

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



**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему Проектування та розробка гри-платформера
(назва кваліфікаційної роботи згідно наказу ОНТУ)
"Manticora's Travel"

Здобувача Бурячека В.С.
(прізвище, ініціали)
4 курсу 543б групи
Керівник: д.т.н., проф. Артеменко С.В.
(посада, прізвище та ініціали)
ст. викл. Слушна Н.В.
(посада, прізвище та ініціали)
Консультанти д.е.н., проф. Басюркіна Н.Й.
(посада, прізвище та ініціали)

(посада, прізвище та ініціали)

Кваліфікаційна робота допускається до захисту

Рішення кафедри від 10.06 2023 р., протокол № 8

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

Одеса - 2023 рік

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
« 10 » серпня 2022 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Бурячека Вячеслава Сергійовича

1. Тема роботи Проектування та розробка гри-платформера "Manticora's Travel"

Затверджена наказом університету від « 10 » квітня 2023р., наказ № 147-03

2 Термін здачі здобувачем закінченої роботи 5 червня 2023 р.

3. Вихідні дані роботи

1. Середовище програмування «Visual Studio».

2. Двигун для розробки двовимірних та тривимірних додатків та ігор «Unity».

3. Ассети персонажів. 4. Відеоредактор MS MovieMaker

4. Перелік питань, які потрібно розробити

1. Вступна частина. 2. Аналіз предметної області.

3. Проектування гри. 4. Розробка гри-платформера.

5. Економічні розрахунки. 6. Охорона праці.

7. Загальні висновки.

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

Слайд 1. Тема роботи. Слайд 2. Мета. Слайд 3. Постановка задачі. Слайд 4. Ігровий жанр. Слайд 5. Аналоги програмного продукту. Слайд 6. Засоби реалізації.

Слайд 7. Схема бази даних. Слайд 8. Графічна реалізація проекту. Слайд 9. Рівні з гри. Слайд 10. Техіко-економічні показники. Слайд 11. Висновки.

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Економіка</i>	<i>Басюркіна Н.Й., д.е.н., проф.</i>		
<i>Охорона праці</i>	<i>Артеменко С.В., д.т.н., проф.</i>		
<i>Нормоконтроль</i>	<i>Жуковецька С.Л., ст. викл.</i>		

7. Дата видачі завдання 30.09.2022

Керівники _____

Сергій АРТЕМЕНКО

Наталія СЛУШНА

Завдання прийняв до виконання _____

Вячеслав БУРЯЧЕК

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	<i>Дослідження предметної області.</i>	15.02.2023	
2.	<i>Дослідження і аналіз існуючих аналогів.</i>	01.03.2023	
3.	<i>Постановка технічного завдання.</i>	11.03.2023	
4.	<i>Обґрунтування засобів реалізації</i>	23.03.2023	
5.	<i>Проектування.</i>	04.04.2023	
6.	<i>Анімування.</i>	26.04.2023	
7.	<i>Побудова рівнів.</i>	07.05.2023	
8.	<i>Візуалізація.</i>	18.05.2023	
9.	<i>Економічні розрахунки.</i>	01.06.2023	
10.	<i>Оформлення пояснювальної записки.</i>	25.05.2023	
11.	<i>Підготовка графічного матеріалу.</i>	04.06.2023	

Керівники роботи _____

Сергій АРТЕМЕНКО

Наталія СЛУШНА

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 _____

Вячеслав БУРЯЧЕК

АНОТАЦІЯ

Дипломна робота присвячена проектуванню та розробці 2D-гри платформеру. Метою роботи є створення захопливого та неповторного ігрового всесвіту, у який гравці зможуть зануритися захопливі пригоди.

Перша частина дипломної роботи присвячений дослідженню предметної області. Були вивчені основні поняття комп'ютерних ігор, їх види та особливості. В ході виконання проектного аналізу розглянуті аналоги ігрових продуктів різних компаній. Такий аналіз допоміг виявити їх переваги і недоліки і оцінити конкурентну перевагу гри, що розробляється, а також поглибити знання про створення оригінального концепту гри та виділити найбільш істотні переваги та недоліки цих аналогів.

Друга глава присвячена проектуванню. Рішення, прийняті на цьому етапі, стосуються структурних, поведінкових та інтерфейсних аспектів застосування. Вони дали змогу перейти до практичної реалізації ігри. Розроблено концептуальний документ гри.

