

Міністерство освіти і науки України
Одеський національний технологічний університет
Кафедра комп'ютерної інженерії



**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему Розробка комп'ютерної 3D квест-гри
(назва кваліфікаційної роботи згідно наказу ОНТУ)

Здобувача Лебедева А.К.
(прізвище, ініціали)

4 курсу 542a групи

Керівники: Асистент Колумба І.В.
(посада, прізвище та ініціали)

к.т.н., доц. Шестопалов С.В.
(посада, прізвище та ініціали)

Консультанти: _____
(посада, прізвище та ініціали)

Phd, ст.викл. Богданов О.О.
(посада, прізвище та ініціали)

Кваліфікаційна робота допускається до захисту

Рішення кафедри від 05.06 2024 р., протокол № 8

Завідувач кафедри комп. інженерії Сергій АРТЕМЕНКО
(назва кафедри) (підпис) (Ім'я ПРІЗВИЩЕ)

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерної інженерії, програмування та кіберзахисту
Кафедра комп'ютерної інженерії
Ступінь вищої освіти бакалавр
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма Розробка ігор та інтерактивних медіа у віртуальній
реальності

ЗАТВЕРДЖУЮ

Зав. кафедри комп'ютерної інженерії
Сергій АРТЕМЕНКО
« _____ » _____ 2024 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Лебедева Андрія Костянтиновича

1. Тема роботи Розробка комп'ютерної 3D квест-гри

Затверджена наказом університету від « 30 » серпня 2023 р., № наказ 442-03

2. Термін задачі здобувачем закінченої роботи 28 травня 2024 р.

3. Вихідні дані роботи

1. Середовище розробки Visual Studio. 2. Ігровий двигун Unity.

3. Мова програмування C#.

4. Перелік питань, які потрібно розробити

1. Вступ. 2. Аналіз та постановка задачі. 3. Проектування комп'ютерної гри.

4. Розробка демонстраційної версії гри. 5. Економічна частина.

6. Охорона праці. 7. Загальні висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Економіка</i>	<i>Phd, ст. викл. Богданов О.О.</i>		
<i>Охорона праці</i>	<i>ст.викл. Слушина Н.В.</i>		
<i>Нормоконтроль</i>	<i>асистент Колумба І.В.</i>		

7. Дата видачі завдання _____

Керівники _____ *Ірина КОЛУМБА*

_____ *Сергій ШЕСТОПАЛОВ*

Завдання прийняв до виконання _____ *Андрій ЛЕБЕДЕВ*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	<i>Дослідження предметної області.</i>	<i>20.02.24</i>	
2.	<i>Огляд існуючих аналогів.</i>	<i>01.03.24</i>	
3.	<i>Постановка завдання.</i>	<i>07.03.24</i>	
4.	<i>Створення концепт-документу.</i>	<i>14.03.24</i>	
5.	<i>Проектування.</i>	<i>28.03.24</i>	
6.	<i>Вибір програмного засобу.</i>	<i>11.04.24</i>	
7.	<i>Розробка комп'ютерної гри.</i>	<i>20.04.24</i>	
8.	<i>Тестування.</i>	<i>18.05.24</i>	
9.	<i>Економічні розрахунки.</i>	<i>27.05.24</i>	
10.	<i>Оформлення пояснювальної записки.</i>	<i>2.06.24</i>	
11.	<i>Підготовка графічного матеріалу.</i>	<i>3.06.24</i>	

Здобувач-дипломник _____ *Андрій ЛЕБЕДЕВ*

Керівники роботи _____ *Ірина КОЛУМБА*

_____ *Сергій ШЕСТОПАЛОВ*

Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.

Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.

Здобувач-дипломник _____ *Андрій ЛЕБЕДЕВ*

АНОТАЦІЯ

Дипломна робота присвячена розробці тривимірної комп'ютерної гри з елементами квесту. У грі створені різноманітні завдання та квести для гравця за допомогою застосування спеціального ігрового рушію.

В першому розділі було проаналізовано предметну область, виявлені відмінності між двовимірними та тривимірними іграми, оглянуто існуючі аналоги створюваної гри.

В другому розділі було створено концепт-документ гри, розроблено моделі персонажів та оточення, були створені анімації та ігрові рівні.

У третьому розділі було вибрано програмний засіб та розроблено гру за допомогою нього.

Четвертий розділ розраховує усі витрати на створення проекту та виявляє його економічну ефективність.

У п'ятому розділі розглянуто питання охорони праці під час роботи з комп'ютером.

Результатом роботи є тривимірна квест-гра, створена за допомогою інструменту *Unity*.

Ключові слова: комп'ютерні ігри, квест, кросплатформовий ігровий рушій *Unity*.

ABSTRACT

This thesis is devoted to the development of a three-dimensional game with quest elements. There are a variety of tasks and quests made in the game by using a special game engine.

The first section analyzes the subject area, detects the differences between two-dimensional and three-dimensional games, examines the existing analogues of the created game.

In the second section, a concept document of the game was created, models of characters and surroundings were developed, animations and game levels were created.

The third section includes software selection and game development using this software.

The fourth section has calculations of all the costs of creating a project and its economic efficiency.

The fifth section describes the issue of labor protection while working with a computer.

The result is a three-dimensional adventure game created using Unity.

Keywords: *computer games, quest, cross-platform game engine Unity*

ЗМІСТ

	стор.
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ.....	11
1.1 Аналіз предметної області.....	11
1.2 Відмінності між 2D та 3D квестами	13
1.3 Процес створення квесту	16
1.4 Огляд існуючих аналогів	19
Висновок до першого розділу	24
РОЗДІЛ 2 ПРОЕКТУВАННЯ КОМП'ЮТЕРНОЇ ГРИ.....	26
2.1 Створення концепт-документу.....	26
2.2 Опис гри	29
2.2.1 Інвентар.....	29
2.2.2 Крафт.....	30
2.2.3 Використання предметів	30
2.2.4 Руйнування об'єктів	31
2.2.5 Покрокова збірка предметів.....	31
2.2.6 Ігрове оточення	32
2.3 Текстурування та анімація моделей	35
2.4 Запис та обробка звукових ефектів.....	37
2.5 Проектування рівнів гри	38
Висновок до другого розділу.....	42
РОЗДІЛ 3 РОЗРОБКА ГРИ.....	44
3.1 Вибір та обґрунтування програмного засобу	44
3.2 Створення основних кімнат.....	47
3.3 Розміщення моделей, ігрових об'єктів та освітлення	50
3.4 Написання основних скриптів.....	54

					КРБ.КІ.1.442-03.1.11			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка комп'ютерної 3D квест-гри	Літ.	Арк.	Акрушіє
Розробив		Андрій ЛЕБЕДЕВ						
Перевірів		Ірина КОЛУМБА					6	
Рецензент		Олександр ЛАТЕНКО				ар. 542, ОНТУ		
Нормоконтроль		Ірина КОЛУМБА						
Затвердив		Сергій АРТЕМЕНКО						

3.5 Додавання аудіоефектів	57
3.6 Створення інтерфейсу користувача та системи управління грою	59
3.7 Тестування гри	62
Висновок до третього розділу	67
РОЗДІЛ 4 ЕКОНОМІЧНА ЧАСТИНА	68
4.1 Організаційно-економічне обґрунтування роботи	68
4.1.1 Актуальність та результати проекту	68
4.1.2 Огляд аналогів і конкурентів	68
4.1.3 Порівняльний аналіз	73
4.2 Визначення мети і результатів роботи	75
4.3 Розрахунки ціни програмного продукту.....	77
4.4 Бізнес план стартапу проекту.....	87
Висновок до четвертого розділу	88
РОЗДІЛ 5 ОХОРОНА ПРАЦІ.....	89
5.1 Основні положення охорони праці	89
5.2 Правила охорони праці при роботі з комп'ютером	91
Висновок до п'ятого розділу	95
ЗАГАЛЬНІ ВИСНОВКИ.....	96
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	97
ДОДАТКИ.....	99
Додаток А Скрипти для гри.....	99
Додаток Б Графічний матеріал.....	109

ВСТУП

Комп'ютерні ігри – один з типів відеоігор, що створюються спеціально для персональних комп'ютерів. Одними з основних переваг гри на комп'ютері являється більш різноманітне та направлене на користувача апаратне забезпечення та програмне забезпечення. Також під важливими характеристиками зазначають значну потужність, графіку та обробку відео та аудіо даних.

Відеоігри можуть бути двовимірними та тривимірними. Тривимірні ігри зазвичай складніше та ставлять перед гравцем не тільки задачу пройти гру, а ще й дослідити місцевість, вирішити додаткові задачі. У той же час двовимірні ігри відрізняються використанням плоскої графіки та пересуванням гравця вліво і вправо, вниз та вгору. Більшість таких ігор лінійні, а це означає, що основна ціль гравця – просто пройти гру з початку до кінця. 2D ігри мають менше можливих рухів та взаємодій з іншими об'єктами. У багатьох 2D іграх об'єкти являють собою «спрайти» – двовимірні зображення. Також у таких іграх сильно спрощена камера, вона не має такої перспективи, як у 3D іграх.

Історія розвитку обчислювальної техніки тісно зв'язана з розвитком комп'ютерних ігор. Вони беруть свій початок у 50-ті роки двадцятого століття. Першою відеогрою вважаються «хрестики-нулики», що були перенесені розробником А. С. Дугласом з паперу на екран комп'ютера у 1952 році. Така гра отримала назву «ОХО» та являла собою програмну реалізацію відомої усім гри. Також, однією з перших таких ігор була комп'ютерна гра «Теніс для двох», створена фізиком Вільямом Гігінботемом. В неї можна було грати удвох. Гра дійсно стала дуже важливою для історії, тому що була першою електронною грою з графічним дисплеєм.

Подальші кроки у створенні комп'ютерних ігор здійснювалися разом зі створенням нових технологій та розвитком персональних комп'ютерів. Наприклад, створення такого пристрою, як комп'ютерна миша несе за собою нові ідеї та підвищення інтересу до відеоігор у населення.

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дат		

Поступово з'являються та розвиваються нові жанри комп'ютерних ігор, наприклад, *Duke Nukem 3D* та *Quake* – перші повністю тривимірні ігри жанру шутер від першої особи.

Світ комп'ютерних ігор включає до себе багато різних жанрів, за якими класифікують ігри. Така класифікація в першу чергу допомагає гравцям орієнтуватися під час вибору нової гри. Одним з основних жанрів стає квест. Такий жанр представляє собою інтерактивну історію, в якій є головний герой, керуючи яким треба вирішувати різноманітні задачі. Гравець обдумує поставлені задачі, шукає підказки та сховані деталі. В такій грі гравець повинен вирішити загадку, не знаючи методів її рішення, у цьому йому допомагають підказки інших персонажів або об'єктів. В комп'ютерних іграх жанру квест відсутні або зведені до мінімуму характерні для інших жанрів елементи, такі як: бої, економічне планування, та задачі, які потребують від гравця швидких дій у відповідь. Ігри жанру квест також розділяють на декілька видів: текстові квести, графічні, квести-головоломки, візуальні новели та симулятори ходьби.

Першими іграми жанру квест були *Colossal Cave Adventure*, *Hunt the Wumpus* та *Zork*. Вони були текстовими та використовували інтерпретатори тексту, що вводив гравець. Такі відеоігри не були дуже зручними в управлінні, але з розвитком комп'ютерної техніки все змінилось, що допомогло квестам мати все більш зручніший інтерфейс, цікавіший геймплей та графіку.

Також виділяють окремий жанр, що об'єднує в собі характерні ознаки квестів і жанру *action*, та називають його *action-adventure*. Ігри жанру *action-adventure* зазвичай характеризуються великою кількістю персонажів, діалогами, можливістю дослідження рівнів, різноманітними задачами для гравця, динамічними сутичками з ворогами та складним сюжетом.

Розвиток комп'ютерних ігор з самого початку їх створення і до сьогодні указує на те, як швидко вони стали частиною сучасної культури та вплинули на нас.

					КРБ.КІ.1.442-03.1.11	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дат		

Зазначимо мету, об'єкт, предмет та задачі, які необхідно вирішити в ході даної роботи.

Метою роботи є розробка 3D комп'ютерної гри з елементами.

Об'єктом дослідження є процеси розробки 3D комп'ютерної гри з елементами.

Предметом дослідження є методи та процеси розробки 3D комп'ютерної гри з елементами.

Основними задачами, які необхідно вирішити в ході роботи є:

1. Аналіз предметної області.
2. Створення концепт – документу.
3. Розробка гри.
4. Тестування гри.

Методи розробки (дослідження). Для досягнення поставленої мети використовуються методи статистичного аналізу ігор жанру квесту. В методиці проектування та розробки використовуються методи теорії алгоритмів, кінцевих автоматів та інформатики в цілому.

Наукова новизна одержаних результатів. Запропоновано методи створення системи збору предметів для головного предмету. А також системи, які реалізують основні механіки ігор жанру квесту.

Практичне значення одержаних результатів. Розроблено скрипти для реалізації механік ігор жанру квест. Розроблено демонстраційну версію гри жанру квест гри з використанням запропонованих методів та скриптів. Скрипти та демонстраційна версія можуть бути використані в навчальному процесі здобувачів освіти університету.

					КРБ.КІ.1.442-03.1.11	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 1

АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Робота над поставленим завданням починається з аналізу предметної області. Мета завдання полягає у створенні тривимірної комп'ютерної гри з елементами квесту. Тому розглянемо основні відомості та властивості.

На сьогоднішній день існує безліч комп'ютерних ігор різноманітних жанрів, вони швидко розвиваються та вражають своїми можливостями. Такі ігри дають можливість освоювати і розвивати навички спілкування, соціалізації, розширювати кругозір. Кожен може вибрати для себе самий підходящий варіант або жанр.

Існує багато жанрів ігор, жанри виділяють на основі спільних характеристик та мети ігрового процесу. Також жанри можуть поєднуватися у одній грі.

Одними з основних жанрів ігор можна виділити:

- пригодницькі ігри;
- екшн;
- гоночні;
- рольові;
- стратегічні;
- симулятори;
- навчальні;
- спортивні.

Зараз більшість людей вважає комп'ютерні ігри розвагою, способом заспокоєння та проведення часу. Але перші ігри слугували скоріше для наукових цілей, а не розваги. Тому історія розвитку обчислювальної техніки тісно зв'язана з розвитком комп'ютерних ігор.

					КРБ.КІ.1.442-03.1.11	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дат		

Вони беруть свій початок у 50-ті роки двадцятого століття. Першою відеогрою вважаються «хрестики-нулики», що були перенесені розробником А. С. Дугласом з паперу на екран комп'ютера у 1952 році. Така гра отримала назву «ОХО» та являла собою програмну реалізацію відомої усім гри. Також, однією з перших таких ігор була комп'ютерна гра «Теніс для двох», створена фізиком Вільямом Гігінботемом. В неї можна було грати удвох. Гра дійсно стала дуже важливою для історії, тому що була першою електронною грою з графічним дисплеєм.

Одними з перших розробників відеоігор була група студентів з Массачусетського технологічного інституту. Студенти створили гру «*Spacewar!*» для нової ЕОМ та надали можливість управління космічними кораблями за допомогою контролерів.

Spacewar! стала дуже популярна у всьому світі через те, що була вбудована в комп'ютери у виді програми, за допомогою якої перевірялась їх працездатність.

Однак, перші ігри жанру квест з'явилися лише у 1970-х роках, коли Вільям Краудер розробив програму під назвою *Colossal Cave Adventure*. Інтерфейс гри був текстовим, гравець застосовував прості текстові команди. Гра швидко набувала популярності та розповсюджувалась.

Ігри з елементами квесту можна описати як інтерактивну історію, де гравець керує головним персонажем та виконує різноманітні завдання для досягнення мети. Геймплей полягає у розв'язуванні головоломок, логічних та динамічних, що вимагає від гравця мати здібність логічно мислити та швидко приймати рішення. В такій грі також присутні багато додаткових задач, а характерні для інших жанрів елементи, такі як: бої, економічне планування та інші, відсутні або зведені до мінімуму.

Квест ігри зазвичай розділяють на декілька видів, такі як текстові пригоди, графічні квести, квести-головоломки, візуальні новели та симулятори ходьби.

					КРБ.КІ.1.442-03.1.11	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дат		

Перші пригодницькі ігри були виключно текстового змісту, графіка в них не використовувалась через технічні обмеження комп'ютерів. В цих іграх основні задачі гравця включали в себе встановлення діалогу з ігровим світом, дослідження словникового запасу програми, вирішення логічних загадок та задач.

В перших іграх-квестах графічні ілюстрації були декоративними, частіше графіка була векторною, тому що це не потребувало великих витрат пам'яті. Гравець керував діями персонажа, вводячи послідовності команд на клавіатурі, а зображення було статичним та слугувало для стимулювання уяви гравця.

Тривимірні квести мали лише обмежений успіх. Багато прихильників класичних квестів, таких як *Day of the Tentacle* та *Space Quest*, не змогли пристосуватися до таких змін, а інші люди почали віддавати перевагу більш реалістичним іграм та жанру *RPG*, що починав набирати популярність. Проблемою також стала відмова від зручного та звичного для усіх режиму *point-and-click*, тепер гравець керував персонажем за допомогою стрілок клавіатури.

1.2 Відмінності між 2D та 3D квестами

Оглядаючи процеси розробки ігор, можна зрозуміти основні відмінності між створенням двовимірних та тривимірних ігор. 2D або двовимірна графіка працює із зображенням, що сформоване у двох вимірах – висоті та ширині. А об'єктом тривимірної графіки є зображення, що сформовано у трьох вимірах – висоті, ширині та глибині.

Таким чином, 3D дозволяє оцінювати розмір об'єкту та відстань до нього. На відміну від 2D, тривимірні ігри мають більш глибоку графіку, яка дозволяє персонажу переміщатися вільно та без багатьох обмежень.

Загалом, створення 3D-гри представляє собою складний процес, її розробка потребує набагато більше часу та фінансових витрат, ніж під час

					КРБ.КІ.1.442-03.1.11	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дат		

створення аналогічної, але двовимірної гри. Організація даних, засобів для проектування та рендеру – все це набагато складніше. Під-час створення тривимірного середовища дуже важливо працювати поетапно та обережно, бо лише одна помилка може спричинити дуже багато проблем та змусити персонажа або інші об'єкти зникнути.

Окрім звичних *2D* та *3D*-ігор існують також ігри з застосуванням тривимірного оточення та персонажів, але ігровий процес обмежений двома вимірами. Такі ігри називають *2,5D*. Камера у них може демонструвати оточення збоку та рухатися зліва направо, але гравець рухатися лише у двох вимірах. Такий ефект тривимірності грає лише візуальну роль під час освоєння ігрового світу.

Зазвичай над одною тривимірною грою працює велика група розробників, вона включає в себе експертів з візуального мистецтва, світла, анімації, розробників *3D*-моделей, художників, дизайнерів рівнів. Розробники повинні постійно точно координувати свої дії.

В той же час, двовимірне ігрове середовище складається з лише єдиного шару, в якому персонаж пересувається та взаємодіє з предметами. Це набагато спрощує процес розробки. Створення такої гри відбувається набагато легше, ніж тривимірної, до цього залучають менше людей і сам процес розробки відбувається набагато швидше.

Основні переваги тривимірної розробки ігор:

1. *3D*-ігри більш видовищні, вони можуть краще занурити гравця в ігровий світ.
2. Вони представляють цілий спектр додаткових можливостей для розробників.
3. Такі проекти виконують великі команди розробників, що може прискорювати процес створення.
4. *3D*-напряму більш перспективний в сфері розробки ігор.

Але, незважаючи на це, у двовимірної розробки ігор також є свої переваги.

