
    if (pm.coinCount < price) { denySound.Play(); return; }
    pm.coinCount -= price;
    coinText.text = pm.coinCount.ToString();
    sword.damage++;
    StartCoroutine(Effect());
}

IEnumerator Effect()
{
    foreach (var sr in GetComponentsInChildren<SpriteRenderer>()) sr.enabled = false;
    halo.enabled = false;
    Destroy(ui.gameObject);
    buySound.Play();
    yield return new WaitForSeconds(buySound.clip.length);
    Destroy(halo.gameObject);
    Destroy(gameObject);
}
}

```

Розробка 2D-гри у жанрі Hack and slash з елементами RPG на програмному рушії Unity

Дипломний проект студента групи 4РП-08: Ходневича Максима Олеговича
Керівник: Джабраїлов Д. В.

Аналіз існуючих RPG-систем



RPG-системи дозволяють розвивати персонажа різними способами. В цьому проєкті реалізовано через магазин і монети.

Обрання засобів розробки



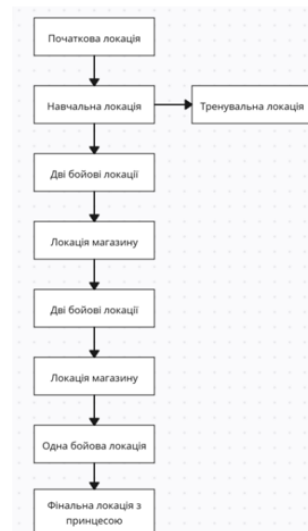
Unity і Photoshop були використані для створення гри та ручної графіки у стилі вугільного малюнка.

Концепція ігрового процесу



Гравець керує лицарем-кішкою, проходить рівні з випадковими боями та прокачкою в магазині.

Основні елементи гри



Реалізація рівнів та генерації довкілля



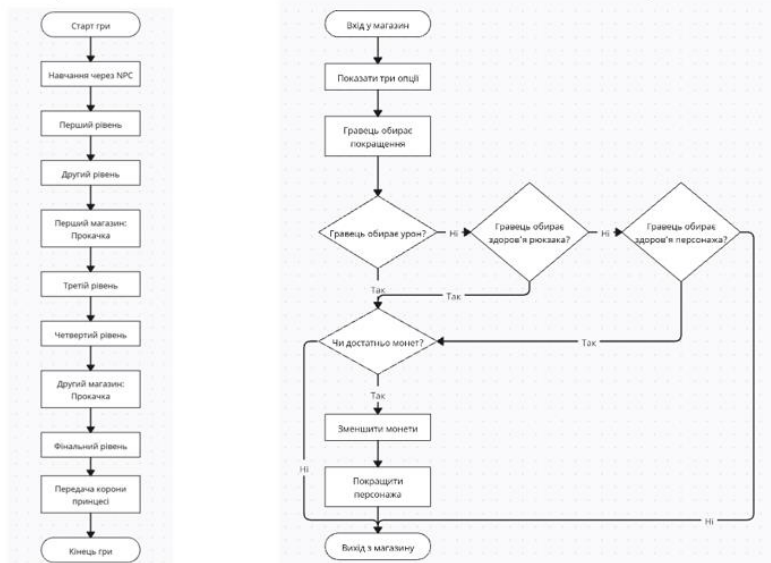
Гра має послідовні рівні з генерацією об'єктів і ворогів, що забезпечує унікальність і зростання складності.

Реалізація ворогів

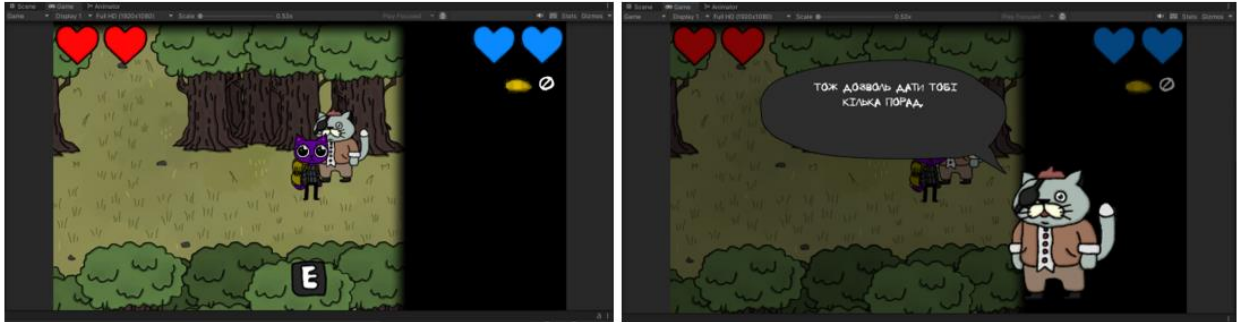


У грі п'ять ворогів з унікальною поведінкою, анімаціями, звуками, індикаторами здоров'я та дропом монет.