У третьому розділі приведений аналіз програмних засобів, які могли б бути використані при розробці гри. З використанням ігрового редактора *Unity* проведена побудова рівнів (три ігрових рівня та вороги декількох типів, а також бос), анімація персонажів, додання музичного супроводу. Створений відеоряд на основі результатів. Створений та оброблений у відеоредакторі *MS MovieMaker* підсумковий ролик. У четвертому розділі було протестовану гру, оцінено основні моменти гри для забезпечення якісного ігрового досвіду.

Результатом усієї виконаної роботи стала 2D-гра платформер, що вигідно відрізняється від конкурентів на ринку за такими особливостями як атмосфера, гарний музичний супровід, класична механіка платформерів, варіативність у проходженні рівнів.

Ключові слова: редактор *Unity*, комп'ютерна гра, 2D платформер, анімація, відеоряд.

ABSTRACT

The thesis is devoted to the design and development of a 2D platform game.

The first section of the explanatory note to the thesis is devoted to the study of the subject area. The basic concepts of computer games, their types and features were studied. In the course of the project analysis, analogues of gaming products from different companies were considered. Such an analysis helped to identify their advantages and disadvantages and assess the competitive advantage of the game being developed, as well as deepen knowledge about the creation of the original game concept and highlight the most significant advantages and disadvantages of these analogues.

The second chapter is devoted to design. The decisions made at this stage concern the structural, behavioral and interface aspects of the application. They allowed us to move on to the practical implementation of the game.

The third chapter provides an analysis of software tools that could be used in the development of the game. Levels were built using the Unity game editor (three game levels and an enemy of several types, as well as a boss), character animation, adding musical accompaniment. Created footage based on the results. The final video created and processed in the MS MovieMaker video editor. In the fourth chapter, the game was tested, the main points of the game and their implementation were evaluated.

The result of all the work done was a 2D platformer game that compares favorably with competitors on the market in terms of such features as atmosphere, good music, classic platformer mechanics, and variability in passing levels.

Keywords: *Unity editor, computer game, 2D platformer, animation, video sequence.*

ЗМІСТ

ВСТУП	8
1 ПОСТАНОВКА ЗАВДАННЯ ДИПЛОМНОГО ПРОЕКТУВАННЯ І АНАЛІЗ ПРЕДПРОЕКТНОЇ СИТУАЦІЇ.....	11
1.1 Аналіз та загальна характеристика комп'ютерних ігор.....	11
1.2 Аналіз існуючих розробок.....	14
1.3 Постановка задачі для розробки.....	23
1.4 Технічне завдання до дипломної роботи.....	24
Висновки першого розділу.....	27
2 ПРОЕКТУВАННЯ ГРИ.....	28
2.1 Створення концепт документу.....	28
2.2 Розробка ігрових моделей персонажів та оточення.....	33
2.3 Анімація моделей.....	36
2.4 Проектування гри.....	37
2.5 Проектування алгоритмів для імітації штучного інтелекту.....	41
Висновки другого розділу.....	46
3 РОЗРОБКА ГРИ.....	47
3.1 Вибір та обґрунтування програмних засобів.....	47
3.2 База даних.....	57
3.3 Створення основної сцени.....	59
3.4 Створення головного героя.....	60
3.5 Налаштування камери.....	63
3.6 Створення перешкод, монстрів та боса.....	64
3.7 Створення бонусів та переходів між рівнями.....	70
3.8 Додавання аудіоефектів.....	72
3.9 Створення інтерфейсу користувача та системи управління грою.....	72
3.10 Опис рівнів.....	73
3.11 Тестування та відлагодження гри.....	75

КРБ.КІ.1.147-03.3.5

Змн.	Арк.	№ докум.	Підпис	Дата				
		Вячеслав БУРЯЧЕК			<i>Проектування та розробка гри- платформера "Manticora's Travel"</i>	Лім.	Арк.	Акрушів
		Сергій АРТЕМЕНКО					6	
		Олександр ЛАТЕНКО				гр. КІ-5436, ОНТУ		
		Світлана ЖУКОВЕЦЬКА						
		Сергій АРТЕМЕНКО						