					КРБ.КІ.1.442-03.1.11	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дат		

Такими перевагами можуть бути:

- невеликі вимоги для рендеру;
- така розробка потребує менше сил та ресурсів для реалізації;
- зосередження на художніх деталях, історії та геймплеї, без витрати часу на роботу з фізикою та моделями у тривимірному просторі;
- можливість гри отримати популярність саме за рахунок індивідуальної атмосфери та деталей.

Ігри у жанрі квест за свою історію пройшли через багато перетворень. Перші такі ігри були текстового змісту та не могли використовувати графіку через технічні обмеження комп'ютерів. Спочатку ілюстрації були декоративними, зображення на екрані було статичним, а гравець все ще вводив послідовності команд з клавіатури для керування персонажем.

Згодом з'явилась можливість нової взаємодії з ігровим світом: можна було натискати мишкою на деякі ділянки екрану та отримувати заданий результат. Таким чином, розвиток квестів привів до появи повноцінних двовимірних відеоігор з застосуванням звукових та інших технологій.

Але вважається, що поява тривимірних квестів не мала великого успіху серед любителів даного жанру ігор. Їх прихильникам не подобались такі значні зміни та потреба пристосовуватись до них, а іноді вони і взагалі віддавали перевагу більш реалістичним іграм та жанру *RPG*, який починав набирати популярність.

Однією з проблем була відмова від зручного для усіх режиму *point-and-click*, тепер гравець керував персонажем за допомогою стрілок клавіатури.

Головні відмінності між *3D* та *2D* квест-іграми полягають у камері, способах управління персонажа та у складності проходження самої гри.

У *2D*-іграх камера дуже спрощена, нема такої перспективи, як у *3D*. Елементи управління досить прості та персонаж не має так багато можливих взаємодій з іншими об'єктами. Також у такій двовимірній грі може бути присутній ефект паралаксу, при якому фон прокручується з іншою швидкістю, ніж передній план. Такий ефект застосовується для створення ілюзії глибини.

					КРБ.КІ.1.442-03.1.11	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дат		

У 3D-квестах камера забезпечує можливість дивитися на світ під різними кутами. Переміщення камери в певне положення може бути дуже важливим, наприклад це може бути ключем до вирішення даної нам задачі. Анімація персонажів у тривимірному світі складніше. Використовується не тільки декілька встановлених анімацій, як у 2D, а всі можливі варіанти руху об'єктів або персонажів у тривимірному ігровому просторі. Загалом, всі елементи гри, особливо освітлення та звук, застосовуються для надання більшій реалістичності грі.

Ще одна відмінність від двовимірних квестів полягає в збільшенні поставлених перед гравцем задач, це збільшує складність проходження гри. Завдяки таким змінам основною метою ігрового процесу стає не просто проходження гри від початку до кінця, а дослідження простору, вирішення додаткових головоломок та багато чого іншого.

1.3 Процес створення квесту

Під час створення гри розробники мають перед собою багато задач та обов'язків. Для створення успішного квесту розробники використовують декілька їх обов'язкових властивостей.

Перша та головна властивість – у таких ігор завжди є кінцева ціль. Гравець обов'язково розуміє, що він повинен досягнути та які для цього виконуються дії. Як приклад можна привести *The Legend of Zelda*: головний герой усвідомлює, що його головний квест у грі – це спасіння принцеси. Про цю ціль йому нагадають під час проходження гри.

В іграх бувають також і інші активності. Якщо ви самі вирішуєте вивчати локації, збирати пасхалки або шукати рідкісну броню, то це вже побічні активності, які не є квестами і майже не впливають на проходження гри. Наприклад, в *Call of Duty: Modern Warfare* ви можете збирати ноутбуки під час місій, але вас не просять це робити.

					КРБ.КІ.1.442-03.1.11	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		16

Кожен квест передбачає якийсь геймплей, адже поставлене завдання потрібно вирішити і досягти мети. Квест насамперед має генерувати комплексний та цікавий геймплей, але водночас підтримувати основну сюжетну лінію, давати інформацію про ігровий світ.

Ще одна важлива властивість – це нагорода по завершенню завдань. Часто наш мозок однаково сприймає будь-яку нагороду, чи то ігрова валюта, чи корисний предмет, чи пара пікселів. Їх значимість буде відрізнятися не від форми, а залежно від прикладених зусиль. Саме головне, щоб для самого гравця це було емоційно важливо.

Під час створення ігор даного жанру доводиться вигадувати багато завдань для гравця. Оглядаючи подібні ігри можна побачити основні принципи, за якими їх складають.

1. Гравець завжди правий. Такий підхід означає повну орієнтованість на гравця, його побажання та погляди на те, як може бути організований ігровий процес.

2. Підказки потрібні, але не можуть бути «прозорими». Будь-яка підказка повинна натякати на правильну відповідь, але не розкривати її суть повністю.

3. Гравець повинен мати спроможність дійти до кінця, навіть не маючи досвіду в аналогічних іграх.

4. Не давати завдання, на виконання яких не має шансу.

5. Не змушувати гравця виконувати одні і ті самі завдання. За кожним завданням повинно бути інше, вони будуть логічно зв'язані та мати якусь мету. Загалом, створення квесту починається з побудови сторілайну. Це розгорнутий синопсис усієї гри або якоїсь значущої її ділянки. Його пишуть під час препродакшену, він також входить у дизайн документ, який у свою чергу визначає основне бачення гри: механіки, стилістику, цільову аудиторію та ще багато чого.

Після сторілайну пишуться ланцюжки квестів. Зазвичай їх складають у вигляді візуальної схеми, де один пункт – це один визначений квест, проте

					КРБ.КІ.1.442-03.1.11	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дат		

пізніше, під час безпосередньо розробки, він може бути роздроблений. На цьому етапі найчастіше основні механіки гри вже вигадані та викладені в дизайн документі.

Головне завдання дизайнера – це передати написане в сценарії через квести в рамках вже існуючих механік.

Коли ланцюжок квестів затверджується, починається робота над окремими складовими. Тут доводиться враховувати набір нюансів. Дизайнеру потрібно розуміти, які ресурси він має: скільки художників, аніматорів і розробників моделей працює в студії і який обсяг їх часу можна витратити на один квест.

Не менш важливо знати, які ігрові механіки доступні. На основі всього цього наративний дизайнер прописує основні дії в квесті і намагається досягти хорошого темпоритму, щоб активності не повторювалися.

Затвердивши основні дії, наративний дизайнер приступає до наповнення квесту – ставить завдання художнім відділам, а сам приступає до текстової обв'язки: діалогів, записок, описів та іншого.

Потім все це збирається та починаються тести. До них підключаються як самі творці, так і тестовий відділ. Після фіксування всіх багів та виправлення помилок, квест готовий. Цей підхід являється найпоширенішим, так працюють одні з найкращих студій.

Як і в будь-якій великій комплексній системі, під час роботи над квестами з'являється купа проблем. Під час їх виправлення, кожна студія у цьому процесі змінює щось своє. Наприклад, окремо пишуть сюжет, не думаючи про обмеження механік, двигуна або бюджету. А потім віддають цю історію квест-дизайнеру, який піклується про те, як будуть грати в гру з таким сюжетом. Квест-дизайнер заказує механіку, планує рівні, прописує діалоги, заказує графіку. Потім все збирається в першу версію гри, яку можна пройти. Після тестування та фінального схвалення контент віддається назад для фінальних діалогів, записок та іншого.

					КРБ.КІ.1.442-03.1.11	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дат		

1.4 Огляд існуючих аналогів

Для аналізу ігор з елементами квесту, розглянемо наступні приклади.

Syberia – комп'ютерна квест гра, з видом від третьої особи. Гра була випущена французькою студією *Microïds* у 2002 році. Головна мета гравця полягає у розв'язуванні різноманітних головоломок і виконванні певних процедур для того, щоб сюжетна лінія продовжилася.

Переваги:

- унікальна атмосфера та стиль;
- цікавий сюжет;
- характерні персонажі;
- музичний супровід;
- загадки та головоломки;

Недоліки:

- повільний темп гри;
- відсутність озвучки всіх діалогів;
- застарілі елементи геймплея.

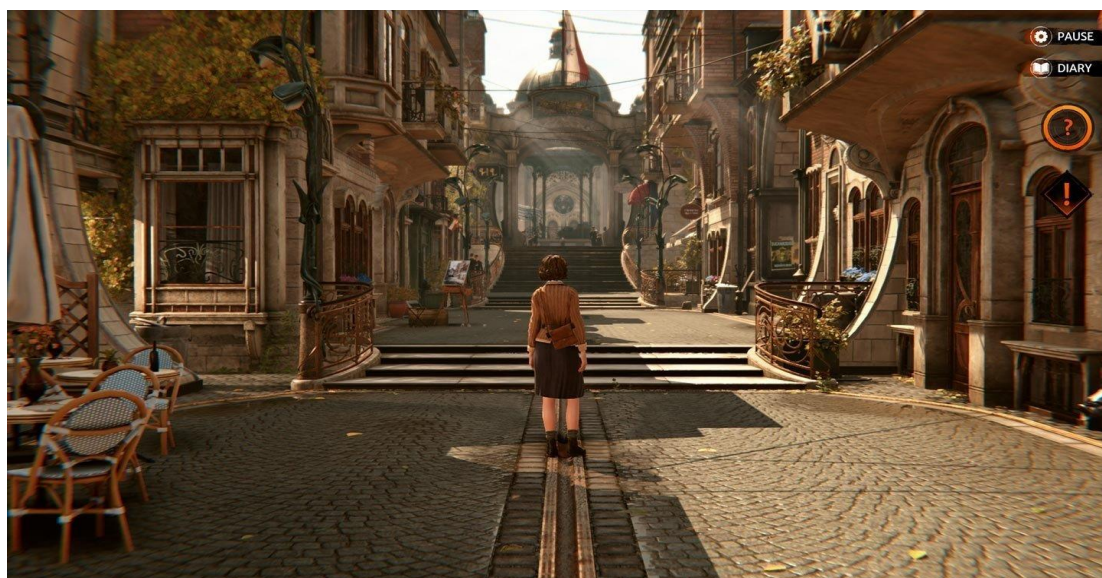


Рис. 1.1 – Проходження гри *Syberia*

					КРБ.КІ.1.442-03.1.11	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		19

The Longest Journey – комп'ютерна гра у жанрі квест. Була розроблена норвезькою компанією *Funcom* та випущена у 1999 році. Сюжет представляє собою історію про дівчину котра мандрує між паралельними світами, щоб відновити баланс.

Переваги:

- глибокий сюжет;
- багатий світ;
- візуальний стиль;
- невеликі системні вимоги.

Недоліки:

- складність головоломок.



Рис. 1.2 – Проходження гри *The Longest Journey*

The Black Mirror 3 – комп'ютерна гра жанру квест, що була розроблена німецькою компанією *Cranberry Production* та випущена у 2011 році. Гра представляє собою класичний point-and-click квест. Мета гри полягає в тому щоб досліджувати різні локації, збирати предмети, вирішувати головоломки.

Переваги:

- дуже влучна музика та інші звуки;

					КРБ.КІ.1.442-03.1.11	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дат		

- атмосферний сюжет;
- якісна графіка;
- цікаві механіки;
- глибока проробка персонажів;

Недоліки:

- обмежена інтерактивність;
- неоднозначна кінцівка.



Рис. 1.3 – Проходження гри *The Black Mirror 3*

Scratches – комп'ютерна гра в жанрі хорор-квест від першого обличчя. Була розроблена аргентинською студією *Nucleosys* та випущена у 2007 році. *Scratches* являє собою point-and-click квест з атмосферою хорора.

Переваги:

- напружена атмосфера;
- захоплюючий сюжет;
- деталізовані локації;
- невеликі системні вимоги.

Недоліки:

					КРБ.КІ.1.442-03.1.11	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		21

- повільний темп гри;
- складні головоломки;
- обмежена інтерактивність.



Рис. 1.4 – Проходження гри *Scratches*

Fetch – комп'ютерна гра з елементами квесту та аркади. Представляє собою фантастичні пригоди головного героя з цікавими завданнями та вражаючим візуальним оформленням.

Переваги:

- логічна та зрозуміла механіка гри;
- вражаючі ілюстрації;
- невеликі системні вимоги;
- велика кількість головоломок;
- підходить для широкої аудиторії.

Недоліки:

- занадто швидке проходження гри;
- музичне супроводження та інші звуки.

					КРБ.КІ.1.442-03.1.11	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дат		



Рис. 1.5 – Проходження гри *Fetch*

Через аналіз існуючих аналогів можна побачити їх основні переваги та недоліки. Такий аналіз здійснюється для виявлення найважливіших характеристик, яким треба приділяти увагу при створенні власної гри.

Таким чином, ми можемо побачити, що одними з найважливіших чинників є: сюжет, графіка, управління у грі, музичне супроводження, системні вимоги та складність проходження. Музика та візуальна складова у багатьох випадках є перевагами, вони налаштовують приємну атмосферу. Також однією з переваг можна вважати невеликі системні вимоги, тому що це означає можливість грати навіть з не дуже потужних пристроїв.

Основними проблемами є: управління, складання ігрових завдань для гравця та сама складність гри. Зазвичай такі проблеми виникають через неможливість розробників передбачити погляди гравців на організацію ігрового процесу.

Беручи до уваги всі ці переваги та недоліки, ми можемо почати створювати власну гру з метою позбавитися проблем, які зазвичай виникають, та урахувати побажання та погляди майбутніх гравців.

1.4.1 Вимоги до створення власної гри

За результатами аналізу тривимірних ігор з елементами квесту ми виявили основні характеристики, що являються важливими при їх створенні. У всіх ігор є свої недоліки, проблемні сторони, які потрібно брати за увагу під час створення власної. Виходячи з цього, можна створити основні вимоги до її створення.

Дуже важливою є повна орієнтованість на гравця, тому одна з перших і найголовніших вимог – це зручне управління у грі. Це означає налаштування його таким чином, щоб гравці не повинні були пристосовуватися до багатой кількості змін, або давати їм можливість вносити свої зміни.

Сюжет також є важливим, але не можна концентруватися лише на ньому, тому що гра має багато інших характеристик, які повинні гармонійно поєднуватися. Наприклад, музика та інші звуки. Вони мають бути влучними та не надокучати, але гравець все одно повинен мати можливість вимкнути їх у будь-який момент. Ще однією важливою вимогою можна вважати здібність не навантажувати гру непотрібними додатковими елементами.

До уваги також треба брати самі квести та їх виконання. Головними вимогами до них є: наявність підказок, можливість виконання різними способами та оригінальність. Останнє особливо показує те, як прихильникам даного жанру ігор важлива різноманітність завдань та головоломок. Квести також повинні бути різної складності.

Такі основні вимоги можна використовувати під час розробки власної гри, особливо приділяючи увагу проблемам, які виникли у вже існуючих аналогах.

Висновок до першого розділу

За допомогою детального огляду предметної області було проаналізовано основні властивості комп'ютерних ігор, а саме ігор жанру

					КРБ.КІ.1.442-03.1.11	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дат		

квест. Можна побачити, через яку довгу історію вони пройшли, починаючи від ігор виключно текстового змісту, без застосування графіки, та закінчуючи сучасними тривимірними квестами з вражаючим геймплеєм та ігровим світом. Ми дізналися про відмінності між двовимірними та тривимірними іграми, що в основному полягають у складності реалізації та кількості людей, залучених до створення даної програми.

Було оглянуто процеси створення квестів, основні етапи, властивості та принципи, якими користуються розробники під час створення.

Було також проаналізовано існуючі аналоги з метою виявлення їх переваг та недоліків. Як ми можемо бачити, основними проблемами є: управління, складність поставлених задач та, навпаки, проходження гри у занадто короткий час, що може вказувати на дуже легкі ігрові завдання. Також було виявлено, що візуальна складова та музичне доповнення є дуже важливим та сильно впливає на реакцію публіки на дану гру.

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						25
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>		

РОЗДІЛ 2

ПРОЕКТУВАННЯ КОМП'ЮТЕРНОЇ ГРИ

2.1 Створення концепт-документу

Концепт-документ створюється для короткого опису концепції гри, чим вона буде цікава для гравців та її особливостей. Такий документ містить стандартні пункти, які відповідають на основні питання, які можуть виникнути до майбутньої гри.

Головний герой опиняється в себе вдома. Він не пам'ятає майже нічого з минулого тижня. Головний герой знаходить записку з малюнком, яка може допомогти у відтворенні своїх спогадів. Записка з незрозумілим малюнком має бути підказкою, що він повинен робити. Чи зможе герой з'ясувати, для чого служить ця записка та хто він є насправді?

За жанром гра є квестом (*Walking Simulator*).

Гра орієнтована на широку аудиторію, але через наявність табачних виробів.

Мінімальний вік гравця – 18 років.

Основні особливості гри:

- незвичайні та захоплюючі локації та завдання;
- цікава історія;
- деталізовані локації;
- в залежності від вибраного шляху, можна завершити гру декількома різними кінцівками;
- оригінальний підхід до проектування рівнів та багато особливостей;
- невисокі системні вимоги;
- динамічне оточення.

					КРБ.КІ.1.442-03.1.11	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дат		

Головним завданням гравця є проходження сюжету, виконуючи різноманітні квести. З кожним етапом гравець все більше дізнається про ігровий світ.

Мета – отримання унікального ігрового досвіду під час проходження квестів та повне проходження гри.

Різні предмети можуть допомагати герою, можуть також заплутувати його.

Пересуваючись по ігровому світу, гравець отримує квести та знаходить предмети, він має виконувати дані йому завдання для досягнення своєї мети – він хоче знайти те, що допоможе йому згадати своє минуле. Питання про те, чи зможе герой це здійснити – залишається за гравцем.

Гра представляє собою тривимірний квест, в якому проходження рівнів складається з виконання спеціальних завдань та квестів. З кожним рівнем складність виконання може підвищуватися. На кожному рівні на екрані з'являються підказки у виді предметів, які треба знайти та зібрати гравцю.

Підказки є важливими перевагами, але іноді можуть заплутати гравця.

Ми починаємо гру в замкнутому просторі, перше, що треба зробити – знайти якесь джерело світла. Для цього гравець повинен знайти потрібний предмет та застосувати його.

Предмети у грі можна збирати, додавати до інвентарю, комбінувати та згодом застосовувати для виконання завдань у грі.

Для виконання самого квесту переходимо у інші кімнати, першим завданням є збір необхідних предметів та уникнення перешкоди у вигляді стіни.

Предмети ми повинні скомбінувати за допомогою системи крафту, а для дверей можна використати спеціальний ключ, який гравець повинен зібрати, пересуваючись по ігровому рівню.

					КРБ.КІ.1.442-03.1.11	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дат		

Коли завдання буде виконано – стіна розтрощується за допомогою молотка.

Тепер персонаж може пройти далі та виконати інше завдання. Його можна виконати тільки з допомогою предмета, який ми повинні були забрати ще на початку гри. У випадку, коли гравець не забрав його – виконання завдання неможливе, треба повернутися назад. У іншому випадку можемо продовжити.

Наступне завдання полягає у застосуванні потрібного предмету та переходу на наступний рівень.

У разі помилок під час виконання завдань гравець отримує відповідне повідомлення на екрані. Після виконання завдання можна отримати бонуси та повернутися назад для повного завершення квесту та отримання додаткового балів досвіду.

Інші квести будуються за аналогією, але стають все складніше, залежно від ігрового рівня.

Основні локації гри: перший поверх, другий поверх та горище.

Для виконання завдань використовуються усі ці локації, в залежності від теми самого завдання.

Гра є оригінальною, але деякі її концепції були запозичені або являються схожими з іншими іграми даного або схожого жанру.

Dear Esther – це атмосферний інтерактивний квест від першої особи, який можна віднести до жанру walking simulator. Ця гра має спільні риси з грою, що розроблюється.