Проектування RPG-системи гри



Реалізація RPG-системи



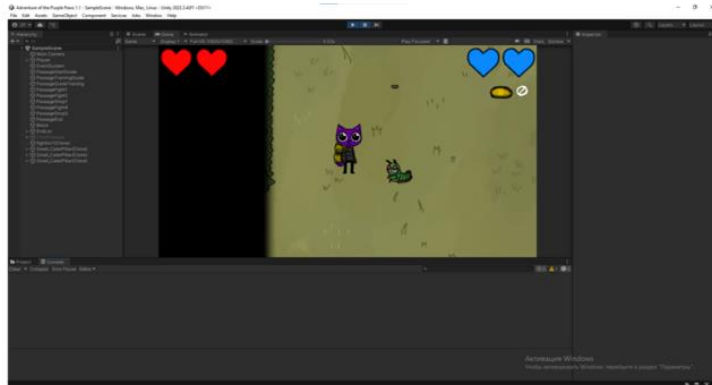
У грі NPC-наставник старий кіт з одним оком, що через діалог пояснює сюжет, управління та RPG-систему.

Реалізація RPG-системи



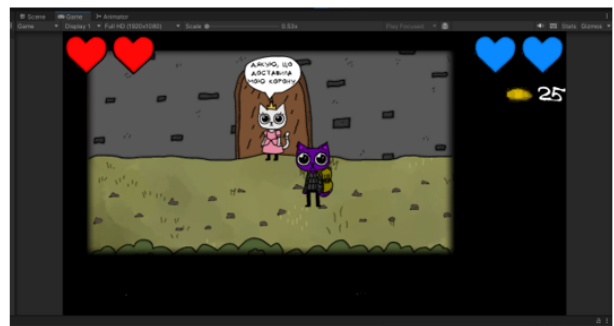
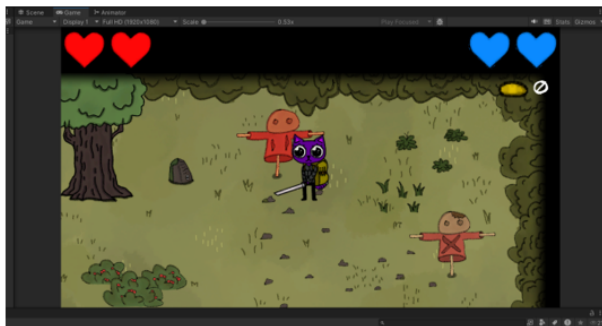
RPG-система через магазини після кожних двох рівнів дає гравцю миттєві покращення за монети.

Тестування гри



Під час тестування перевіряли всі механіки, баланс RPG-системи, стабільність генерації та коректність переходів.

Скріншоти розробленого проекту



РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Ходневича Максима Олеговича

(прізвище, ім'я та по батькові)

Спеціальність 121 "Інженерія програмного забезпечення"

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Джабраїлов Дмитро Володимирович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка 2D-гри у жанрі Hack and slash з елементами RPG

Обсяг розрахунково-пояснювальної записки 72 сторінок

Обсяг графічної (презентаційної) частини 12 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню
Представлений на рецензію дипломний проект повністю відповідає меті проектування та технічному завданню. Тематика дипломного проекту є актуальною для своєї галузі та присвячена питанням створення ігрових продуктів в цілому та розробці ігор на ігровому програмному рушії Unity зокрема.

б) характеристика виконання кожного розділу дипломного проекту (роботи)
Дипломний проект складається зі вступу, трьох розділів, висновків, переліку використаних джерел. У основному розділі розглянуті питання проблематики розробки гри на ігровому програмному рушії Unity, сформовано концепцію гри згідно до теми дипломного проекту та завданню, виконано проектування основних аспектів розробляємої гри. За допомогою відповідного програмного забезпечення реалізовані всі намічені роботи з ігровим процесом.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи)
Графічна частина виконана на достатньо високому рівні у вигляді презентації із використанням офісного пакету Microsoft PowerPoint та Visio. Пояснювальна записка виконана акуратно та у відповідності до норм оформлення документів із використанням офісного пакету Microsoft Word. Загальна якість виконання документації – добра, академічного плагіату у роботі не виявлено

г) перелік позитивних якостей дипломного проекту (роботи) _____

1. Грамотно обґрунтовано вибір рушія та жанру гри;

2. Використання професійних інструментів;

3. Детально описано етапи розробки гри.

д) основні недоліки дипломного проекту (роботи) _____

Складність розширення — частина функціоналу реалізована без уніфікації, що ускладнює модифікацію. У документації не достатньо висвітлено питання оптимізації продуктивності чи ресурсоефективності. Це може мати значення для гри, яка базується на процедурній генерації та численних об'єктах, що вимагають розумної роботи пам'яті та процесора.