Висновки третього розділу	77
4 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ПРОЄКТУ	79
4.1 Основні завдання організаційно-економічного та маркетингового обґрунтування проєкту	79
4.2 Обґрунтування ефективності проєкту	81
4.3 Оцінка науково-технічного рівня розробки	84
4.4 Розрахунок економічної ефективності проєкту	86
5 ОХОРОНА ПРАЦІ	89
5.1 Шкідливі та небезпечні фактори в роботі користувача ПК.....	89
5.2 Методи зниження впливу шкідливих та небезпечних факторів при роботі за ПК	90
5.3 Техніка безпеки при роботі за ПК (та при роботі з програмних продуктів).....	91
5.4 Вимоги безпеки після закінчення роботи з комп'ютером, принтером, ксероксом, сканером та іншою оргтехнікою.....	92
5.5 Вимоги техніки безпеки та безпеки життєдіяльності в аварійних ситуаціях при роботі з комп'ютером та іншою оргтехнікою	93
Висновки п'ятого розділу	93
ЗАГАЛЬНИЙ ВИСНОВОК.....	94
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	96
Додаток А – Презентація (копії слайдів).....	99
Додаток Б – Програмний код	106

ВСТУП

Світове суспільство постійно еволюціонує. Наші технологічні досягнення привели нас до того, що зараз ми маємо цілий світ у наших руках. Інтернет, безумовно, надав нам різноманітні можливості та доступ до великої кількості інформації. Телефони та ноутбуки допомагають залишатися на зв'язку. Телефон для кожного з нас – велика частина життя. Фотографії зберігаються в Інтернеті, контакти – на хмарному сервісі, програми в соціальних мережах завжди доступні. Наш сьогоднішній світ безперечно цифровий. Нові технології – від соціальних медіа та систем GPS до штучного інтелекту – роблять планету, на якій ми живемо, абсолютно новою і створюють новий цифровий світ.

Комп'ютер – примітивна річ, без якої, мабуть, не обійдеться жодна людина. Всього за кілька десятиліть років комп'ютерні технології розвинулися до неможливих меж. І навіть зараз їх розвиток не перестає дивувати. З кожним роком ми можемо побачити, як на полицях магазинів з'являються нові товари, які тісно зв'язані з комп'ютером, а й також являються їхніми невід'ємними частинами, без яких комп'ютер не зможе працювати. Такий швидкий розвиток зумовлений високою комп'ютеризацією. Складно уявити сучасну велику компанію або невелику фірму, які обходяться без комп'ютерів і комп'ютерних мереж. Адже момент винаходу комп'ютерної технології став переломним в житті людства, що спричинило великий вплив у розвиток промисловості, підвищило ефективність військової техніки, внесло безліч змін в роботу засобів масової інформації. Комп'ютери стрімко зайняли важливу роль у сучасних людей. З ними ми взаємодіємо майже постійно – на роботі, вдома, в машині чи літаку. Разом з появою комп'ютерів з'явилися комп'ютерні ігри, які відразу ж здобули велику популярність. Розвиток і вдосконалення комп'ютерних ігор також тісно зв'язані з розвитком комп'ютерних технологій, що дало змогу багатьом людям створювати власні ігри з характерними рисами

					КРБ.КІ.1.147-03.3.5	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

в кожній з них. Таким чином комп'ютерні ігри стали поділятися на ігрові жанри, яких на сьогоднішній день далеко за один десяток.

Ігрова індустрія багатогранна та відкрита для нових ідей, технологій та досягнень. На сьогоднішній день ігри – це не тільки спосіб цікаво провести час, це яскраві емоції, враження, незабутній досвід, це спільноти, що стрімко розвиваються, це професія і це спорт. Розробка ігор – це мистецтво.

Люди грають в ігри не стільки для самої гри, скільки для досвіду, який гра створює: захоплюючий прилив адреналіну, загадкові пригоди, психологічний виклик. Люди грають в ігри, щоб створити короточасні емоції та переживання, чи подолають вони складні ігрові виклики, шукаючи позбавлення від щоденних турбот.

Інформаційні технології, з одного боку, ускладнили життя сучасних маркетологів, ввівши нові терміни, поняття та визначення, нові методи і принципи роботи, а з іншого – надали нові можливості, розширивши межі діяльності компаній.

На сьогоднішній день є багато технічних можливостей для створення багатоплатформених ігор для різних ігрових девайсів, що дозволяє запускати програмний продукт у сегменті ринку, що росте і розвивається найшвидше у світі. Зважаючи на популярність комп'ютерних, і ігор зокрема жанру Платформер, взагалі тема кваліфікаційної роботи є актуальною.