No One Lives Under the Lighthouse – це атмосферний інді-хорор від першої особи. Гра має схожу атмосферу та геймплей.

					КРБ.КІ.1.442-03.1.11	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дат		

Системні вимоги

Комплектуючі	Мінімальні вимоги
Оперативна система	<i>Windows 10</i>
Процесор	<i>2.5GHz</i>
Оперативна пам'ять	<i>4GB ОЗУ</i>
Відеокарта	<i>2024Mb</i>

2.2 Опис гри

2.2.1 Інвентар

Інвентар у грі — це системний елемент, який дозволяє гравцю зберігати, організувати і використовувати знайдені предмети протягом гри. Інвентар може бути реалізований у різний спосіб залежно від жанру і механік гри.

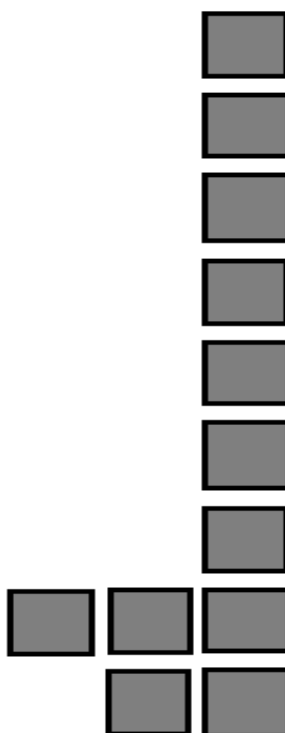


Рис.2.1 – Схема інвентарю

2.2.2 Крафт

Крафт в іграх — це механіка, яка дозволяє гравцям створювати нові предмети, збираючи і комбінуючи різні ресурси або компоненти. Цей процес зазвичай включає видобуток матеріалів, знаходження рецептів або схем і використання інструментів чи верстатів для створення кінцевого продукту.

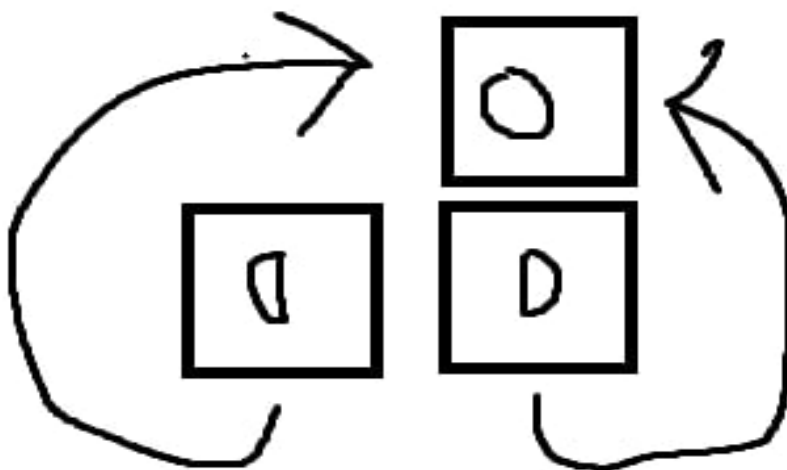


Рис.2.2 – Схема крафту

2.2.3 Використання предметів

Використання предметів в іграх є однією з ключових механік, яка дозволяє гравцям взаємодіяти з ігровим світом і вирішувати різні завдання. Ця механіка включає багато різних аспектів і може бути реалізована по-різному залежно від жанру ігри.

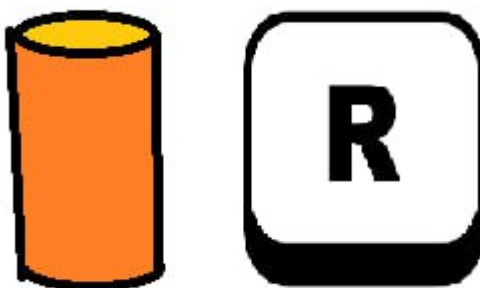


Рис.2.3 – Схема використання предмету

Змн.	Арк.	№ докум.	Підпис	Дат

2.2.4 Руйнування об'єктів

Руйнування об'єктів у грі — це механіка, яка дозволяє гравцям впливати на навколишнє середовище, руйнуючи різні об'єкти. Ця механіка додає ігровому процесу реалістичності, динамічності та нових стратегічних можливостей.

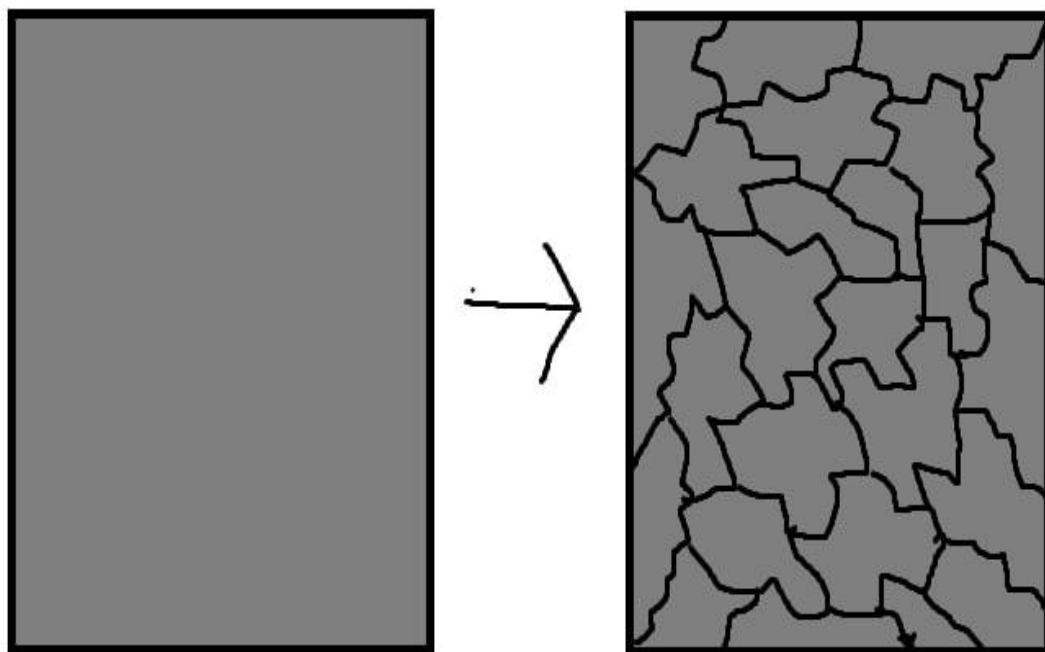


Рис.2.4 – Схема руйнування стіни

2.2.5 Покрокова збірка предметів

Покрокова збірка предметів у іграх — це процес, який вимагає від гравця поетапного збору компонентів або ресурсів для створення або завершення певного об'єкта чи механізму. Ця механіка зазвичай включає кілька етапів або кроків, кожен з яких повинен бути виконаний в правильному порядку, щоб досягти кінцевого результату.

Змн.	Арк.	№ докум.	Підпис	Дат

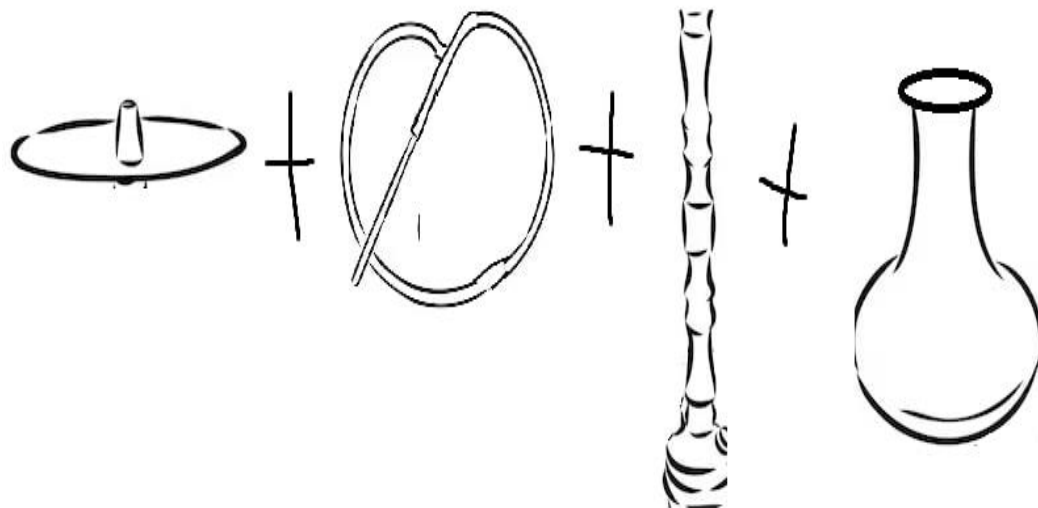


Рис.2.5 – Схема покрокової збірки предмету

2.2.6 Ігрове оточення

Основна локація (квартира) має шість приміщень, в кожному з приміщень будуть заховані об'єкти для збірки одного із трьох головних предметів запланованого квесту.

1. Ігрова кімната – це приміщення звідки починається гра. Вона представляє собою невелике приміщення, яке має стілець, стіл, комп'ютер, квіти та картини. Тут гравець знаходить записку з малюнком, котра допомагає зрозуміти ціль гри.

2. Вітальня кімната саме найбільше приміщення в грі, котре заповнене різними об'єктами. Такими як: шафи, стіл, стільці, тарілки, настільна лампа. У цій кімнаті гравець може знайти ліхтарик, який буде корисним для пошуку інших об'єктів для головного завдання.

3. Кухня містить сучасну побутову техніку, кухонні шафи та духовку, котра після збору та використання кальяну відкриється. В духовці гравець повинен знайти ще одну частину молотку, котра буде використана в крафті молотку.

4. Вбиральня містить в собі частину цілого об'єкту.
5. Ванна кімната оснащена стандартними сантехнічними приладами: ванною, умивальником, дзеркалом і туалетом. Тут також гравець має знайти ще одну частину об'єкту.
6. Комора – це кімната в котрій наявні шафа, стіл та частина об'єкту.

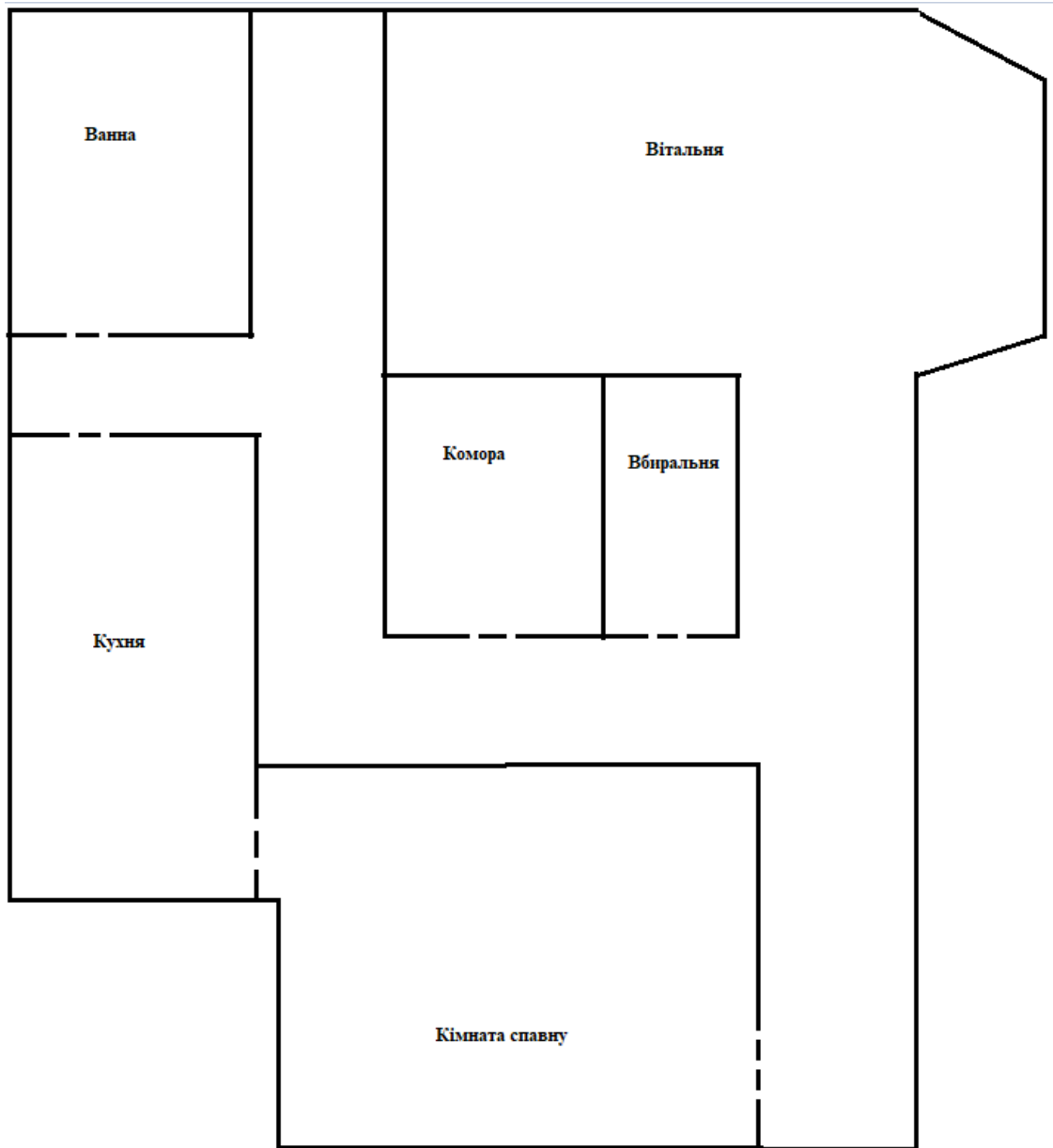


Рис. 2.6 – Схема дому

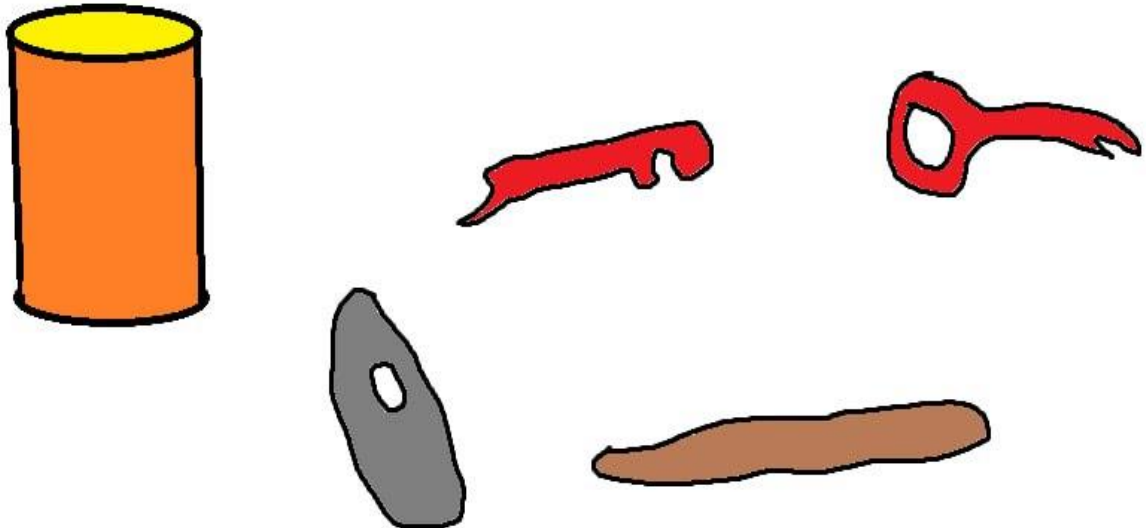


Рис. 2.8 – Приклади предметів для взаємодії

Локація доповнюється предметами та об'єктами, з якими гравець може взаємодіяти. Наприклад, таким предметом можуть бути двері, ліхтарі та інше. Предмети можна буде забирати до інвентарю, що створюється разом з інтерфейсом користувача.

Робота та управління інвентарем буде здійснюватися за допомогою спеціальних скриптів та компонентів *UI* у вибраному програмному засобі. Інвентар можна буде відкрити та закрити за допомогою спеціальної кнопки.

У грі з різних предметів можна буде створювати інші предмети, переходити на інші рівні з їх допомогою та виконувати різноманітні завдання.

Гравець буде взаємодіяти з об'єктами за допомогою скриптів та їх функцій, які ми будемо створювати на наступному етапі. Скрипти будуть допомагати рухати персонажа, камеру за персонажем, програвати створені нами анімації, звуки та музику.

2.3 Текстурування та анімація моделей

Текстурою називається растрове зображення, яке застосовується до моделі. Створення таких текстур відбувається через малювання її у графічному редакторі, побудування на основі процедурних карт або застосуванням процедурних технік 2D або 3D формату. У цілому, текстури представляють собою спосіб надання деталям поверхні. Частіше такі зображення імітують якісь природні матеріали: папір, камінь, дерево, метал та інші.

Для утворення параметрів візуального відображення застосовують карти текстур. Типами таких карт можуть бути:

1. *Diffuse map* – карта базового кольору.
2. *Specular map* – карта відбиття.
3. *Bump map* – карта рельєфу.
4. *Normal map* – карта для імітації рельєфу поверхні.

Також бувають карти нормалей, що застосовуються для об'єктів, що не деформуються та, навпаки, для можливості деформувати об'єкти.

У 3D-іграх анімація є важливим аспектом, який додає реалістичності та життєвості персонажам та об'єктам. Існує кілька методів для створення анімації у 3D-іграх.

Ключова анімація (Keyframe Animation) включає створення ключових кадрів, які визначають основні позиції персонажа або об'єкта у важливі моменти руху. Потім програмне забезпечення інтерполює проміжні кадри, щоб створити плавний перехід між ключовими позиціями. Цей метод дозволяє контролювати деталі руху, забезпечуючи високий рівень точності.

Програмна анімація (Procedural Animation) створюється за допомогою алгоритмів і коду, а не ручного анімування. Цей метод використовується для створення анімацій, які змінюються в залежності від умов гри, наприклад, фізичні реакції, рухи натовпу або інші динамічні ефекти. Програмна анімація дозволяє адаптувати рухи до змінних ситуацій у реальному часі.

					КРБ.КІ.1.442-03.1.11	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дат		

Під час створення гри застосовується велика кількість текстур. Для текстурування персонажів та подальшого створення анімацій їх рухів ми застосовуємо ми застосовуємо процедурну анімацію, яка дозволяє створювати плавні та динамічні рухи у реальному часі.

Для створення анімації броску об'єкта використовується наступна схема (рис. 2.6).

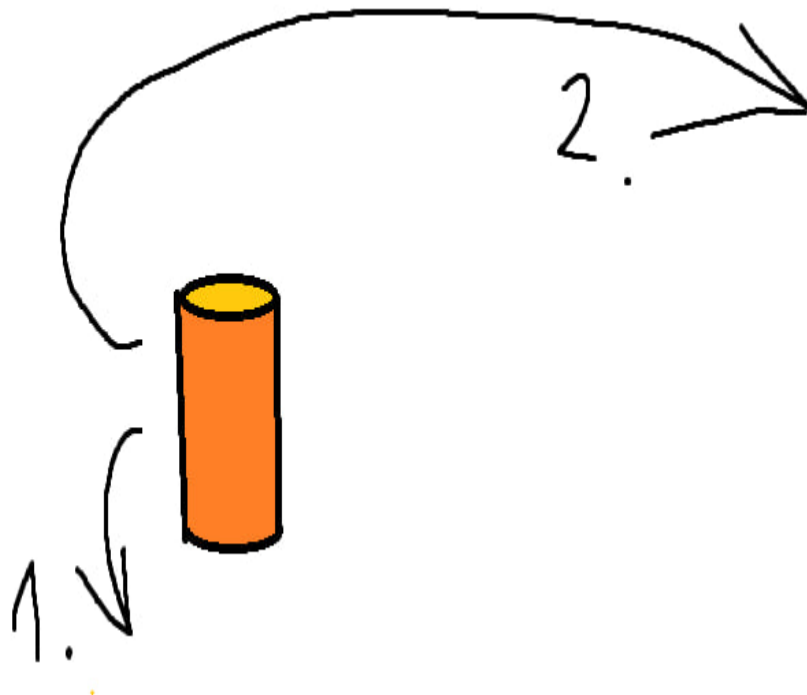


Рис. 2.6 – Зображення для анімації броску

Загалом, головний персонаж має 4 основні види анімації:

1. *Idle* (анімація персонажа у стані спокою).
2. *Jump* (анімація стрибка).
3. *Run* (анімація бігу).
4. *Fall* (анімація падіння).
5. *Throw* (анімація броску).