Оцінка розрахункової частини Добре

Оцінка графічної частини Добре

Загальна оцінка Добре

Прізвище, ім'я, по батькові рецензента к.т.н. Шибасва Наталя Олегівна

Місце роботи і посада рецензента Національний університет «Одеська політехніка»,
доцент кафедри інформаційних технологій



[Handwritten signature]

червня 2025 р.

ВІДГУК

керівника на дипломний проєкт здобувача (здобувачки) освіти
відділення комп'ютерних систем

Ходневича Максима Олеговича

(прізвище, ім'я та по батькові)

Спеціальність: 123 "Комп'ютерна інженерія"

Освітньо-професійна програма: «Розробка програмного забезпечення»

Тема дипломного проєкту: Розробка 2D-гри у жанрі Hack and slash з елементами RPG

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЄКТУ

- а) обсяг і якість виконання проєкту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проєкт виконано відповідно технічному завданню. Пояснювальна записка містить 72 сторінки. У пояснювальній записці виконано опис етапів розробки 2D-гри у жанрі Hack and slash з елементами RPG, а також її програмного забезпечення. Графічна частина складається з 12 слайдів мультимедійної презентації, які також містять креслення, передбачені технічним завданням. Якість виконання пояснювальної записки добре, якість графічної частини відмінна, розробку виконано в повному обсязі.
- б) самостійність роботи над проєктом: Протягом всього строку дипломного проєктування та переддипломної практики здобувач освіти Ходневич Максим Олегович поступово та послідовно виконував всі етапи розробки. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника
- в) теоретична підготовка випускника (випускниці): Здобувач освіти Ходневич Максим Олегович під час роботи над дипломним проєктом вивчив достатню кількість літературних джерел та матеріалів за даною тематикою. Вважаю, що теоретична підготовка дипломника добра і він готовий до захисту дипломного проєкту

г) вміння розв'язувати виробничі та конструкторські питання _____
Під час дипломного проектування здобувач освіти Ходневич Максим Олександрович продемонстрував вміння ефективно розв'язувати питання пов'язані з проектуванням та реалізацією, що виникали в процесі створення комп'ютерної гри в жанрі Hack and slash з елементами RPG.

Оцінка розрахункової частини Добре
Оцінка графічної частини Відмінно
Загальна оцінка Добре

Прізвище, ім'я, по батькові керівника дипломного проекту _____
Джабраїлов Дмитро Володимирович

Місце роботи і посада керівника дипломного проекту _____
*ВСП "Одеський технічний фаховий коледж ОНУ", викладач
спецдисциплін комісії комп'ютерних технологій та програмної інженерії*

Підпис 

« 16 » 06 2025 р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Ходневич М.О.

здобувач освіти гр. 4РП-08, та

Джабраїлов Д.В.,

керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка 2D-гри у жанрі Hack and slash з елементами RPG» (автор роботи – Ходневич М.О., керівник роботи – Джабраїлов Д.В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

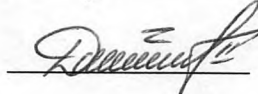
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Ходневич М.О. /

Керівник



/ Джабраїлов Д.В. /

«16» червня 2025 р.

Д О В І Д К А

циклової комісії КТ та ПІ
про допуск до захисту дипломного проєкту
здобувача (здобувачки) освіти ІV курсу
відділення комп'ютерних систем групи 4РП-08

Ходневича Максима Олеговича

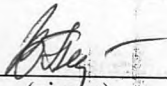
на тему Розробка 2D-гри у жанрі Hack and slash
з елементами RPG

Висновок відповідальної особи за проведення нормоконтролю:

пояснювальна записка до дипломного проєкту виконана з деякими

порушеннями ДСТУ та оформлена відповідно до вимог Положення про

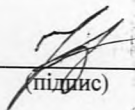
дипломне проєктування


(підпис)

16.06.2025
(дата)

Петрашова В.І.
(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного плагіату згідно звіту про перевірку від 14.06.2025 р. значення коефіцієнту подібності в роботі становить 12,42%, коефіцієнт цитування – 1,75%.


(підпис)

16.06.2025
(дата)

Краснокутська К.Г.
(П.І.Б.)

Попередня експертиза (малий захист) дипломного проєкту

здобувача (здобувачки) освіти

Ходневича М.О.
(П.І.Б.)

проведена « 16 » червня 2025 р.

Висновки Пояснювальна записка до дипломного проєкту виконана у повному
обсязі. Випускна кваліфікаційна робота (дипломний проєкт) відповідає
вимогам Положення про дипломне проєктування та рекомендована до
захисту.