Актуальність теми дипломної роботи полягає в тому, що з розвитком інформаційних технологій необхідно створювати нові програмні продукти, що будуть розвивати ігровий ринок та індустрію розваг в цілому.

Об'єктом дослідження даної дипломної роботи є інформаційні процеси, пов'язані зі створенням власної комп'ютерної гри.

Темою даної дипломної роботи є «Проектування та розробка гри-платформера "Manticora's Travel"».

Предметом дослідження є методи проектування і розробки комп'ютерної гри жанру Платформер.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Комп'ютерна гра – це тип відеоігор, в які грають на персональному комп'ютері (ПК), а не на ігровій консолі чи аркадному автоматі. Це обумовлено рядом причин, досліджених у дипломній роботі. Гра, як і будь-який програмний продукт проходить певні визначені етапи від задумки до готового продукту та має свої особливості щодо створення та реалізації.

Мета дипломної роботи: проектування ігрового процесу і здійснення етапів розробки комп'ютерної гри.

Для досягнення мети дипломної роботи поставлено такі завдання:

- дослідити ринок ігрової індустрії;
- дати характеристику основним програмним засобам для створення ігор;
- проаналізувати основні етапи створення комп'ютерної гри;
- на основі даних аналізу, розробити комп'ютерну гру.

Наукова новизна кваліфікаційної роботи полягає в пошуку удосконалення існуючих методів розробки в іграх жанру Платформер.

Результатом роботи буде служити гра, що повною мірою вирішить поставлені завдання.

					<i>КРБ.КІ. 1.147-03.3.5</i>	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

ПОСТАНОВКА ЗАВДАННЯ ДИПЛОМНОГО ПРОЕКТУВАННЯ І АНАЛІЗ ПРЕДПРОЕКТНОЇ СИТУАЦІЇ

1.1 Аналіз та загальна характеристика комп'ютерних ігор

Комп'ютерна гра – комп'ютерна програма, що служить для організації ігрового процесу зв'язку з партнерами по грі, або сама виступає в якості партнера. Під час комп'ютерної гри за допомогою спеціальних програм створюється імітація прямої взаємодії у віртуальному просторі між ігровим персонажем та користувачем (або групою користувачів) за певним алгоритмом. Спостерігаючи ігрову ситуацію на екрані монітора, гравець впливає на неї за допомогою клавіатури, «миші», джойстика, тощо.

Комп'ютерні ігри часто створюються на основі сторонніх джерел, таких як книги та фільми, але останнім часом існують приклади і зворотного напрямку, коли на основі відомої гри чи ігрової серії починають з'являтися додаткові матеріали, що розширюють всесвіт гри.

Більш того, спеціально розроблені ігри можуть виступати в якості навчального матеріалу або дозволяють використовувати гравців в науково-дослідних цілях. За деякими популярними іграми проводяться змагання різних видів масштабності, від регіональних до світових, які мають окрему назву «кіберспорт».

Комп'ютерні ігри мають настільки істотний вплив на сучасне суспільство, що останнім часом з'являється стійка тенденція до гейміфікації для неігрового прикладного програмного забезпечення.

Таким чином, в деяких європейських навчальних закладах почали використовувати відомі ігри для навчання, а для потреб армій створюються симулятори для тренування солдатів. Деякі країни зараховують кіберспорт до

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

офіційного виду спорту, та, наприклад, уряд США у 2011 році визнав комп'ютерні ігри окремим видом мистецтва, поряд із театром та кіно.

З усього цього можна зробити висновок, що комп'ютерні ігри щільно влились до нашого нинішнього життя. Сфера їх використання за останні роки продовжує зростати, ігри використовують не тільки для розваг, але й для проведення наукових досліджень та навчання.

Внаслідок того, що критерії приналежності гри до того чи іншого жанру не визначені однозначно, класифікація комп'ютерних ігор недостатньо систематизована, і в різних джерелах дані про жанр конкретного проекту можуть розрізнятися. Проте, існує консенсус, до якого прийшли розробники ігор, і приналежність гри до одного з основних жанрів майже завжди можна визначити однозначно.