Кожна з них згодом буде застосовуватися при різних діях героя.

2.4 Запис та обробка звукових ефектів

Музика у грі – це найкращий інструмент для задання правильного настрою під час проходження цієї гри. Вона є однією з найважливіших частин, тому що має спроможність позитивно впливати на ігровий процес, покращувати його.

Для надання грі потрібної атмосфери в ній застосовується різноманітна музика та звуки. Наприклад, це можуть бути звуки пересування персонажу, звуки інтерфейсу, які застосовуються у меню, або звуки під час отримання нових досягнень. Також є головна тема та застосування різної музики на різних рівнях.

Ігровий саундтрек може складатися з декількох основних елементів:

1. Музика.
2. Звуковий інтерфейс.
3. Фонові звуки.
4. Озвучування персонажів.

Правильно підібрані звуки мають допомагати геймплею, просувати головні ідеї сюжету та допомагати персонажу орієнтуватися у грі. Такі звуки зазвичай різняться від одного жанру до іншого. Наприклад, ми не можемо використовувати занадто позитивну та веселу музику у іграх з елементами жанру хорор, або іграх з дуже сумним сюжетом.

Для створення власної гри будемо застосовувати декілька музичних треків для головної теми, меню та різних рівнів. Також будуть застосовуватися звуки у користувацькому інтерфейсі, звуки отримання виконання завдань.

Основні музичні треки – це самостійно записаний трек за допомогою *FL Studio*. Трек є головною темою, яку можна почути більшість часу проходження гри. Звуки для гри були записані самостійно за допомогою підручних матеріалів, в програмі «Audacity».

					КРБ.КІ.1.442-03.1.11	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дат		

Створенні музика та звуки ідеально підходять до заданої атмосфери та жанру гри, вони не заважають під час ігрового процесу та навіть мають доповнені функції. Це означає, що у деякі моменти вони можуть допомогти гравцеві у виборі або у вирішенні завдань.



Рис. 2.7 – Звуки гри

Для програвання музики або інших звуків гри ми додаємо об'єкт *Audio Source*, тобто джерело звуку. Цей об'єкт містить в собі можливість змінювати гучність звуку, редагувати його та зациклювати. Також для програвання музики у грі треба переконатись у тому, що камера містить об'єкт *Audio Listener*.

2.5 Проектування рівнів гри

Одна із основних задач на проектування власної гри – це правильне побудування ігрових рівнів.

Методика проходження рівнів повинна бути зрозумілою гравцеві, а при наявності питань гравець повинен мати можливість отримати підказки та додаткову інформацію.

Сам процес створення оточення та дизайну рівнів може виглядати таким чином:

1. Основна ідея для рівня.
2. Базові начерки оточення для пересування персонажа.
3. Збір референсів.
4. Моделювання оточення.
5. Фінальний продукт.

Такий підхід до проектування рівнів можна вважати простим і зрозумілим, але на практиці все набагато складніше. Зазвичай, з'являється велика кількість проблем після яких рівні та способи їх проходження піддаються змінам.

Для створення рівнів у власній грі треба розуміти свою передбачувану аудиторію, що її може зацікавити та відштовхнути.

Розробка цікавої та захоплюючої квест-гри не означає створити її максимально складною, з дуже великою кількістю завдань, які майже неможливо виконати. Це особливо стосується ігор, які направлені не тільки на дорослу аудиторію, а і на дітей різного віку.

Аналізуючи схожі за жанром ігри та реакцію на них, можна виявити головні властивості, які треба брати до уваги, проектуючи рівні гри:

- рівні повинні бути розважальними;
- навігація у рівнях дуже важлива;
- непередбачуваність є перевагою;
- наявність додаткових елементів для огляду;
- дизайн рівнів має нести в собі ідею та принципи гри;
- сам герой та ігрове оточення мають доповнювати один одного;
- елементи, з яких складається рівень мають показувати частину сюжету;
- дуже важливий баланс між візуальним оформленням та основною ідеєю;
- гравець має відчувати, що саме він приймає рішення.

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дат		

Застосовуючи ці знання, починаємо проектувати рівні для своєї гри.

Усі рівні в основному будуються за схожою схемою:

1. Персонаж переходить на цей рівень з меню або граючи послідовно, тобто, переходячи с першого рівня на другий, з другого на третій і так далі.
2. Опиняючись на новому рівні треба прочитати підказки, які гравцю дають.
3. Далі гравець має оглянути ігрове оточення на наявність схованих предметів, об'єктів та персонажів, з яких можна отримати інформацію.
4. Визначаючи квест або завдання, гравець має виконати його. Вони можуть бути як дуже швидкими загадками, які треба виконати за певний проміжок часу, так і квестами, потребуючими роздумів.
5. Після виконання певних завдань гравець може перейти на наступний рівень, а у разі появи труднощів скористуватися підказками від оточення.

Для побудування рівнів у нашій грі створимо схему їх проходження. Така схема буде схожою для всіх рівнів, за виключенням розташування деяких ігрових предметів, об'єктів та за наявності різних квестів.

Дана схема ілюструє саме процес проходження першого рівня (рис. 2.8).

Інші рівні будуються за аналогічною схемою.

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дат		

Таким чином, можна дізнатися, що для виконання квесту та повернення назад треба також виконати завдання для уникнення перешкоди (такої як: стіна). Для цього збираємо потрібні предмети, застосовуємо їх та переходимо перешкоду.

Наступна ціль – це завершення виконання квесту. На першому рівні він не складний – гравець повинен вирішити досить просту задачу, а потім піти до заблокованої двері, щоб відкрити її та перейти на наступний рівень.

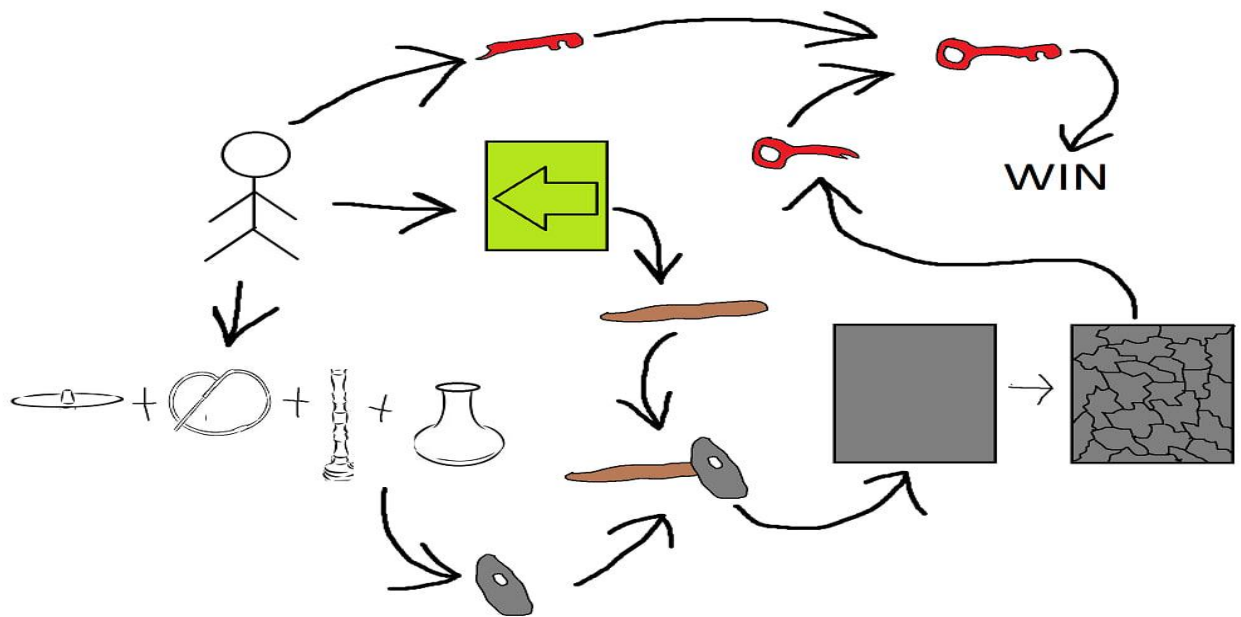


Рис. 2.9 – Схема проходження ігрових завдань

Висновок до другого розділу

Під час проектування власної гри було отримано багато нової інформації про розробку моделей, створення текстур та анімацій, та процеси проектування рівнів у грі. На початку проектування було створено концепт-документ з основними та додатковими відомостями про гру. В такому документі було описано короткий зміст гри, її жанр і аудиторію, особливості, порівняння зі схожими іграми та системні вимоги. Ці відомості відповідають на одні з основних питань до гри.

Було вибрано та створено моделі для персонажів та оточення, відібрано звукові ефекти та музику для власної гри. Ми також засвоїли процеси анімації та текстурування персонажів, за основою яких зробили власні анімації персонажів. Вони згодом будуть використовуватися для відтворення їх різних станів.

Під час проектування рівнів було утворено загальну схему їх побудування та опис одних з перших ігрових квестів.

					КРБ.КІ.1.442-03.1.11	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 3

РОЗРОБКА ГРИ

3.1 Вибір та обґрунтування програмного засобу

Перед початком розробки гри треба вибрати програмний засіб для її створення. Цей етап є дуже важливим, тому що правильно підібраний ігровий двигун та інші способи реалізації гри допомагають розробникам зробити її якомога краще.

Для створення двовимірної гри будемо застосовувати двигун *Unity*.

Unity – це багатоплатформовий інструмент, за допомогою якого можна розробляти відеоігри, застосунки та інше. Він дає можливість створювати програми, що будуть працювати на настільних персональних комп'ютерах, консолях, мобільних пристроях, пристроях з використанням доповненої або віртуальної реальності. На *Unity* було написано тисячі ігор, додатків, візуалізацій математичних моделей, що можуть охоплювати велику кількість платформ та жанрів.

У редакторі *Unity* відтворений інтерфейс *Drag&Drop*, що означає спосіб керування елементами інтерфейсу за допомогою комп'ютерної миші або сенсорного екрану. Він здійснюється через перетягування віртуальних об'єктів з одного місця у інше для виконання певних дій у програмі. Також у редактор можна встановити плагіни *KALI*, що слугують для налагодження гри прямо у редакторі.

Для написання скриптів *Unity* використовує *C#* та модифікацію *JavaScript*, відому як *UnityScript*, але вона вже не використовується у нових версіях *Unity*. Система скриптів зроблена на *Mono*, що представляє собою вільний відкритий проект з реалізації *.NET Framework*. Починаючи з версії 3.0 можна використовувати *MonoDevelop* для налагодження скриптів.

Скрипти у редакторі прикріплюються до об'єктів у виді окремих компонентів. А з версії *Unity 5.2*, що вийшла у 2015 році, редактор надає

					КРБ.КІ.1.442-03.1.11	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дат		

можливість редагувати скрипти у середовищі *Visual Studio*, яке ми і будемо використовувати для написання скриптів для нашої гри.

У *Unity* також є фізичні розрахунки. Їх відтворює двигун для фізики *PhysX* від *NVIDIA* для *3D* та *Box2D* для *2D* фізики. Редактор підтримує фізику твердих тіл та тканин, а також фізику типу *Ragdoll*. Також є графічний *API*(прикладний програмний інтерфейс) – *DirectX* для ефективного вирішення завдань при написанні відеоігор.

Проект *Unity* поділяється на сцени (або рівні), які являють собою окремі файли з ігровими світами та своїм набором об'єктів та налаштувань. У об'єктів є деякі набори компонентів, з якими і можуть взаємодіяти скрипти. У них може бути назва, тег, шар для відображення та інші властивості. Наприклад, у кожного об'єкта на сцені є компонент *Transform*, що містить у собі координати розташування, повороту та розмірів об'єкта по трьох осях.

Редактор має свою систему наслідування об'єктів: дочірні об'єкти повторюють усі змінення позицій, поворотів та масштабів батьківського об'єкту.

Двигун підтримує безліч популярних форматів. Для передачі своїх моделей, звуків, текстур, скриптів та матеріалів іншим розробникам, їх можна запакувати у формат *.unitypackage* та викладати у вільний доступ. Такий формат також використовується у внутрішньому магазині *Unity Asset Store*, де розробники можуть викладати різноманітні елементи для створення гри.

Unity підтримує написання та редагування шейдерів, має компоненти для створення анімацій.

Підтримується також і система *Level Of Detail*, що відтворює заміну на менш деталізовані моделі на далеких відстанях від гравця та навпаки. Така система та система *Occlusion Culling* знижують навантаження на центральний процесор та дозволяють оптимізувати проект.

Розглянемо головні переваги та недоліки редактору *Unity*.

Переваги:

— багато функціональних можливостей для створення гри;

					КРБ.КІ.1.442-03.1.11	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дат		

- візуальне середовище розробки;
- багатоплатформна підтримка;
- модульна система компонентів.

Недоліки:

- робота з багатокомпонентними сценами дуже ускладнюється;
- складне редагування префабів (шаблонів екземплярів);
- відсутність підтримки у *Unity* посилань на зовнішні бібліотеки.

За допомогою редактору *Unity* було створено багато відомих ігор, додатків та симуляцій. Він використовується як дуже відомими розробниками, так і для створення інді-ігор.

Одними з найвідоміших 3D-ігор, створеними завдяки *Unity* є:

1. *7 Days to Die*.
2. *Escape from Tarkov*.
3. *Firewatch*.
4. *Fall Guys*.

Створення двовимірної гри на *Unity* є досить зручним. Редактор містить у собі багато інструментів для її створення, також можна застосовувати ігрові набори та шаблони. У *Unity* є окремий двигун для 3D фізики, що робить оптимізацію доступною тільки для двовимірних об'єктів. 3D об'єкти представляють собою меши, тобто об'єкти різних предметів. Вони є стандартними, але для більш ефективного об'єднання та управління мешами у процесі розробки можна використовувати спеціальні техніки, що містить *Unity*. Наприклад, редактор колайдерів, що дозволяє отримувати потрібні колайдери та редагувати компоненти цих моделей предметів всередині готового об'єкту.

Розглянуті властивості *Unity* є дуже сприятливими для створення гри за допомогою цього інструменту та його компонентів.

					КРБ.КІ.1.442-03.1.11	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дат		

3.2 Створення основних кімнат

Сцени в *Unity* містять об'єкти гри, що розроблюється. Вони можуть застосовуватися для створення головного меню, окремих рівнів та іншого. Кожен файл сцени є окремим ігровим рівнем. В кожній сцені можна розмістити об'єкти оточення, декорації та інші об'єкти, таким чином, створюючи дизайн та саму гру.

Після створення нового проекту у *Unity* створюється нова сцена. Вона є порожньою, на ній нема нічого, окрім стандартних об'єктів – камери та джерела світла.

Кожен ігровий об'єкт складається з компонентів. Їх можна додавати до певного об'єкта та видаляти. Скрипти, наприклад, теж відображаються у панелі компонентів.

Із самих ігрових об'єктів можна створювати префаби. Префабом в *Unity* називають тип об'єктів, який дозволяє зберігати об'єкт зі всіма його компонентами та значеннями властивостей. Він виступає у ролі шаблону для створення екземплярів об'єкта, що зберігається на сцені. Будь-які зміни в префабі призводять до змін в усіх його екземплярах.

Геймплей у грі повинен бути відтворений так, щоб було зрозуміло, коли завершується рівень та при яких умовах завантажується сцена для наступного рівня або завершення гри.

Наступний рівень зазвичай завантажується при таких умовах:

- гравець виконав усі завдання, що дає йому змогу перейти на наступний рівень;
- гравець дістався певного місця у грі;
- у інтерфейсі гри є кнопки для переходу на різні рівні.

Спочатку будемо створювати сцену з головним меню. З нього можна буде почати гру, переходити на налаштування, обнулити ігровий процес, виходити з гри та переходити до панелі із кінцівками до гри.

Потім створюємо локацію квартири. На цій сцені будуть такі об'єкти:

					КРБ.КІ.1.442-03.1.11	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дат		

шафи, кімнати, столи та інші об'єкти, що потрібні для її реалізації. Створюємо на сцені спеціальне освітлення. Для цього додаємо до неї *Point Light*, тобто точкове освітлення, та редагуємо його налаштування.



Рис. 3.1 – Кімната «Вітальня» у локації «Квартира»

Створюємо панель із інвентарем . Для цього застосовуємо такі елементи *UI* (користувацького інтерфейсу), як *Image*. Також створюємо кнопки для відкриття інвентарю, переходу до панелі с предметами та виходу з початої гри. Натискаючи на «вихід» гравець потрапляє у головне меню.

Далі створюємо більше кімнат, які будуть репрезентувати «кухню», «вбиральню» і так далі. Головні об'єкти додаються на різні кімнати сцени, також відтворюється ефект освітлення різних кімнат сцени. Як і усі інші сцени гри, ця сцена має об'єкт *Canvas*, на який додаються усі головні елементи та кнопки, що потрібні під час гри: панель з відображенням інвентарю, вимкнення звуку, ігрові предмети та елемент для виходу зі гри.

Для того, щоб герой міг правильно пересуватися додаємо до об'єктів, що знаходяться на землі *Box Collider*.

Коллайдери в *Unity* використовуються для обробки фізичних властивостей об'єкту. Існують звичайні коллайдери для застосування до

тривимірних об'єктів . Вони мають свої функції (*Density, Material, Is Trigger, Offset, Size* та інші).