Голова ЦК КТ та ПІ


(підпис)

Кривченко Ю.В.
(П.І.Б.)

Звіт подібності

метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Розробка 2D-гри у жанрі Hack and slash з елементами RPG

Автор

Науковий керівник / Експерт

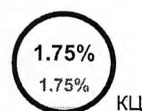
Ходневич Максим ОлеговичДжабраїлов Дмитро Володимирович

підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

12129

Кількість слів

90605

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		16
Інтервали		0
Мікропробіли		15
Білі знаки		0
Парафрази (SmartMarks)		88

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	Колір тексту
		КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	146 1.20 %
2	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	135 1.11 %
3	https://card-file.ontu.edu.ua/server/api/core/bitstreams/c5cd348b-fc64-4a25-9a5b-6cc8d62db909/content	114 0.94 %
4	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	85 0.70 %
5	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	72 0.59 %

6	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	71 0.59 %
7	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	59 0.49 %
8	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	45 0.37 %
9	https://card-file.ontu.edu.ua/server/api/core/bitstreams/c5cd348b-fc64-4a25-9a5b-6cc8d62db909/content	40 0.33 %
10	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	37 0.31 %

з домашньої бази даних (0.25 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Розробка 3D-гри у жанрі survival-horror з налаштуваннями рівнів складності 6/12/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	23 (3) 0.19 %
2	Розробка web-застосунку для генерації повідомлень із використанням технологій штучного інтелекту 6/14/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	7 (1) 0.06 %

з програми обміну базами даних (0.55 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Куліковський_Денис_Дипломна_робота_ІПЗ_4_0_1 (6) 5/29/2025 State University of Intellectual Technologies and Communications (Кафедра інженерії програмного забезпечення)	32 (4) 0.26 %
2	Дослідження особливостей технологій програмної реалізації екшн гри Castle transylvania 10/28/2024 University of Customs and Finance course papers (University of Customs and Finance course papers)	21 (3) 0.17 %
3	Григоренко_Д.В._12201_КВР 6/15/2024 Odessa National Polytechnic University (Новокаховський політехнічний фаховий коледж)	7 (1) 0.06 %
4	ПРОЦЕС СТВОРЕННЯ ТА РЕАЛІЗАЦІЯ ПОВЕДІНКОВИХ МОДЕЛЕЙ У ПЕРСОНАЖІВ У 2D-ПЛАТФОРМЕРІ 6/11/2025 International University of Economics and Humanities named after Academician Stepan Demianchuk (International University of Economics and Humanities named after Academician Stepan Demianchuk)	7 (1) 0.06 %

з Інтернету (11.62 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	791 (17) 6.52 %
2	https://card-file.ontu.edu.ua/server/api/core/bitstreams/c5cd348b-fc64-4a25-9a5b-6cc8d62db909/content	249 (11) 2.05 %
3	https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download	84 (4) 0.69 %
4	https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download	77 (3) 0.63 %

5	https://card-file.ontu.edu.ua/bitstreams/e4afae26-0a7e-4a4d-afc2-94341838de2a/download	48 (3) 0.40 %
6	https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download	33 (1) 0.27 %
7	https://pastebin.com/AfDsyLB6	24 (3) 0.20 %
8	https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download	22 (2) 0.18 %
9	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	20 (1) 0.16 %
10	https://card-file.ontu.edu.ua/bitstreams/5240e379-7721-49f0-8ee8-27140b0b473a/download	15 (1) 0.12 %
11	https://stroustrup.com/books.html	13 (2) 0.11 %
12	https://card-file.ontu.edu.ua/server/api/core/bitstreams/4bb7255e-46d4-4349-9726-9698476da02d/content	10 (1) 0.08 %
13	https://studopedia.org/9-59191.html	7 (1) 0.06 %
14	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a05c07c5-bf65-4cb0-bdfa-e28694707551/content	6 (1) 0.05 %
15	https://card-file.ontu.edu.ua/bitstreams/bbaf3f38-16a8-4070-bead-5562769b7c71/download	5 (1) 0.04 %
16	https://thelib.info/ekologiya/2269519-vimogi-do-primishhen-obladnannya-ta-organizacii-robochogo-miscya/	5 (1) 0.04 %

Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-------	---------------------------------------

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма: «Розробка програмного забезпечення»
Група: 4РП-08

Дипломний проєкт
здобувача освіти денної форми навчання
РП.08.25.000.ДП

ХОДНЕВИЧА
МАКСИМА
ОЛЕГОВИЧА

м. Одеса
2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма: «Розробка програмного забезпечення»
Група: 4РП-08

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломного проєкту на тему:

Проектний матеріал складається з пояснювальної записки на _____ сторінках та графічного (презентаційного) матеріалу на _____ аркушах (слайдах)

Дипломник _____ (Ходневич М.О.)