В основі сучасних розподілів відеоігор на жанри лежить вид активності, який найчастіше здійснює гравець в іграх даного жанру. Так відеоігри в загальному можуть поділятися на ігри руху, планування і сюжету або спілкування, дії та контролю. В багатьох класифікаціях визначення жанру відбувається за кількома осями. Наприклад, за двома осями сюжет – свобода дії, або трьома абстракція – симуляція – свобода. Проте найчастіше використовуваною класифікацією, хоч і не прийнятою усіма, жанри з якої зустрічаються в більшості існуючих, є наведена нижче, яка виключає осі або багаторівневі поділи:

Екшн – в іграх такого жанру необхідно використовувати рефлексі та швидкість реакції для подолання ігрових обставин. Це один із базових жанрів і водночас найпоширеніший. Як правило екшн-ігри пов'язані із агресивними діями щодо противників і/або оточення. Персонаж гравця повинен битися, стріляти, переслідувати ціль чи самому уникати переслідування. Стосовно екшн-ігор, де наявні значні елементи пригодницьких ігор, застосовується термін *Action-adventure*. З-поміж них часто виділяється напрям аркадних ігор,

					КРБ.КІ.1.147-03.3.5	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

ігровий процес яких вирізняється простотою та легкістю освоєння.

Цей жанр поділяється на велику кількість піджанрів, серед яких основними є шутери, платформери, файтинги та лабіринти.

Стратегія – сенс стратегічних ігор полягає в плануванні дій та виробленні певної стратегії для досягнення якоїсь конкретної мети, наприклад, перемоги у військовій операції. Гравець керує не одним персонажем, а цілим підрозділом, підприємством чи навіть всесвітом. Відповідно до реалізації ігрового часу, стратегічні відеоігри поділяються на два основних різновиди: покрокові стратегічні та стратегічні ігри у реальному часі.

Симулятор – в широкому розумінні всі ігри є симуляторами. У вужчому значенні це відеоігри, призначені для складання уявлення про дійсність за допомогою відображення певних реальних явищ та властивостей у віртуальному середовищі. Існує чимало піджанрів, як технічні (управління складними технічними пристроями, авіаційною технікою та інші), аркадні (відрізняються від аркад наявністю спрощеної фізичної моделі), спортивні, економічні, побачень та інші.

Пригоди – в пригодницьких відеоіграх гравець керує ігровим персонажем, який рухається по сюжету та виконує зумовлені сценарієм завдання, покладаючись на свою уважність та логіку, здійснює пошуки підказок і вирішує загадки.

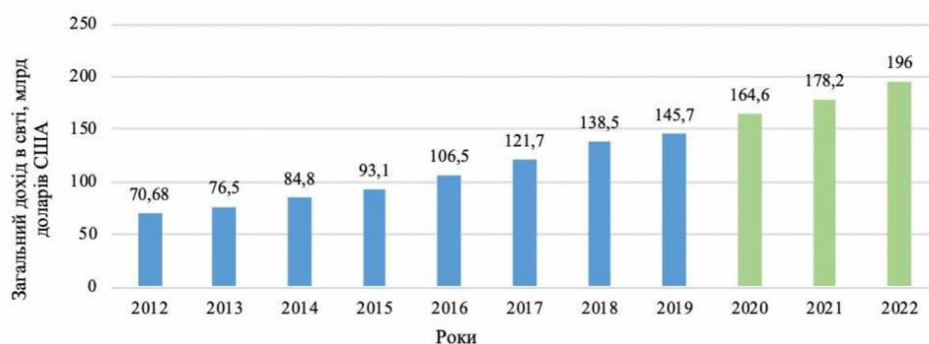


Рис.1.1 – Динаміка розвитку ігрової індустрії

					КРБ.КІ.1.147-03.3.5	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз існуючих розробок

Платформер — жанр відеоігор, ігровий процес в якому складається зі стрибків персонажа по різноманітних платформах (звідси і назва) та через перешкоди, збирання предметів, звичайно необхідних для завершення рівня.

В платформерах гравець керує персонажем, який рухається рухомими чи нерухомими платформами різної висоти. У традиційних двовимірних платформерах персонаж рухається зліва направо, деякі ігри дозволяють рухатися в будь-якому напрямку (зазвичай це загадки). Інколи рух можливий тільки вперед і не залежить від бажання гравця (автоскролінг). Цей елемент рідко зустрічається у всій грі, найчастіше в окремих епізодах, де персонаж, наприклад, повинен тікати від небезпеки. Деякі предмети, звані пауер-апами, наділяють керованого гравцем персонажа додатковими корисними характеристиками, які як правило вичерпується з часом (наприклад: силове поле, прискорення, збільшення висоти стрибків). Колекційні предмети, зброя і пауер-апи збираються зазвичай простим дотиком персонажа і для застосування не вимагають виконання спеціальних дій з боку гравця. Рідше предмети збираються в «інвентар» героя і застосовуються спеціальною командою (така поведінка більш характерна для аркадних головоломок).