Для тривимірних об'єктів застосовують:

1. *Box Collider.*
2. *Mesh Collider.*
3. *Capsule Collider.*
4. *Sphere Collider.*
5. *Terrain Collider.*
6. *Wheel Collider.*

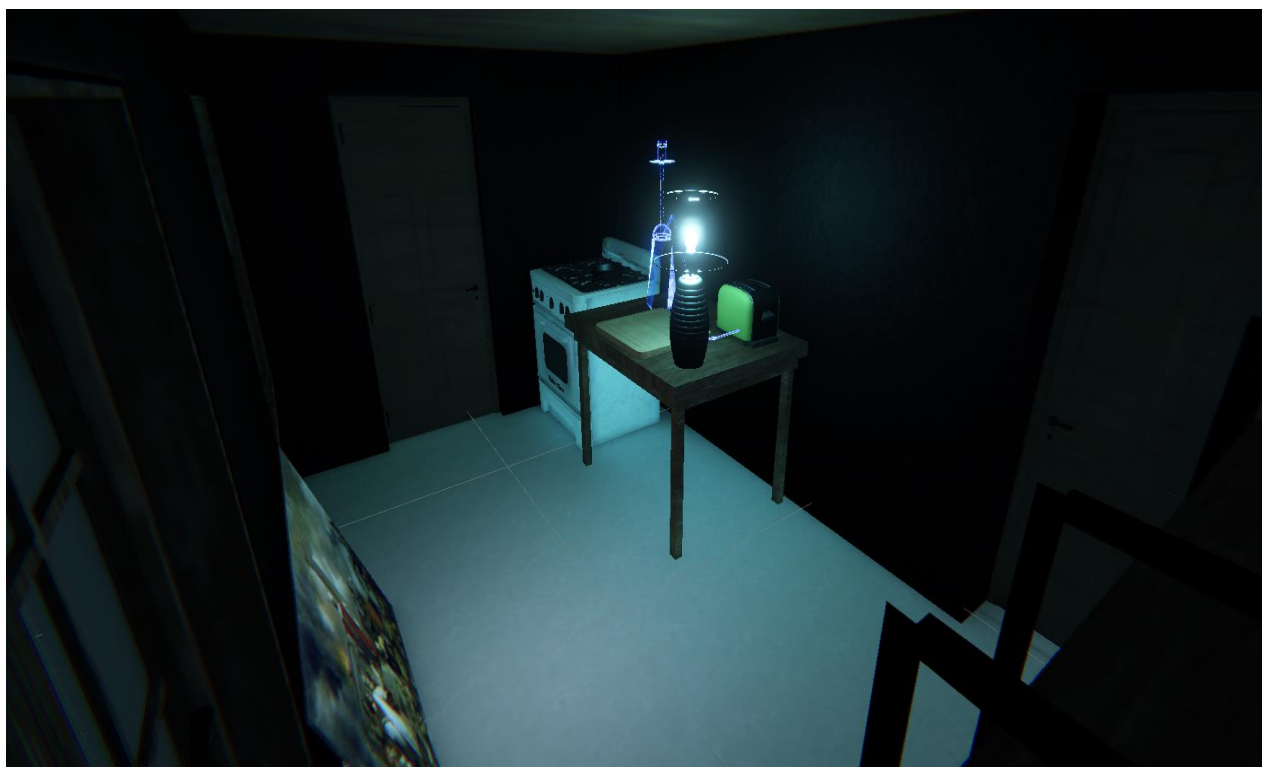


Рис. 3.2 – Кімната «Кухня» у локації «Квартира»

Коллайдери будуть широко використовуватися у процесі подальшої розробки гри.

Інші кімнати створюються за схожою схемою. Наступним етапом буде розміщення усіх ключових та декоративних об'єктів, додавання анімацій до певних об'єктів та написання скриптів.

3.3 Розміщення моделей, ігрових об'єктів та освітлення

На цьому етапі розміщуємо усі ігрові об'єкти, моделі та елементи, що потрібні для створення квестів, на створені раніше ігрові сцени. Редагуємо розміри та інші властивості всіх об'єктів. На наступному етапі розробки ми будемо створювати скрипти для взаємодій усіх цих предметів, об'єктів та персонажів з ними.

Для взаємодії персонажів та інших об'єктів з навколишнім світом додаємо до них коллайдери. Тип коллайдера використовується в залежності від форми об'єкта. Для головного персонажа це в першу чергу потрібно, щоб він не проходив крізь інші об'єкти та мав змогу пересуватися. Для пересування персонажа створюємо скрипт. Згодом будемо додавати скрипти для усіх взаємодій персонажа у грі та для самих квестів.

Освітлення додається та редагується після розміщення об'єктів.

У грі можуть бути різні типи освітлення, наприклад:

1. *Directional Light*.
2. *Point Light*.
3. *Spotlight*.
4. *Area Light*.

Спочатку розміщуємо всі об'єкти для кімнати «кухня».

Для пересування головного персонажу по сцені додаємо до нього коллайдер, компонент *Rigidbody 3D* та спеціальний об'єкт для перевірки того, чи стоїть персонаж на землі. Компонент *Rigidbody 3D* додається до усіх потрібних об'єктів для застосування фізики твердих тіл до них. Так ми зможемо реалістично переміщати їх, вони зможуть взаємодіяти з іншими предметами, а також можна буде керувати ними через скрипти.

Також для пересування персонажу, для відтворення його функціоналу та для правильного відтворення його анімацій створюється скрипт. Для аналогічних дій інших персонажів також створюємо скрипти.

					КРБ.КІ.1.442-03.1.11	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дат		

Переміщаємо на головну сцену потрібні предмети: ліхтарик, деталі об'єкту, який гравець буде збирати, ключ, що буде використовуватися для переходу на іншу локацію, та інші. Додаємо спеціальні іконки для навігації. Так, проходячи повз потрібні предмети, їх можна буде забрати та додати до інвентарю. Для цього будемо створювати спеціальні скрипти.



Рис. 3.3 – Розміщення предметів на сцені

Після збирання предмету від потрапляє до інвентарю, який можна відкрити за допомогою відповідної кнопки. Деякі квести створюються саме за таким принципом – треба зібрати потрібну кількість предметів за певний час або після збирання перемістити їх у інше місце.

У іншій кімнаті розміщуємо стіну для руйнування, стіл, комп'ютер, персонажа та інші об'єкти. На ній буде знаходитися картина, з якою можна взаємодіяти. Взаємодія відбувається за допомогою платформи, що з'являється у іншій кімнаті. Із інших опцій гравця є:

- комбінування предметів;
- відкриття дверей;
- збирання головного предмету.

Опція комбінування предметів дає гравцю можливість створювати нові предмети, які потрібні для проходження рівня.

					КРБ.КІ.1.442-03.1.11	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		51

Опція відкриття дверей дає змогу пересуватися між кімнатами для вивчення локації та пошуку предметів.

Опція збирання предметів потрібна для того, щоб можна було виконати квест на збір предмету, який після виконання дає змогу просуватися далі по рівню.



Рис. 3.4 – Розміщення об'єктів на сцені

Іконки, що відображаються над персонажами та потрібними гравцю предметами, анімуються. Для цього створюється анімація по ключовим кадрам, де дана іконка трохи підіймається в гору та опускається. Цю анімацію ми створюємо для привернення уваги гравця до об'єкту, з яким йому треба взаємодіяти.

Після цього створюємо інші сцени подібними способами. Розміщуємо на них інші об'єкти.

Розробляємо елементи, потрібні для створення квестів. Також для них створюються анімації та потрібні скрипти.

					КРБ.КІ.1.442-03.1.11	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дат		

Загалом, проходження рівнів буде складатися з деяких завдань, які гравцю треба пройти. Під час їх виконання гравець може нові предмети у свій інвентар. Умови для отримання предметів та їх реалізація описується під час створення скриптів для гри. Квестові предмети також дають спеціальні можливості, якими гравець може користуватися, пересуваючись по ігровим кімнатам.



Рис. 3.5 – Розміщення об'єкту дверей

На деяких ігрових сценах також створюються об'єкти дверей, що допомагають персонажу пройти на інші кімнати. Іноді до них треба буде використати такий предмет, як ключ. Керування ними здійснюється через певні скрипти, у яких ми пишемо умови, за яких треба завантажувати ті чи інші сцени та локації.

					КРБ.КІ.1.442-03.1.11	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дат		

3.4 Написання основних скриптів

Поведінка ігрових об'єктів контролюється за допомогою компонентів, що приєднуються до них. Компоненти можуть бути різними, але вони не можуть описати весь потрібний функціонал у грі. Тому в *Unity* можна створювати скрипти для опису додаткових властивостей, взаємодії об'єктів, ігрових подій, зміни параметрів компонентів та іншого.

Скрипти для гри в *Unity* створюються на мові програмування *C#*. Писати та редагувати їх можна в *Visual Studio*. Такі скрипти взаємодіють з внутрішніми механізмами *Unity* за допомогою створення класів, що успадковуються від вбудованого класу *MonoBehavior*.

При створенні нового скрипту в ньому створюється дві функції: *Update* та *Start*.

Update потрібна для коду, який буде обробляти оновлення кадру для ігрового об'єкта. Це може бути рух, спрацювання дій та відповідна реакція на введення користувача, усе, що повинно бути оброблено за певний час в ігровому процесі. *Start* визивається до початку ігрового процесу, тобто, до першого виклику функції *Update*, слугує для ініціалізації.

Ім'я класу у скрипті береться з імені, яке ми вказуємо при створенні файлу. Ім'я класу та ім'я файлу повинні бути однаковими для того, щоб скриптовий компонент можна було приєднати до ігрового об'єкту.

Основні скрипти, що використовуються у грі перераховані далі.

Player – використовується для відтворення руху персонажа по сцені. За допомогою нього також вмикаються потрібні анімації персонажа у певний момент гри.

Скрипт контролює швидкість пересування персонажа, силу стрибка, поворот персонажа для руху у іншу сторону, взаємодію персонажа з іншими об'єктами.

Має такі методи:

					КРБ.КІ.1.442-03.1.11	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дат		

1. *Start()*.
2. *Update()*.
3. *Awake()*.

MouseLook – скрипт для руху камери за персонажем.

Створюється спеціально для керування камерою.

Має методи *Update()* та *Awake()*, що відтворюють переміщення камери за персонажем.

Scenes – скрип, що потрібен для керування сценами та переходами між ними. Використовує методи *ChScenes()* та *Exit()*.

endLevel – скрипт для завантаження наступного рівня. Має метод *OnCollisionEnter3D(Collision3D coll)*.

DoorController – скрипт для відкриття ігрових дверей. Після відкриття програє анімацію дверей, програє вказаний у *Audio Source* звук для відкриття дверей через функцію *audioSource.Play()*.

Має методи:

1. *Start()*.
2. *OnTriggerEnter3D(Collider3D collider)*.

Inventory – скрипт для інвентарю. За допомогою нього інвентар можна закривати та відкривати. Має методи *Start()* та *InventoryCheck()*. За допомогою інших скриптів в інвентар додаються предмети, які гравець збирає на сцені.

HandChecker1 – скрипт для керування об'єктами, збирання їх, додавання в інвентар.

Slot та *Spawn* – також є скриптами для управління інвентарем.

Slot має методи:

1. *Start()*.
2. *Update()*.
3. *DropItem()*.

Методи скрипту *Spawn*:

1. *Start()*.
2. *SpawnDroppedItem()*.

Quests – скрипт для реалізації квестів.

Має методи:

1. *OnTriggerEnter3D(Collider 3D other)*.
2. *CheckQuest()*.

QuestWall – також потрібен для реалізації деяких квестів. Програє анімацію падіння стіни, коли гравець виконує потрібну дію, та дає гравцю змогу пройти далі.

Має методи:

1. *Start()*.
2. *OnTriggerEnter3D(Collider 2D other)*.

Wardrobe – скрипт для взаємодії з об'єктом шафи. За допомогою методів *OnTriggerEnter3D(Collider3D other)* та *OnTriggerExit3D(Collider3D other)* гравець має можливість взаємодіяти із шафою після наближення до неї.

Raycast – скрипт для відтворення механік миши у грі.

RaycastManager – існує для керування механіками та взаємодією з іншими персонажами.

Має такі методи:

1. *Start()*.
2. *StartRaycast()*.
3. *DisplayNextObject()*.
4. *TypeRaycastText(string raycastText)*.
5. *EndRaycast()*.

RaycastAnimator – скрипт для підсвічування предметів. Потрібен для того, щоб предмети для взаємодій підсвічувались тільки тоді, коли герой наводить курсор поруч з квестовим предметом.

Використовується в основному для спеціальних предметів, як частини ключу, шмати молоту, частини головного предмети та так далі. Реалізується за допомогою завантажених шейдерів для підсвічування предметів. Керується також у скрипті *RaycastAnimator*.

3.5 Додавання аудіоефектів

Для роботи з аудіо в *Unity* є компонент *Audio Source*, тобто джерело звуку, за допомогою якого можна відтворювати аудіоефекти на сцені. У такого компонента є властивість програвати аудіо з 3D ефектом. Аудіо вмикається у певний момент та стає тише або гучніше, в залежності від відстані до якогось об'єкту.

Властивості звуку, які можна редагувати у компоненті *Audio Source*: зациклення звуку, гучність, висота, різні ефекти та можливість вимкнути його за необхідністю.

Audio Listener – компонент, який слухає аудіо, що програвється на сцені та передає його на динаміки комп'ютера. Для правильного функціонування цього компонента він повинен бути тільки один на сцені. За замовчуванням цей компонент підключений до основної камери.

Audio Listener та *Audio Source* можна редагувати за допомогою *Audio Filters*. Такі компоненти можуть змінювати реверберацію, фільтрацію, різні спотворення звуку, кожен фільтр представляє собою окремий компонент зі значеннями налаштувань звуку.

Для власної гри застосовуємо музику для головної теми, деяких рівнів, головного меню. Різні звуки застосовуються при використанні інструментів користувацького інтерфейсу, при виконанні квестів, отриманні нагород та при перемогах. Звуки самого пересування персонажу не будуть використовуватися, тому що у даній грі це буде заважати геймплею.

Для програвання музики головної теми додаємо файл з відповідним аудіо до головної сцени, а саме у компонент *Audio Source*, він буде відображатися у полі *Audio Clip*. Тепер редагуємо його звук у компоненті, особливу увагу приділяємо гучності звуку.

Інші музичні файли аналогічно додаються до інших ігрових сцен та рівнів. Також налаштуємо ці звуки для приємного звучання та обов'язково робимо їх зациклення, щоб музика не переривалась під час проходження рівня.

					КРБ.КІ.1.442-03.1.11	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дат		

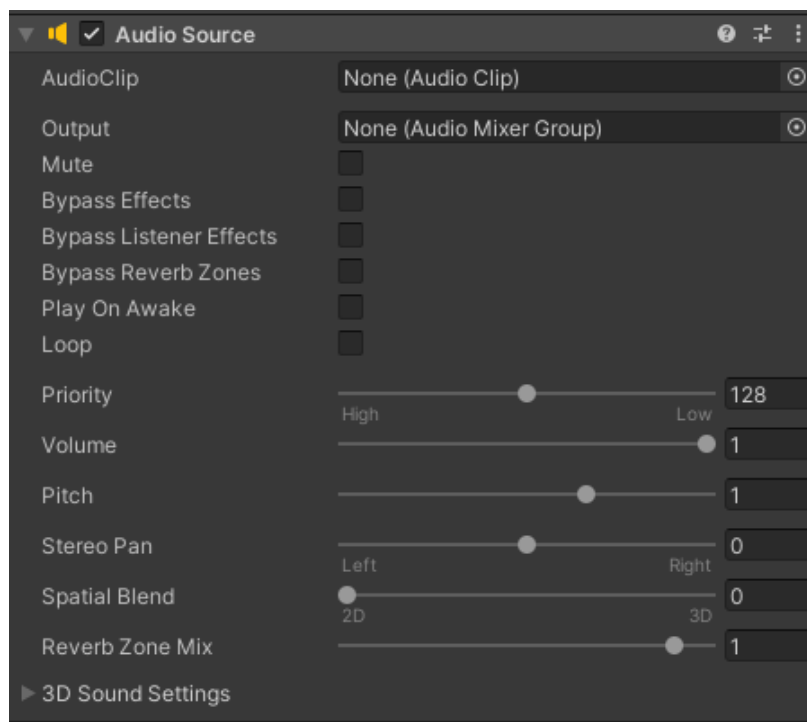


Рис. 3.6 – Компонент для програмування аудіо

Для відтворення інших звуків, таких як звук для збирання предметів та відкриття дверей, також додаємо *Audio Source* до потрібного об'єкта. Такий звук не повинен відтворюватися одразу під час завантаження сцени, на якій об'єкт знаходиться, а тільки, коли наш персонаж активує його. Тому пишемо спеціальний скрипт для збирання предметів, у якому за допомогою функції *audioSource.Play()* вмикаємо цей звук.

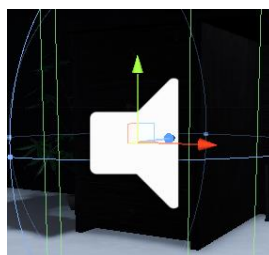


Рис. 3.7 – Додавання звуку до предметів

Головне меню також застосовують джерела звуку. Звук програватиметься при натисканні на такі кнопки, як: *Start*, *Exit*, кнопки вибору рівнів та інші.

Головне меню має відмінну від основної музику, що вмикається одразу після переходу на сцену з головним меню.

3.6 Створення інтерфейсу користувача та системи управління грою

Інтерфейс користувача застосовується для зручної взаємодії користувача з інформаційною системою. У розробці ігор грає дуже важливу роль, тому що це частина ігрової механіки, саме завдяки якій гравець може здійснювати різні дії, контролювати свій ігровий процес. Інтерфейс користувача складається з кнопок, вікон, вкладок та інших варіантів меню.

Меню та увесь інтерфейс користувача повинен бути зрозумілим та зручним для гравця. Він також повинен бути інтуїтивним, для чого зазвичай кнопки «прийняти», «так» та інші мають зелений колір, а такі кнопки як «відхилити», «ні» та інші схожі мають навпаки красний колір. Це створюється спеціально для моментального осмислення гравцем того, яку саме кнопку їм потрібно обрати.

Інтерфейс повинен показувати тільки потрібну інформацію, усі довгі тексти з описами квестів, завдань та іншого не є важливими, вони повинні бути чіткими та стислими.

Основні елементи *UI* в *Unity*:

- *canvas*;
- *panel*;
- *text*;
- *image*;
- *button*;
- *toggle*;
- *slider*;
- *scrollbar*;
- *dropdown*;
- *input field*.

Під час розробки нашої гри застосовуємо більшість з них.

На початку розробки інтерфейсу створюємо панель для головного меню, вона містить:

					КРБ.КІ.1.442-03.1.11	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		59

1. Кнопку «*Start*» для початку гри.
2. Кнопку «*Endings*» для переходу на панель отриманих кінцівок.
3. Кнопку «*Delete Savings*» для видалення ігрового процесу.

«*Endings*» дозволяє слідкувати, які кінцівки отримав гравець на різних рівнях. «*Start*» починає гру з початку, тобто з першого рівня.

Головна панель також містить кнопки для вимкнення або увімкнення звуку, налаштувань гри або кнопки для виходу зі гри. При на цю кнопку гравець закінчує гру.

Панелі «*Endings*» та «*Delete Savings*» створюються у тій же сцені, що і головне меню, але на початок гри є вимкненими та включаються лише, якщо гравець натискає відповідну кнопку. Такий функціонал в *Unity* відтворюється за допомогою події *OnClick()*.



Рис. 3.8 – Головне меню гри

З панелі «Start» перехід на рівні здійснюється за допомогою натискання. Для цього створюється скрипт «Scenes», за допомогою якого ми із коду завантажуюмо потрібну сцену.

Окрім головного меню та інших панелей у грі також є панель «Endings», на ній відображаються зібрані та отримані гравцем кінцівки гри. Відкриття цієї панелі здійснюється за допомогою натискання на відповідну кнопку під час ігрового процесу.

Також створюються панелі та кнопки для взаємодії з предметами у інвентарю, за допомогою яких можна пересувати предмети у інтерфес комбінації. Вони комбінуються за принципами відповідного скрипту для комбінацій.