Противники (звані «монстрами» незалежно від зовнішнього вигляду), як правило численні і різноманітні, володіють примітивним штучним інтелектом, керуючись яким намагаються максимально наблизитися до гравця, або не володіють ним зовсім, переміщаючись по круговій тректорії або здійснюючи повторювані дії. Зіткнення з противником забирає очки здоров'я у героя або зовсім вбиває його. Іноді противники здані стріляти або змінюватися при наближенні гравця, наприклад, раптово випускати шипи. Вороги нейтралізуються або стрибком на них, або зі зброї, якщо герой володіє нею. Смерть живих істот звичайно зображується спрощено або символічно (істота зникає або провалюється в низ екрану).

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Рівні часто мають секрети (приховані проходи в стінах, заховані у високих або важкодоступних місцях пауер-апи), які істотно полегшують проходження гри і підігривають інтерес гравця.

Ігри цього жанру, за деякими винятками, характеризуються нереалістичністю, мальованою мультиплікаційною графікою. Героями таких ігор зазвичай бувають міфічні істоти (наприклад дракони, гобліни) або антропоморфні тварини.

Щоб почати роботу над грою, необхідно досконало дізнатися про жанр та існуючі аналоги в цьому жанрі.

1.2.1 Super Mario Bros.

Це платформна відеогра, розроблена та видана компанією Nintendo (відділ *Nintendo EAD*) 1985 року.

Головною дійовою особою гри є сантехнік Маріо. У грі присутній також інший головний персонаж, молодший брат Маріо — Луїджі. Однак він з'являється лише тоді, коли в ігровому процесі одночасно задіяно двоє людей. Обидва персонажі мають аналогічний набір функцій.

Мета гри — пройти через Грибне королівство, вціліти під час зустрічі з ворогами — слугами Боузера — та визволити принцесу. Герой рухається зліва направо, де наприкінці кожного підрівня досягає флагштоку з прапором, який означає перехід на новий підрівень.

У більшій частині всіх рівнів (світів) розкидані монети, які Маріо може збирати, та спеціальні скриньки зі знаком питання на них. Підбиваючи їх кулаком з нижніх позицій, Маріо отримує або монети, або певну корисну річ. Є також секретні, здебільшого невидимі скриньки, в яких знаходяться або велика кількість монет, або корисна річ, яка зустрічається рідко.

Коли Маріо дістає червоний або жовтий гриб, він удвічі збільшується в розмірах, до того ж у нього з'являється здатність розбивати цегляні

					<i>КРБ.КІ. 1.147-03.3.5</i>	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

платформи, що дає йому можливість усувати більше ворогів та перешкод. Гравцю дається певна кількість життів (зазвичай це три), до того ж він має можливість отримати ще одне, якщо отримає зелений або помаранчевий гриб, назбирає сто монет, знищить декількох ворогів із захисним шоломом підряд або знищить певну кількість ворогів шляхом стрибання на них зверху, при цьому не торкаючись земної поверхні. Коли Маріо торкається ворогів, не будучи при цьому збільшеним у розмірах, падає в яму або коли час гри спливає, гравець втрачає одне життя. Коли Маріо втрачає останнє життя, гра закінчується.



Рис.1.2 – Знімок екрана з гри *Super Mario Bros.*

1.2.2 Sonic the Hedgehog

Відеогра у жанрі платформера, розроблена *Sonic Team*, одним з підрозділів корпорації *Sega*. Гра була випущена 23 червня 1991 року в Північній Америці і Європі для консолі *Sega Mega Drive/Genesis*, і пізніше увійшла в комплект ігор *Sega 6-Pak*. Японська версія з'явилася 26 липня.

					КРБ.КІ.1.147-03.3.5	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

На час свого виходу, *Sonic the Hedgehog* була однією з найшвидших ігор-платформерів. Ігровий процес концентрувався на елементах, які найповніше використовували можливості 16-бітної консолі. У *Sonic the Hedgehog* були присутні численні нові для ігор на *Sega* елементи, що тільки сприяли її популярності під час 16-біт ери. Головний персонаж, Їжак Сонік, міг бігати, стрибати та котитися найшвидше з-поміж персонажів інших платформерів того часу. Крім того, рівні гри були зроблені так, щоб гравці прагнули проходити їх якомога швидше.