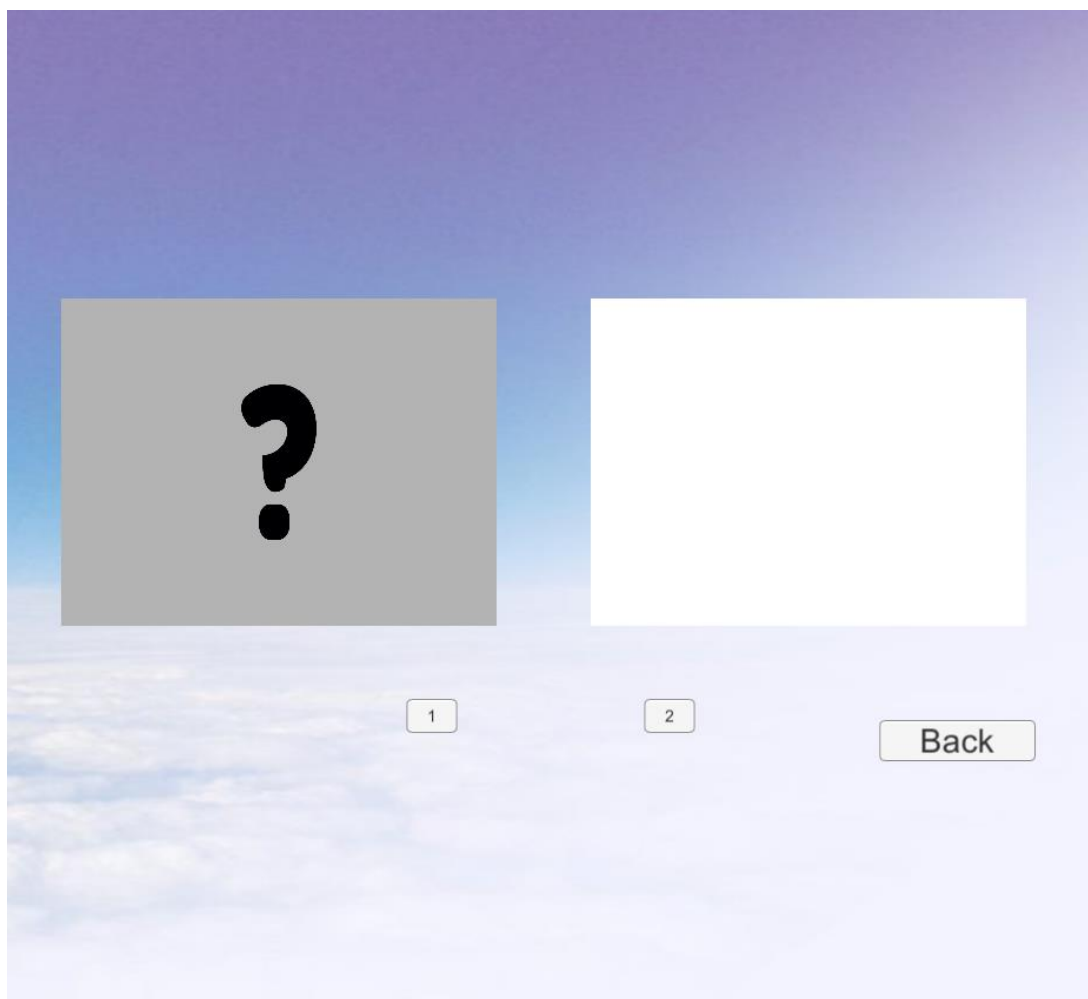


Рис. 3.9 – Панель для закінчення гри

Система керування у грі також є дуже важливою. Більшість комп'ютерних ігор застосовують схожі способи керування. Наприклад, самим розповсюдженим способом керування пересуванням персонажу є використання клавіш *W*, *A*, *S* та *D*, або стрілок.

У розроблюваній грі основними клавішами керування є:

A, *D* або стрілки вперед та назад – пересування персонажу.

Ctrl – присяд персонажу.

Escape – поставити на паузу.

За рахунок того, що гра є квестом, основні дії можна здійснювати за допомогою комп'ютерної миші та клавіатури.

3.7 Тестування гри

Для гарантії якості та забезпечення створення інноваційних, цікавих та інтуїтивно зрозумілих ігор існують різні види та методи тестування ігор. Зазвичай оцінюються такі ризики:

1. Чи приваблює ігровий додаток користувачів?
2. Чи є у додатку цікаві особливості?
3. Сама зручність дизайну додатку.
4. Наскільки легко доступними є ігрові особливості?
5. Якість медіафайлів (зображення, текстів, графіки, відео, аудіо та інших).

Основна мета тестування – це виявлення багів у грі та позбавлення від цих багів.

Одними з популярних видів тестування є:

1. Функціональне тестування. Полягає у перевірці працездатності додатку у відповідності до заданих специфікацій.
2. Комбінаторне тестування. Існує для аналізу продуктивності ігрового додатку. Допомагає виявити помилки на ранньому етапі.

					КРБ.КІ.1.442-03.1.11	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дат		

3. Дослідницьке тестування. При такому виді уся систему тестують у процесі її роботи. Це сприяє кращій працездатності.
4. Тестування сумісності. Виявляє проблеми сумісності, забезпечує роботу додатку у відповідності до очікувань кінцевих користувачів.
5. Методологія *Cleanroom*. Допомогає виявити точні причини багів та помилок, робить ігровий додаток продуктивним та зменшує трудовитрати на тестування.
6. *Play*-тестування. При такому виді тестування команда *QA* грає за гравця та перевіряє гру на помилки.
7. Регресійне тестування. Повторно перевіряється уся функціональність додатку з нуля та виявляються нові помилки.

Для тестування власної гри перевіряємо такі її складові як: адаптивність, фізику об'єктів, звук, музику, виконення самих завдань та роботу інтерфейсу користувача.

Особливо важливою є адаптивність екрану. Гра потребує підтримку широкого спектру роздільної здатності екрану. У системі створення інтерфейсів *Unity* є багата кількість інструментів для реалізації таких можливостей.

Елементи інтерфейсу за замовчуванням прив'язані до центру батьківського об'єкта, це означає, що вони зберігають постійне зміщення відносно центру.

Для того, щоб елементи інтерфейсу не зміщувалися та були рівномірно розподілені використовуємо спеціальні елементи (*anchors*). Ці елементи надають можливість зафіксувати об'єкт на місці та не змінювати його положення при різних варіантах співвідношення сторін. Також *Unity* дає можливість вибрати варіант співвідношення сторін та подивитися як виглядає гра на екранах з різним розширенням.

					КРБ.КІ.1.442-03.1.11	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дат		

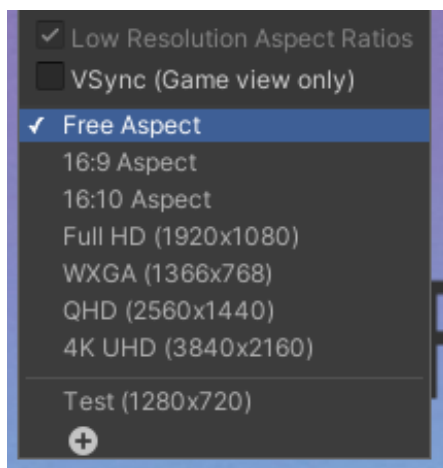


Рис. 3.10 – Варіанти вибору співвідношення сторін

Далі перевіряємо гру на наявність проблем з геймплеєм, пересуванням персонажу та програванням анімацій. Перевіряємо відтворення усіх звуків у грі, переконуємось у тому, що вони працюють правильно та вмикаються лише у потрібний момент.

Для руху персонажа та його взаємодії з іншими об'єктами перевіряємо гру на наявність усіх потрібних елементів, що відповідають за фізику (*Rigidbody 3D*, *Box Collider 3D* та інші).

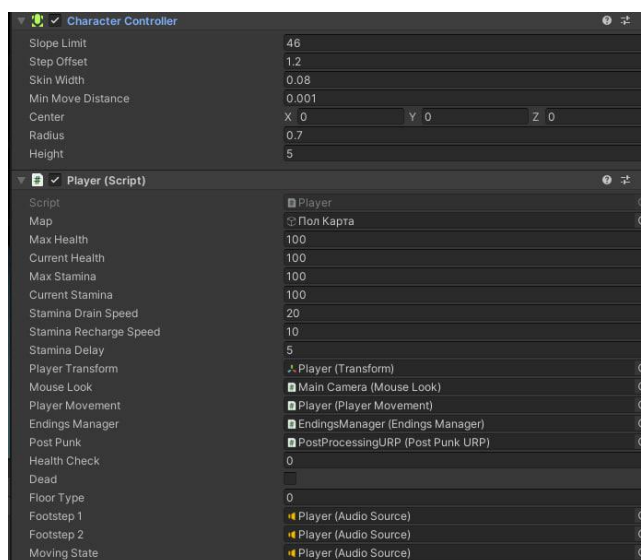


Рис. 3.11 – Налаштування головного персонажу

Під час фінальних перевірок проходимо ігрові рівні та переконуємось у правильній роботі всіх ігрових елементів.

Ігровий процес першого рівня складається з лінійки інтуїтивних квестів. З'являючись у кімнаті, гравець після невеликого огляду помічає на підлозі записку, з якої випливає, що на одну зі стін приміщення потрібно використовувати молоток.

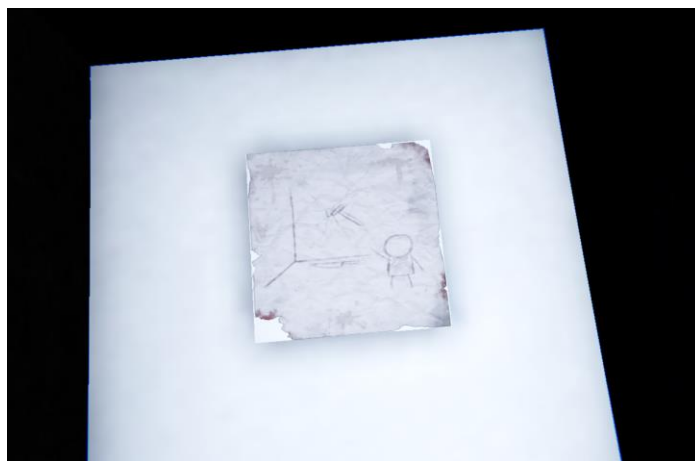


Рис. 3.12 – Підказка із стіною та молотом

У процесі подальшого вивчення локації гравець виходить або у вітальню, або на кухню, в обох випадках принагідно знаходячи шматки головного об'єкта. У вітальні гравець знаходить ліхтарик і шматок червоного ключа, насуваючи його на думку, що є друга половина.



Рис. 3.13 – Шмат червоного ключу

На кухні гравець знаходить місце для збирання предмета, знаходячи застосування шматками, які він знаходив раніше.

					КРБ.КІ.1.442-03.1.11	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дат		

Після цього, детально досліджуючи всю ігрову локацію, гравець знаходить червоні двері, зачинені на замок і здогадується, що для них потрібний червоний ключ.

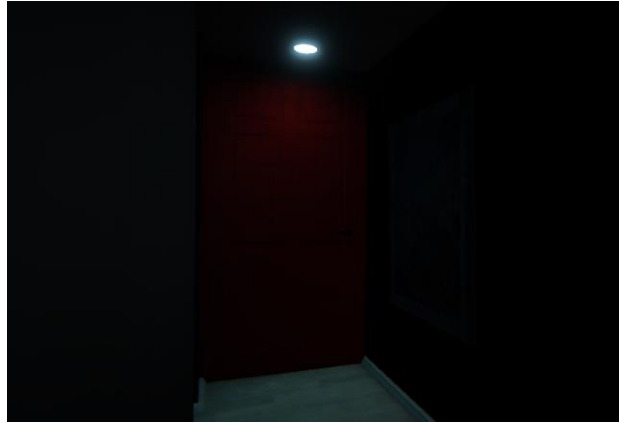


Рис. 3.14 – Червоні двері

Також гравець знаходить усі частини головного об'єкта, активація якого призводить до отримання предмета голову молота. Після цього уважний гравець помітить, що на кухні є натискна плита, яка ненадовго відсуває картину в сусідній кімнаті.



Рис. 3.15 – Натискна плита

За картиною знаходиться ручка молотка. Зібравши воедино цілий молоток, зіставивши його наявність із підказкою з початку гри, гравець розбиває потрібну стіну і знаходить потайну кімнату з другим шматком ключа.

Зробивши цілий ключ, гравець прямує до червоних дверей, відчиняє їх і вирушає на наступний рівень.

					КРБ.КІ.1.442-03.1.11	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дат		

Висновок до третього розділу

Перед розробкою гри було проаналізовано та вибрано програмний засіб для її створення. Було з'ясувано, що інструмент *Unity* є дуже зручним для створення тривимірних ігор, він має багато інструментів для їх створення, результати чого ми можемо бачити на прикладах одних із самих популярних ігор.

Під час розробки гри ми створили ігрові сцени для неї, розмістили моделі та ігрові об'єкти на сценах, створили для них анімації.

Було створено зручний інтерфейс користувача для гри та додано музику та інші звуки ігрового процесу. Для гри було створено спеціальні скрипти для пересування персонажа, різних його дій, квестів та відтворення поведінки ігрових об'єктів.

В результаті розробки ми отримали тривимірну гру з елементами квесту.

					КРБ.КІ.1.442-03.1.11	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Організаційно-економічне обґрунтування роботи

4.1.1 Актуальність та результати проекту

При виконанні дипломної роботи, була створена гра за допомогою середовища розробки *Unity3D* з використанням мови програмування *C#*.

По масштабу проект, що розробляється, відноситься до середніх проектів, які невеликі по масштабу, прості і обмежені об'ємами.

По термінах реалізації проект короткостроковий (термін реалізації до 3 років).

По складності проект, що розробляється, є проектом середньої складності.

По характеру цільового завдання – комбінований проект.

По характеру проекту – галузевий проект.

4.1.2 Огляд аналогів і конкурентів

Titan Souls

«Titan Souls» – це інді-гра, розроблена та випущена британськими розробниками відомими як Acid Nerve. Гра була вперше випущена в квітні 2015 року для Microsoft Windows, PlayStation 4 і PlayStation Vita[23]. Одна з основних особливостей «Titan Souls» полягає в її мінімалістичному підході. Гравець управляє молодим героєм, який має за завдання знищити великі титани величезним стрілом, яке може відновлювати. Кожен титан має свої унікальні характеристики і слабкі місця, і для їх знищення гравець повинен знаходити оптимальні стратегії.

					КРБ.КІ.1.442-03.1.11	Арк. 68
Змн.	Арк.	№ докум.	Підпис	Дат		

Гра відзначається високим рівнем виклику, оскільки помилка гравця може призвести до його швидкої смерті, але в той же час, вона надає гравцеві можливість вивчати і вдосконалювати свої навички.

Творчий дизайн, атмосферний саундтрек і нестандартний підхід до геймплею

роблять «Titan Souls» непересічною грою в своєму жанрі, і вона здобула позитивні відгуки від критиків за її унікальність та високу якість. Вплив серії «Dark Souls» виявляється в «Titan Souls» через його підхід до геймплею та структуру битв. Гра віддзеркалює елементи Soulslike, включаючи високий рівень важкості, критичні ситуації та потребу вивчення ворожих рухів для ефективного використання стріли. Це можна розглядати як ранній етап розвитку жанру Soulslike, де «Titan Souls» вперше впроваджує ці ідеї в новому контексті пригодницької гри.

Розробники Acid Nerve виявили креативність, створюючи унікальний геймплей і дизайн. Гра оцінюється за свою непересічність, творчий підхід до завдань та вміння надихати гравців на дослідження і пристосування до непередбачуваних ситуацій. «Titan Souls» – це свіжий погляд на пригодницький жанр, який відзначається своєю унікальністю та вибором новаторських механік гри.

Гра має такі оцінки на метакритик:

- ПК:74/100;
- PS4:73/100;
- VITA: 74/100.

DarkMaus

«DarkMaus» - це інді-гра в жанрі action RPG, яку створив і розробив Джон Ксав'єр. Гра була випущена в 2016 році для Microsoft Windows. «DarkMaus» отримала певне визнання за свій унікальний підхід до геймплею та високий рівень виклику[25].

					КРБ.КІ.1.442-03.1.11	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дат		

Основна особливість гри полягає в тому, що гравець управляє мишеням, яке вирушає в темний світ, наповнений ворогами та загадками. Гра має динамічну бойову систему, де гравець повинен уникати атак та використовувати стратегію для перемоги в сутичках.

Щодо фінансових питань, конкретні дані про бюджет та збори для «DarkMaus» можуть бути обмеженими. Як і у багатьох інших інді-проектах, бюджет може бути скромнішим порівняно з великими AAA-іграми, і збори можуть значно варіюватися залежно від розміру аудиторії та інших факторів.«DarkMaus» виходить за рамки стандартних ігор і виявляє великий вплив від культової серії «Dark Souls». Очевидно, що розробник, Джон Ксав'єр, взяв натхнення від естетики, атмосфери і геймплейних елементів «Dark Souls», щоб створити свою власну унікальну інтерпретацію в цьому інді-проекті.

Однак «DarkMaus» також має свої унікальні риси та особливості. Наприклад, відсутність класичного героя, який в більшості подібних ігор обирається гравцем, замінено керованим мишеням. Це створює додатковий вимір геймплею та призводить до несподіваних ситуацій і викликів.

Оцінка «DarkMaus» на Metacritic (83/100) та позитивний відгук від

Hardcore Gamer (4 бали з 5) свідчать про успіх гри серед критиків.

Lords of the Fallen

«Lords of the Fallen» представляє собою важливий твір у жанрі екшн-рольових ігор, що вийшов у 2014 році від німецької студії Deck13 Interactive та видавництва CI Games. Гра швидко здобула увагу геймерів своєю комбінацією високої складності, масштабної темної атмосфери і складної бойової системи[21].

Ігровий світ Lords of the Fallen вражає своєю містичною обстановкою і відтворенням темної фентезі. Він розкривається перед гравцем як загадковий і небезпечний, історія якого переплетена з темними силами і загадковим минулим.

					КРБ.КІ.1.442-03.1.11	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дат		

У цьому світі гравці беруть на себе роль героя, який вступає в боротьбу зі злими силами і величезними чудовиськами.

Однією з ключових особливостей гри є бойова система, яка вимагає від гравців не тільки майстерності у власному виборі зброї та обладунків, але й стратегічного мислення та уважності під час кожного бою. Lords of the Fallen пропонує розвиток персонажа, що дає гравцям можливість адаптувати свою стратегію до власного стилю гри.

Гра була натхнення та зазнала впливу інших успішних ігор жанру, зокрема серії «Dark Souls». Однак вона також вносить свої власні унікальні елементи та відзначається окремим стилем.

Інша очевидна джерельна точка натхнення - це темна фентезі література та мистецтво. Гра взяла з собою естетику, атмосферу і тематику, характерну для цього жанру, створюючи містичний і загадковий світ, який гравці можуть досліджувати.

Важливо також зазначити, що «Lords of the Fallen» була спроектована з урахуванням побажань гравців, які цінують виклик і глибину гри. Розробники ставили завдання створити гру з напруженим боєм та складною бойовою системою, а це, безумовно, було натхненням популярністю подібних елементів у грі «Dark Souls».

Інші впливи можуть включати різноманітні аспекти історії, міфології та культур, які призвели до формування унікального світу «Lords of the Fallen». У підсумку, гра представляє собою сполучення різноманітних джерел натхнення, щоб створити свій власний, захоплюючий геймплей. До травня 2015 року було продано понад 900 000 копій гри.

За загальноприйнятими оцінками, бюджет гри становив приблизно 12-14 мільйонів доларів. Проте це лише приблизна цифра, і реальний бюджет може виявитися іншим.

					КРБ.КІ.1.442-03.1.11	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дат		

Розробка гри такого масштабу, зокрема з урахуванням високих стандартів якості для графіки та геймплею, зазвичай вимагає значних фінансових витрат[22].

Гра має такі оцінки на метакритик:

- ПК: 73/100;
- PS4: 68/100;
- XONE: 71/100;
- iOS: 56/100.

Salt and Sanctuary

«Salt and Sanctuary» – це екшн-RPG від студії Ska Studios, яка була випущена в 2016 році для різних платформ, включаючи Microsoft Windows, PlayStation 4, PlayStation Vita, Nintendo Switch та Xbox One. Гра здобула популярність завдяки своєму унікальному стилю і глибокому геймплею[26]. Основна ідея гри полягає в тому, щоб гравець досліджував містику, виконував завдання, брав участь у боях з різними ворогами та босами і розвивав свого персонажа. Однак те, що відрізняє «Salt and Sanctuary», це його зосередженість на 2D-платформенних елементах, що надає грі свіжість у жанрі action RPG. Гра нерідко порівнюється з серією «Dark Souls» через свою атмосферу, важкий бій і систему втрати валюти (солі в «Salt and Sanctuary», відомої як душі в «Dark Souls») при смерті. Ця порівняльна аналогія вказує на великий вплив «Dark Souls» на дизайн ігри.

«Salt and Sanctuary» отримала позитивні відгуки за свій глибокий геймплей, мистецтво, атмосферу та відданість жанру. Гра завоювала певне місце серед фанатів важких ігор та залишається однією з визначних представниць незалежних action RPG.

					КРБ.КІ.1.442-03.1.11	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дат		

Гра має такі оцінки на метакритик:

- PS4:84/100;
- ПК:84/100;
- NS:82/100;
- XONE: 80/100.

4.1.3 Порівняльний аналіз

1. Жанр і Геймплей: – Lords of the Fallen: Екшн-рольова гра з важким боєм та глибокою бойовою системою; – Titan Souls: Мінімалістична інді-гра з високим рівнем виклику, де гравець знищує титанів стрілою; – DarkMaus: Action RPG, де гравець управляє мишеням і досліджує темний світ; – Salt and Sanctuary: Екшн-RPG з 2D-платформенними елементами, порівнюється з Dark Souls.

2. Впливи та Натхнення: – Lords of the Fallen: Натхнення від Dark Souls та темної фентезі літератури; – Titan Souls: Видимий вплив Dark Souls, особливо в елементах Soulslike; – DarkMaus: Сильний вплив Dark Souls, але з унікальним підходом та керуванням мишенем; – Salt and Sanctuary: Порівнюється з Dark Souls за атмосферою та боєм.

3. Фінансові аспекти: – Lords of the Fallen: Бюджет становив приблизно 12-14 мільйонів доларів. Продано понад 900 000 копій гри; – Titan Souls: Дані про бюджет і збори обмежені, але інді-проекти часто мають скромні бюджети; – DarkMaus: Конкретні дані про бюджет та збори також обмежені, але гра отримала певне визнання; – Salt and Sanctuary: Дані про бюджет обмежені, але гра отримала позитивні відгуки і велике визнання в жанрі.

4. Оцінки та Відгуки: – Lords of the Fallen: Оцінки на Metacritic: PC – 73/100, PS4 - 68/100, XONE- 71/100, iOS - 56/100; – Titan Souls: Оцінки на Metacritic: PC - 74/100, PS4 - 73/100, Vita - 74/100. Гра завоювала певне місце серед фанатів важких ігор та залишається визначною в жанрі.

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дат		

Висновок:

Кожна з цих ігор має свою унікальність та неповторність. Lords of the Fallen і Salt and Sanctuary прихильники Dark Souls знайдуть здайливими за їхніми елементами, в той час як Titan Souls та DarkMaus додають унікальність до жанру. Оцінки та фінансові показники свідчать про успіх кожної.

Таблиця 4.1

Порівняльний аналіз

	Lords of the Fallen	Titan Souls	DarkMaus	Salt and Sanctuary
Жанр і Геймплей	Екшн-рольова гра з важким боєм та глибокою бойовою системою:.	Мінімалістична інді-гра з високим рівнем виклику, де гравець знищує	Action RPG, де гравець досліджує темний світ	Екшн-RPG з 2D-платформенними елементами
Фінансові аспекти	Бюджет становив приблизно 12-14 мільйонів доларів. Продано понад 900 000 копій гри	Дані про бюджет і збори обмежені, але інді-проекти часто мають скромні бюджети	Конкретні дані про бюджет та збори обмежені, але гра отримала певне визнання	Дані про бюджет обмежені, але гра отримала позитивні відгуки і велике визнання в жанрі
Оцінки та Відгуки	Оцінки на Metacritic: PC - 73/100, PS4 - 68/100, XONE-71/100, iOS - 56/100;	Оцінки на Metacritic: PC - 74/100, PS4 - 73/100, Vita - 74/100. Гру відзначено за унікальність та високу якість.	Оцінка на Metacritic: 83/100. Гра отримала позитивний відгук від Hardcore Gamer (4 бали з 5)	Оцінки на Metacritic: PS4 - 84/100, PC - 84/100, NS - 82/100, Гра завоювала місце серед фанатів важких ігор.
Впливи та Натхнення	Натхнення від Dark Souls та темної фентезі літератури	Вплив Dark Souls, особливо в елементах Soulslike	Сильний вплив Dark Souls, але з унікальним підходом	Порівнюється з Dark Souls за атмосферою та боєм

4.2 Визначення мети і результатів роботи

Метою дипломної роботи є проектування та розробка 3D комп'ютерної гри. Для досягнення мети були проаналізовані сучасні тенденції у розробці комп'ютерних ігор, досліджені засоби розробки комп'ютерних ігор і проекти конкурентів.

Таблиця 4.2

Склад робіт по життєвому циклу проекту

№ робіт	Назва робіт	Тривалість робіт, дні
0-1	Вивчення предметної області	15
1-2	Вибір мови програмування та середовища розробки	5
2-3	Планування строків виконання необхідних робіт	8
3-4	Концептуальне проектування гри	10
4-5	Проектування ігрових механік	8
5-6	Написання скриптів(реалізація механік)	5
6-8	Створення рівнів	8
5-7	Створення алгоритму роботи користувальницького інтерфейсу	8
7-8	Реалізація користувацького інтерфейсу	6
6-8	Створення тестової версії	8
8-9	Тестування проекту в роботі	6
9-10	Виправлення помилок	2
10-11	Підготовка супровідної документації	2
11-12	Оцінка результатів проекту і підведення підсумків	2
12-13	Підготовка підсумкових документів і закриття проекту	2

Етапи і стадії роботи

№	Стадії	Зміст	Тривалість виконання
1	Технічне завдання	Постановка загальної ідеї ігри, техніко-економічне обґрунтування проекту, вибір мови програмування та середовища розробки.	Лютий
2	Технічний проект	Розробка рівнів та сетінгу. Розробка головного герою та ворогів, написання скриптів та сюжету гри.	Березень
3	Робочий проект	Програмування штучного інтелекту, налагодження ігрових механік та об'єктів на рівнях. Комплексне налагодження головного меню та підготовка до експлуатації.	Березень-квітень
4	Впровадження	Налагодження та виправлення помилок (bugs). Підготовка і передача дизайнерської документації для супроводу.	Травень

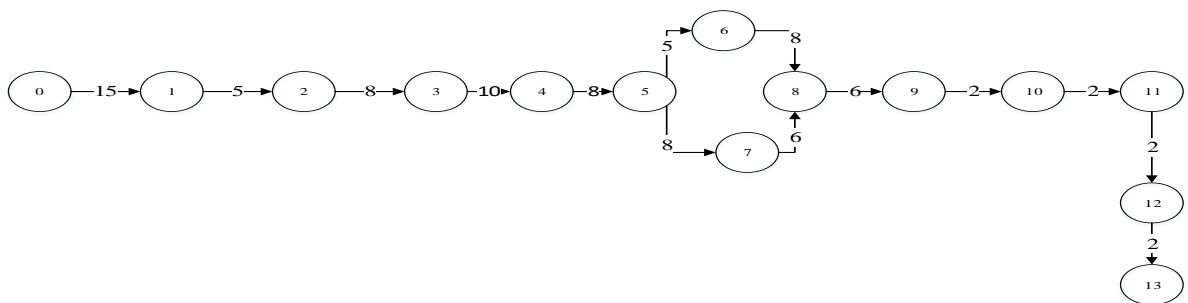


Рис. 4.1 – Мережевий графік проекту

Розрахунок параметрів мережевого графіка:

 T_{ij} – тривалість робіт; $T_{рн}$ – ранній початок робіт; $T_{ро}$ – раннє закінчення робіт; $T_{пн}$ – пізній початок робіт; $T_{по}$ – пізнє закінчення робіт; R_j – резерв часу.

Таблиця 4.4

Розрахунок параметрів мережевого графіку

Назва роботи	Попередня	T_{ij}	T_{pi}	T_{po}	T_{pi}	T_{no}	R_j
0-1	—————	15	0	15	5	5	0
1-2	0-1	5	15	20	23	28	8
2-3	1-2	8	20	28	28	36	8
3-4	2-3	10	28	38	36	46	8
4-5	3-4	8	38	46	38	46	0
5-6	4-5	5	46	51	47	52	1
6-8	5-6	8	51	59	52	60	1
5-7	4-5	8	46	54	46	54	0
7-8	5-7	6	54	60	54	60	0
8-9	6-8; 7-8	6	60	66	60	66	0
9-10	8-9	2	66	68	66	68	0
10-11	9-10	2	68	70	68	70	0
11-12	10-11	2	70	72	70	72	0
12-13	11-12	2	72	74	72	74	0

4.3 Розрахунки ціни програмного продукту

Як вихідні дані для визначення трудомісткості розробки ПП використовується типовий склад етапів і укрупнені норми часу на розробку програмних засобів (ПЗ) .

Розроблювальному ПП відповідає програма-аналог «MemoryGame» з обсягом 6000 умовних машинних команд і трудомісткістю $T_p = 210$ люд.-год.

Трудомісткість розробки ПП включає розробку наступних етапів:

- технічного завдання – ТЗ;
- технічного проекту – ТП;
- робочого проекту – РП;
- впровадження – ВН.

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

$$T_{ТЗ} = T_p \cdot L_1 \cdot K_H; \quad (4.1)$$

$$T_{ТП} = T_p \cdot L_2 \cdot K_H; \quad (4.2)$$

$$T_{рп} = T_p \cdot L_3 \cdot K_H \cdot K_T; \quad (4.3)$$

$$T_{ВН} = T_p \cdot L_4 \cdot K_H; \quad (4.4)$$

де T_p – укрупнена норма часу на розробку аналога ПП, люд.-годин, що коректуються поправочним коефіцієнтом, який враховує умови розробки ПП.

$$K_K = 0,8;$$

$$T_p = 210 \cdot 0,8 = 168 \text{ люд.-год.}$$

Даний проект має доступні аналоги, тому його можна віднести до ступеня новизни: В.

L_j – питома вага і-го етапу розробки в залежності від ступеня новизни:

$$L_1 = 0,12;$$

$$L_2 = 0,11;$$

$$L_3 = 0,61;$$

$$L_4 = 0,16.$$

K_H – поправочний коефіцієнт, який враховує ступінь новизни (0,7);

K_T – поправочний коефіцієнт, який враховує ступінь використання програм в розробці (0,7);

Тоді:

$$T_{ТЗ} = 168 \cdot 0,12 \cdot 0,7 = 14,11 \text{ (дні)}$$

$$T_{ТП} = 168 \cdot 0,11 \cdot 0,7 = 12,93 \text{ (дні)}$$

$$T_{рп} = 168 \cdot 0,61 \cdot 0,7 \cdot 0,7 = 50,21 \text{ (дні)}$$

$$T_{ВН} = 168 \cdot 0,16 \cdot 0,7 = 18,81 \text{ (дні)}$$

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дат		

Тривалість розробки ПП у літах визначається за формулою 4.5

$$T_{\text{ПП}} = T_{\text{тз}} + T_{\text{тп}} + T_{\text{рп}} + T_{\text{вн}} \quad (4.5)$$

де $T_{\text{ПП}}$ – сумарна тривалість розробки.

$$T_{\text{пп}} = 14,11 + 12,93 + 50,21 + 18,81 = 96 \text{ (дні)} = 0,26 \text{ (рік)}$$

За час розробки візьмемо середнє значення розрахованих показників тривалості проектування та розробки ПП:

$$T_{\text{ппсер}} = \frac{74 + 96}{2} = 85 \text{ (дні)}$$

Таблиця 4.5

Розрахунок витрат на матеріали

Найменування матеріальних витрат	Од. вим.	Кількість	Ціна за одиницю, грн.	Вартість, грн.
Диски	шт.	2	10	20
Папір	лист	200	0,24	48
Фарба (картридж)	шт.	1	58	58
Флеш-карта, 4 Gb	шт.	1	130	130
Разом				256
Транспортно-заготівельні витрати – 10%				256·0,10=25,6
Усього:				281

Розрахунок основної заробітної плати

Найменування робіт	Трудомісткість робіт у днях	Місячний оклад	Денна заробітна плата	Заробітна плата
1	2	3	4	5
1. Розробка ПП	85	6500	295	25075
2. Контроль керівника	20	10000	455	9100
Усього:	33	–	–	34175

Додаткова заробітна плата враховує оплату чергових відпусток, премії, інші доплати. Приймається в розрахунках 10% від основної.

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} \cdot 0,10 \quad (4.6)$$

$$ЗП_{\text{дод}} = 34175 \cdot 0,10 = 3418 \text{ грн.}$$

Єдиний соціальний внесок (Єсв) приймаються в розмірі 22% від суми основної і додаткової заробітної плати.

$$Єсв = (ЗП_{\text{осн}} + ЗП_{\text{дод}}) \cdot 0,22 \quad (4.7)$$

$$Єсв = (34175 + 3418) \cdot 0,22 = 8270 \text{ грн.}$$

Витрати, зв'язані з використанням обчислювальної техніки, визначаються:

$$C_{\text{еом}} = t^{\text{еом}} \cdot K_{\text{в}}^{\text{еом}} \cdot Ц^{\text{еом}} \cdot K_{\text{е}}^{\text{еом}}; \quad (4.8)$$

$T^{\text{свм}}$ – час використання ЕОМ для розробки даного ПП, години;

$T^{\text{свм}}$ – 181 годин.

$K^{\text{свм}}$ – поправочний коефіцієнт обліку часу використання ЕОМ (1,08);

$Ц^{\text{свм}}$ – ціна однієї години роботи на ЕОМ, грн. (5 грн.);

$K^{\text{свм}}$ – коефіцієнт обліку швидкодії ЕОМ (1,0 – швидкодія ЕОМ більш 20x10 опер/з; 1,2 – швидкодія ЕОМ менш 20x10 опер/з,)

$$C_{\text{еом}} = 181 \cdot 1,08 \cdot 5 \cdot 1,0 = 977 \text{ грн.}$$

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дат		

Накладні витрати враховують адміністративні, загальновиробничі витрати, витрати на збут. Приймаються в розмірі 50% від основної заробітної плати.

$$H_b = 0,30 \cdot 3P_{\text{осн}} \quad (4.9)$$

$$H_b = 0,50 \cdot 34175 = 17088 \text{ грн.}$$

На підставі здійснених розрахунків складається калькуляція планової собівартості ПП.

Таблиця 4.7

Калькуляція собівартості ПП

Найменування статей Витрат	Сума витрат (грн.)	Питома вага, %
1. Матеріали	281	0,4
2. Основна заробітна плата	34175	53
3. Додаткова заробітна плата	3418	5
4. Єдиний соціальний внесок	8270	13
5. Витрати, зв'язані з обчислювальною технікою	977	1,6
6. Накладні витрати	17088	27
Разом:	64209	100

Ціна ПП визначається по формулі:

$$Ц = С + P_p, \quad (4.10)$$

С – витрати на розробку програмної продукції (планова собівартість), грн.

P_p – розмір прибутку, розрахований по формулі:

$$P_p = С \cdot \%P_n / 100; \quad (4.11)$$

де: P_n – плановий рівень рентабельності (25%);

Прибуток у ціні ПП становить:

$$П = (64209 - 281) \cdot 0,25 = 15982 \text{ грн}$$

Ціна ПП становить: $Ц = 64209 \cdot 1,1 + 15982 = 86612 \text{ грн}$

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дат		

Для розрахунку економічної ефективності проекту визначаються капітальні і поточні витрати, зв'язані з використанням програмного продукту.

Розрахунок капітальних витрат, зв'язаних із впровадженням програмного продукту здійснюється по формулі:

$$K_2 = K_{пп} + K_{п} + K_{ко} + K_{во} \quad (4.12)$$

де: $K_{пп}$ – ціна програмного продукту;

$K_{п}$ – попередвиробничі витрати;

$K_{ко}$ – вартість комп'ютерного устаткування;

$K_{во}$ – вартість допоміжного устаткування, необхідного для надійної роботи грн;

Ціна програмного продукту складає 82640 грн.

Попередвиробничі витрати містять у собі усі витрати, зв'язані з налагодженням і впровадженням гри – постановка задач та їхня алгоритмізація, розробка, налагодження і впровадження програмного продукту (ПП).

Приймаються $K_{п}$ у розмірі 100% від вартості розробленого ПП.

$$K_{п} = 86612 \text{ грн.}$$

Вартість комп'ютера ($K_{ко}$) становить 15000 грн.

Вартість допоміжного устаткування визначається укрупнено в розмірі 10% від вартості комп'ютера.

$$K_{во} = 15000 \cdot 0,10 = 1500 \text{ грн.}$$

$$K_2 = 86612 + 15000 + 1500 = 103112 \text{ грн.}$$

Капітальні вкладення до впровадження гри були відсутні, тому $K_1 = 0$.

Розрахунок поточних (експлуатаційних) витрат, зв'язаних з використанням ПП (C_i), здійснюється по формулі:

$$C_i = C_{опл} + C_a + C_{ел} + C_p + C_{доп} + C_{п} \quad (4.13)$$

де $C_{опл}$ – річний фонд основної і додаткової оплати праці персоналу, що обслуговує програмний продукт з нарахуваннями;

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дат		

C_a – сума річних амортизаційних відрахувань від вартості основного і допоміжного устаткування програмного продукту;

$C_{ел}$ – вартість витрат на електроенергію в рік;

C_p – вартість річного ремонту основного і допоміжного устаткування;

$C_{доп}$ – річна вартість допоміжних матеріалів, зв'язаних з експлуатацією КМ.

$C_{п}$ – вартість річного утримання приміщень.

Під час проведення аналізу предметної області за базовий варіант було обрано діяльність компанії без використання спеціалізованої програми. Весь процес контролю виконують 2 співробітника, заробітна плата яких складає по 7800 грн.

Розрахунок річного фонду основної і додаткової оплати праці персоналу з нарахуванням ($C_{опл}$).

Річний фонд основної заробітної плати персоналу, що обслуговує ПП

$$ЗП_{осн} = \cdot (ЗП_{окі} \cdot Ч_i) \cdot 12 \quad (4.14)$$

де $Ч_i$ – чисельність, фахівців i -тої категорії, що обслуговують ПП;

$ЗП_{окі}$ – місячний оклад фахівця i -тої категорії;

$$ЗП_{осн} = (7800 \cdot 2) \cdot 12 = 187200 \text{ грн.}$$

Фонд додаткової заробітної плати:

$$З_{дод} = ЗП_{осн} \cdot K_{дод} \quad (4.15)$$

де $K_{дод}$ – коефіцієнт додаткової заробітної плати (приймається в розмірі $K_{дод} = 0,2$)

$$З_{дод} = 187200 \cdot 0,2 = 37440 \text{ грн.}$$

Єдиний соціальний внесок приймається в розмірі 22% від суми основної і додаткової заробітної плати.

$$Є_{св} = (ЗП_{осн} + ЗП_{дод}) \cdot 0,22$$

$$Є_{св} = (187200 + 37440) \cdot 0,22 = 49420 \text{ грн.}$$

					КРБ.КІ.1.442-03.1.11	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дат		

Загальні витрати на оплату праці:

$$C_{\text{опл1}} = 3П_{\text{осн}} + 3П_{\text{доп}} + \text{Єсв} = 187200 + 37440 + 49420 = 274060 \text{ грн.}$$

При відсутності інших витрат до впровадження ПП $C_1 = 274060$ грн.

Після впровадження ПП роботу виконує одна людина: фахівець з окладом 7400 грн.

Річний фонд основної заробітної плати фахівця з окладом 7400 грн. становить:

$$3П_{\text{осн}} = 7400 \cdot 12 = 88800 \text{ грн}$$

Фонд додаткової заробітної плати:

$$3П_{\text{доп}} = 88800 \cdot 0,20 = 17760 \text{ грн}$$

Єдиний соціальний внесок:

$$\text{Єсв} = (88800 + 17760) \cdot 0,22 = 23443 \text{ грн}$$

Загальні витрати на оплату праці:

$$C_{\text{опл2}} = 3П_{\text{осн}} + 3П_{\text{доп}} + \text{Відр.сс} = 88800 + 17760 + 23443 = 130003 \text{ грн}$$

Розрахунок амортизаційних відрахувань визначається по формулі:

де H_a – норма амортизаційних відрахувань (для комп'ютерних устаткувань $H_a = 50\%$)

$$C_a = (K_{\text{ко}} + K_{\text{во}}) \cdot H_a / 100 \quad (4.16)$$

$$C_a = (15000 + 1500) \cdot 0,5 = 8250 \text{ грн}$$

Річна вартість споживаної електроенергії $C_{\text{ел}}$, визначається по формулі:

$$C_{\text{ел}} = M_y \cdot T_{\text{ко}} \cdot C_e \cdot K_n \quad (4.17)$$

де M_y – установлена сумарна потужність комп'ютерного устаткування, (0,45кВт);

$T_{\text{ко}}$ – річний фонд часу роботи ЕОМ, який визначається виходячи з кількості робочих днів в році (D_p), тривалості робочого дня (T) і з урахуванням часу на профілактичні огляди за рік ($T_{\text{огл}}$)

$$T_{\text{ко}} = D_p \cdot T - T_{\text{огл}}$$

$$T_{\text{ко}} = 250 \cdot 8 - 300 = 1700 \text{ год.}$$

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дат		

C_e – вартість 1 квт-години ел. енергії (3,22 грн.);

$K_{и}$ – коефіцієнт інтенсивного використання потужності ($K_{и\text{рекком.}}=0,9$)

$$C_{ел} = 0,45 \cdot 1700 \cdot 3,22 \cdot 0,9 = 2217 \text{ грн.}$$

Витрати на ремонт (C_p) приймаються в розмірі 6% від вартості комп'ютерного устаткування:

$$C_p = 15000 \cdot 0,06 = 900 \text{ грн.}$$

Витрати на допоміжні матеріали ($C_{доп}$) приймаються в розмірі 2% від вартості комп'ютерного устаткування:

$$C_{доп} = 15000 \cdot 0,02 = 300 \text{ грн.}$$

Загальні витрати:

$$C_2 = 130003 + 8250 + 2217 + 900 + 300 = 141670 \text{ грн.}$$

Очікуваний економічний ефект визначається по формулі

$$E_o = (C_1 - C_2) - E_n \cdot (K_2 - K_1) \quad (4.18)$$

де C_1, C_2 – поточні витрати відповідно до і після впровадження проекту;

$(C_1 - C_2)$ – річна економія на поточних витратах, грн.;

K_2 – капітальні витрати на впровадження ПП, грн.;

K_1 – капітальні витрати до впровадження ПП, грн.;

E_n – нормативний коефіцієнт ефективності одноразових витрат (рекомендовано $E_n = 0,25$)

$$E_o = (274060 - 141670) - 0,25 \cdot 103112 = 132390 - 25778 = 106612 \text{ грн.}$$

Потім розраховується коефіцієнт ефективності капітальних витрат по формулі:

$$E = (C_1 - C_2) / K_2 - K_1 \quad (4.19)$$

$$E = 132390 / 103112 = 1,28$$

Так, як $E > E_n$, то проект ефективний.

Розраховується строк окупності капітальних витрат на впровадження проекту

$$T = 1/E = 1/1,28 = 0,78 \text{ року} = 9,4 \text{ міс} \quad (4.20)$$

Результати економічних розрахунків відображаються в підсумковій таблиці 4.9

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дат		

Техніко-економічні показники програмного продукту

№	Найменування показників	Одиниця Виміру	Значення показника	
			до впровадження проекту	після впровадження проекту
1	Трудомісткість розробки проекту	днів		85
2	Ціна ПП	грн.		86612
3	Капітальні витрати	грн.	—	103112
4	Поточні витрати	грн/рік	274060	141670
5	Економічний ефект від реалізації проекту	грн/рік		106612
6	Строк окупності	років		0,78
7	Економічна ефективність			1,28

4.4 Бізнес план стартапу проекту

Міні-бізнес план стартап-проекту для гри

Основна ціль проекту	Основною ціллю проекту стартапу розробки гри є досягнення успіху в гейміндустрії шляхом створення високоякісної, цікавої та популярної гри, яка буде привертати гравців та приносити прибуток.
Аналіз ринку	Цільова аудиторія — геймери від 18 років та цінителі квестів. Зростаючий інтерес до квестів у гейміндустрії. Монетизація через продаж гри, мікротранзакції, додатковий контент та рекламу.
Розробка продукту (геймплей, графіка)	Розробка гри орієнтована на реалістичність та захопливість. Гравець досліджує квартиру, знаходячи частини об'єкту та предмети для квестів. Графіка має високу якість, деталізовані моделі, реалістичні анімації та атмосферне освітлення. Дизайн локацій створює привабливе оточення, підсилюючи занурення. Реалістичні фізичні властивості предметів додають глибини та автентичності.
Маркетинг та реклама	Розробка веб-сайту для гри з інформацією, оновленнями та можливістю спілкування між гравцями. Створення сторінок у соціальних медіа для взаємодії з аудиторією. Розробка трейлеру та геймплейного відео для розміщення на YouTube і Twitch для привернення уваги геймерів.
Фінансові доходи	Основним джерелом доходу є продаж гри через платформи, такі як Steam, PlayStation Store та Xbox Live. Також можливі
Висновки	Проект має високий потенціал завдяки популярності квестів і зростаючому інтересу до гейміндустрії.

					КРБ.КІ.1.442-03.1.11	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		87

Висновок до четвертого розділу

Розробка даного проекту є вигідною, бо має швидко окупність і перспективи додаткового прибутку у майбутньому. Розробка ігор є прибутковим бізнесом, попит на який з кожним роком усе зростає. Даний проект має великі перспективи за рахунок популярного жанру, знайомих широкому користувачу механік цікавому сеттингу. Також проект може бути легко адаптований для усіх платформ й має необмежену перспективу росту.

Порівняння коефіцієнту ефективності та одноразових витрат дозволяє зробити висновок, що впровадження даного продукту є економічно вигідним, тому розробка цього програмного продукту є актуальною. Трудомісткість проекту становить 85 днів; строк окупності проекту – 0,78 року (9місяців); капітальні витрати становлять 103112 грн. Ціна програмного продукту – 86612 грн

					КРБ.КІ.1.442-03.1.11	Арк.
						88
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 5 ОХОРОНА ПРАЦІ

5.1 Основні положення охорони праці

Охорона праці – це система збереження життя та здоров'я робітників, що знаходяться у процесі трудової діяльності. Це включає в себе правові, соціально-економічні, організаційні, технічні, санітарно-гігієнічні, реабілітаційні та інші заходи. Саме поняття охорони праці визначається у статті першій Закону України «Про охорону праці».

Склад Законодавства України про охорону праці:

1. Закон «Про охорону праці».
2. «Кодекс законів про працю України».
3. Закон «Про забезпечення санітарного та епідемічного благополуччя населення».
4. Закон України «Про пожежну безпеку».
5. «Норми радіаційної небезпеки України (НРБУ-97)».
6. Інші нормативно-правові акти, що регулюють взаємовідносини між суб'єктами права у сфері охорони праці.

Закон «Про охорону праці» визначає основні положення щодо реалізації конституційного права громадян на охорону їхнього життя і здоров'я в процесі трудової діяльності, також регулює, за участю відповідних державних органів, відносини між власником підприємства, установи, організації або уповноваженим ним органом і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації праці в Україні.

Політика з охорони праці базується на таких принципах:

- пріоритет життя та здоров'я працівників, відповідальність роботодавця за створення безпечних умов праці;

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						89
Змн.	Арк.	№ докум.	Підпис	Дат		

- підвищення рівня безпеки на підприємстві шляхом суцільного технічного контролю стана виробництв, технологій, продукції, а також сприяння підприємствам у створенні безпечних умов праці;
- комплексне розв'язання завдань охорони праці на основі загальнодержавної, галузевих, регіональних програм з цього питання;
- соціальний захист працівників, повне відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві та від професійних захворювань;
- встановлення єдиних вимог з охорони праці для всіх підприємств та суб'єктів підприємницької діяльності, незалежно від форм власності та видів діяльності;
- використання економічних методів управління охороною праці, участі держави у фінансуванні заходів щодо охорони праці, залучення добровільних внесків та інших надходжень на ці цілі;
- адаптація трудових процесів до можливостей працівника з урахуванням його здоров'я та психологічного стану;
- інформування населення, проведення навчання, професійної підготовки і підвищення кваліфікації працівників з питань охорони праці;
- забезпечення координації діяльності органів державної влади, установ, організацій, об'єднань громадян, що розв'язують проблеми охорони здоров'я;
- використання світового досвіду організації роботи щодо поліпшення умов і підвищення безпеки праці на основі міжнародного співробітництва.

Статтю 14 ЗУ «Про охорону праці» передбачено такі обов'язки працівника:

1. Дбати про власну безпеку, про безпеку сторонніх людей при виконанні робіт чи під час перебування на території підприємства.

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дат		

2. Користування засобами колективного та індивідуального захисту.
3. Треба знати і виконувати вимоги нормативно-правових актів з охорони праці, правила поведження з устаткуванням та іншими засобами виробництва.
4. Проходити періодичні медичні огляди, навчальні курси, інструктажі, атестацію знань з безпеки праці.

5.2 Правила охорони праці при роботі з комп'ютером

Розповсюдження персональних комп'ютерів призвело до загострення проблем збереження власного та суспільного здоров'я. Це потребувало удосконалення існуючих та розроблення нових підходів до організації робочих місць, проведення профілактичних заходів для запобігання розвитку негативних наслідків впливу використання персональних комп'ютерів на здоров'я користувачів.

Охорона праці у випадку користувачів ПК розглядають у трьох основних аспектах: соціальний, психологічний та медичний аспект.

Соціальний аспект пов'язаний з розв'язання проблем по оптимізації умов життя, праці, відпочинку, харчування, побуту, розвитку культури, транспорту.

Медицина складає значну роль у профілактиці захворювань користувачів ПК. Існує перелік профілактичних заходів для користувачів ПК, що включає як складові первинної профілактики здоров'я, так і вторинної, яка направлена на зниження ймовірності розвитку перевтоми та перенапруження. Ці комплексні заходи спрямовані на відновлення функціонального стану зорового та опорно-рухового апарату.

У нашій країні розроблені національні нормативні документи, що спрямовані на охорону праці користувачів ПК.

Найбільш повним нормативним документом щодо забезпечення охорони праці користувачів ПК є "Державні санітарні правила і норми роботи

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дат		

з візуальними дисплейними терміналами (ВДТ) електронно-обчислювальних машин".

Даний документ складається з таких розділів:

1. Загальні положення. Розділ вказує, на кого поширюються самі правила, особлива увага звертається на те що дотримання вимог, що викладені в правилах, значно знизить наслідки несприятливої дії на працівників шкідливих та небезпечних факторів, які супроводжують роботу з відео-дисплейними терміналами. Це стосується зорових та нервово-емоційних перевантажень, серцево-судинних захворювань.
2. Вимоги до виробничих приміщень для експлуатації ВДТ. У розділі наведена характеристика приміщень, де експлуатуються ВДТ. Наведені їх параметри. Розміщення робочих місць з ВДТ ЕОМ у підвальних приміщеннях, на цокольних поверхах заборонено. Площа на одне робоче місце становить не менше ніж 6,0 м², а об'єм - не менше ніж 20,0 м³. Приміщення для роботи з ВДТ повинні мати природне та штучне освітлення відповідно до СНиП И-4-79. Виробничі приміщення повинні обладнуватись шафами для зберігання документів, магнітних дисків з урахуванням вимог до площі приміщень. У приміщеннях з ВДТ слід щоденно робити вологе прибирання. Приміщення з ВДТ мають бути оснащені аптечками першої медичної допомоги.
3. Гігієнічні вимоги до параметрів виробничого середовища приміщень з ВДТ. Це вимоги до параметрів мікроклімату, освітлення, шуму і вібрації, рівнів електромагнітного та іонізуючого випромінювання. У виробничих приміщеннях на робочих місцях з ВДТ мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря.

					КРБ.КІ.1.442-03.1.11	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дат		

Норми мікроклімату для приміщень з ВДТ ЕОМ

Пора року	Категорія робіт	Температура повітря, °С, не більше	Відносна вологість повітря, %	Швидкість руху повітря, м/с
Холодна	легка – 1а	22-24	40-60	0,1
	легка – 1б	21-23	40-60	0,1
Тепла	легка – 1а	23-25	40-60	0,1
	легка – 1б	22-24	40-60	0,2

До категорії 1а належать роботи, що виконуються сидячи і не потребують фізичного напруження, витрати енергії складають до 139 Вт.

До категорії 1б належать роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням, при яких витрати енергії становлять від 140 до 174 Вт.

Штучне освітлення в приміщеннях з робочими місцями, обладнаними ВДТ має здійснюватися за допомогою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, у разі переважної роботи з документами, допускається застосування системи комбінованого освітлення (крім системи загального освітлення, додатково встановлюються світильники місцевого освітлення).

Значення освітленості на поверхні робочого столу в зоні розміщення документів має становити 300-500 лк. Якщо це неможливо забезпечити системою загального освітлення, допускається використовувати місцеве освітлення.

Рівні звукового тиску в октавних смугах частот, рівні звуку та еквівалентні рівні звуку на робочих місцях, обладнаних ВДТ, мають відповідати вимогам СН 3223-85, ГОСТ 12.1.003-83, ГР 2411-81.

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
						93
Змн.	Арк.	№ докум.	Підпис	Дат		

Таблиця 5.2

Допустимі рівні звуку, еквівалентні рівні звуку і рівні звукового тиску
в октавних смугах частот

Вид діяльності	Рівні звукового тиску в дБ в октавних смугах із середньгеометричними частотами, Гц									
	31,5	63	125	250	500	1000	2000	4000	8000	Еквівалентні рівні звуку, дБАекв
Програмісти ЕОМ	86	71	61	54	49	45	42	40	38	50
Оператори ЕОМ	96	83	74	68	63	60	57	55	54	65
В приміщеннях шумних агрегатів ЕОМ	103	91	83	77	73	70	68	66	64	75

На користувачів під час роботи з комп'ютерною технікою можуть діяти такі види небезпек:

- ураження електричним струмом;
- енергетична небезпека (виникає через коротке замикання: опіки, електрична дуга, викид розплавленого металу);
- небезпека загоряння;
- термонебезпека (дія високих температур через нагрівання конструктивних елементів);
- механічна небезпека (травми через падіння, дію рухомих частин, поріз за гострі частини конструктивних елементів);

- небезпека випромінювання (дія звукового (акустичного), високочастотного, інфрачервоного, ультрафіолетового й іонізуючого випромінювання, а також видимого світла когерентної високої інтенсивності (лазерного випромінювання));
- хімічна небезпека (контакт із деякими хімікатами, які використовують для того, щоб обслуговувати обладнання, або від вдихання їх парів).

Висновок до п'ятого розділу

Під час вивчення правил охорони праці було оглянуто її основні положення та правила охорони праці при роботі з комп'ютером. Оглядаючи основні положення, ми дізнались про діючі закони щодо збереження життя та здоров'я працівників, вивчили політику з охорони праці та основні обов'язки працівника. Основні документи щодо забезпечення охорони праці користувачів ПК свідчать про дотримання вимог до виробничих приміщень, освітлення, звуку у них та самої площі приміщення. Також було освоєно види небезпек, які можуть діяти на користувачів персональних комп'ютерів, такі як: енергетична, хімічна, механічна небезпека, ураження електричним струмом та інші види небезпек.

					<i>КРБ.КІ.1.442-03.1.11</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		95

ЗАГАЛЬНІ ВИСНОВКИ

Основними результатами виконання роботи є:

1. За допомогою детального огляду предметної області було проаналізовано основні властивості комп'ютерних ігор, а саме ігор жанру квест.

2. Під час проектування власної гри було отримано багато нової інформації про розробку моделей, створення текстур та анімацій, та процеси проектування рівнів у грі.

3. В результаті розробки ми отримали тривимірну гру з елементами квесту.

4. Було також проаналізовано усі витрати на створення даної гри. Економічні розрахунки показали ефективність нашого проекту, а також саму ціну розробки. Розробка даного проекту є вигідною, бо має швидку окупність і перспективи додаткового прибутку у майбутньому. Розробка ігор є прибутковим бізнесом, попит на який з кожним роком усе зростає.

5. Під час вивчення правил охорони праці було оглянуто її основні положення та правила охорони праці при роботі з комп'ютером. (розроблено ігрові рівні та сцени, скрипти для взаємодій персонажа з об'єктами та інтерфейс користувача).

Після створення гри було проведено її тестування, а також проаналізовано основні положення охорони праці під час роботи з комп'ютером.

					КРБ.КІ.1.442-03.1.11	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дат		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Halpern J. *Developing 3D Games with Unity*. Berkeley, CA : Apress, 2019. – 405 с.
2. Lanzinger F. *Programming C# in Unity. 3D Game Development with Unity*. 2020. С. 17–39.
3. Lanzinger F. *2D Tools in Unity. 3D Game Development with Unity*. 2020. С. 161–177.
4. Lanzinger F. *Menus. 3D Game Development with Unity*. 2020. С. 205–223.
5. Lanzinger F. *The Unity Interface. 3D Game Development with Unity*. 2020. С. 75–94.
6. Lanzinger F. *Animating the Player Character. 3D Game Development with Unity*. 2020. С. 225–240.
7. Lanzinger F. *Lives, Level Design, and Old School Scoring. 3D Game Development with Unity*. 2020. С. 365–383.
8. Lanzinger F. *Testing. 3D Game Development with Unity*. 2020. С. 399–401.
9. Thorn A. *Getting Started. Pro Unity Game Development with C#*. Berkeley, CA, 2014. С. 37–80.
10. Troelsen A., Japikse P. *Working with Interfaces. Pro C# 9 with .NET 5*. Berkeley, CA, 2021. С. 297–339.
11. *Unity Asset Store* [Електронний ресурс]. – Режим доступу: <https://assetstore.unity.com/>.
12. Історія комп'ютерних ігор [Електронний ресурс]. – Режим доступу: <https://sites.google.com/site/komputerniigrividadoa/>.
13. Різниця між 2D і 3D [Електронний ресурс]. – Режим доступу: <https://www.avktarget.com/articles/tehnologii/>.
14. Ігри-квести на ПК [Електронний ресурс]. – Режим доступу: <https://hi-news.pp.ua/>.

					КРБ.КІ.1.442-03.1.11	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		97

15. Проектування комп'ютерних ігор для навчання : навчальний підручник / Т.А. Лугова, О.А. Блажко. – Одеса : ФОП «Побута». – 2018. – 212 с.
16. *Getting Started with Unity for Creating a 3D Game* [Електронний ресурс]. – Режим доступу: <https://www.mongodb.com/developer/languages/csharp/getting-started-unity-creating-3d-game/>.
17. *Setting up a 2D Scene in Unity* [Електронний ресурс]. – Режим доступу: <https://eledris.com/setting-up-a-3d-scene-in-unity/>.
18. *Scenes – Unity Documentation* [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/CreatingScenes.html>.
19. *GameObjects – Unity Documentation* [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/GameObjects.html>.
20. *Physics – Unity Documentation* [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/PhysicsSection.html>.