У цій грі вперше з'явився персонаж Їжак Сонік, а сама гра дозволила консолі *Sega Genesis* отримати велику популярність у Північній Америці. Після свого виходу вона стала продаватися разом з консоллю, замінивши попередньо встановлювану гру *Altered Beast*.

Гравець керує людиноподібним їжаком Соніком, що може бігати, стрибати й згортатися в клубок. Йому належить пройти шість ігрових зон, поділених кожна на три акти, вцілівши та вклавшись у ліміт часу. На проходження кожного акту дається 10 хвилин, на заваді чому стають перешкоди, пастки і вороги. На старті Соніку дається 4 життя. На актах він збирає кільця, та маючи хоча б одне, може вберегтися від потрапляння в пастку чи від атаки ворога. В цьому разі Сонік просто розгубить кільця, але сам залишиться живий. Кільця, що розсипалися при атаці, можна зібрати, але в кількості не більше 20-ти та за обмежений час, інакше вони незабаром зникнуть. В разі смерті Соніка гравець починає акт заново або ж зі спеціальних контрольних точок, якщо гравцеві сталося відзначитися на них. Залежно від витраченого на проходження часу та донесених до кінця акту кілець, в кінці акту гравцеві присуджуються очки. Також очки нараховуються за кожного знищеного ворога.

Почавши біг, Сонік швидко прискорюється, якщо на шляху не зустрічаються перешкоди. Їжак здатний скочуватися та бігати по схилах (при

					КРБ.КІ.1.147-03.3.5	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

цьому вони мають плавну геометрію, на відміну від інших ігор того часу); спускаючись зі схилу Сонік додатково прискорюється. Біжучи досить швидко, він може долати 360-градусні мертві петлі (які згодом стали незмінним атрибутом в іграх серії).

На бігу їжачок може згорнутися в колючий клубок – в такому стані він швидше скочується зі схилів, але йому важко котитися вгору. При достатній швидкості він може пробивати стіни в деяких місцях аби скоротити шлях або відкрити сховки з корисними речами. Стрибаючи, Сонік також згортається в кулю, і в цьому стані він може атакувати ворогів – роботів – та знищувати їх простим дотиком. Сонік здатний рухатися під водою, але в ній має обмежений запас повітря – якщо протягом 18 секунд не вистрибувати на поверхню чи не торкатися бульбашок з повітрям, Сонік потоне.

Дійшовши до кінця перших двох актів кожної зони, Сонік повинен торкнутися таблички з зображенням головного лиходія гри – Егмана, відзначивши їх завершення; в кінці кожного третього акту гравця чекає бій з босом – самим Егманом, що керує різноманітними бойовими машинами.

Окрім кілець, у грі зустрічаються бонуси, що зберігаються в спеціальних моніторах. Щоб отримати їх, Соніку потрібно просто розбити монітор. У *Sonic the Hedgehog* монітори містять такі види бонусів:

- Додаткові 10 кілець;
- Щит, який огортає Соніка у вигляді сфери і дає додатковий захист (зникає при першому контакті з ворогом);
 - Тимчасове прискорення: Сонік може досягати ще більшої швидкості і здійснювати довгі стрибки;
 - Тимчасова невразливість: Соніка обволікає розсип зірочок і він стає невразливим для ворогів, і до того ж вбиває їх простим дотиком;
 - Одне додаткове життя.

Коли Сонік закінчує перший або другий акт зони мінімум з 50 кільцями,

					КРБ.КІ.1.147-03.3.5	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

з'являється можливість перейти в один з шести бонусних рівнів – *Special Stages*

– шляхом стрибка у велике кільце в кінці акту. Ці сюрреалістичні простори являють собою лабіринт із блоків, який обертається навколо своєї осі. У ньому є Смарагд Хаосу, а завданням Соніка є дістатися до нього, не торкнувшись блоків із написом

«Goal», що переливаються синім та червоним кольорами. Якщо Сонік збере 50 кілець на бонусному рівні, йому дається додаткове продовження (можливість продовжити гру після втрати всіх життів).



Рис.1.3 – Знімок екрана з гри *Sonic the Hedgehog*

1.2.3 Temple Run

Temple Run – відеогра для мобільних пристроїв жанру нескінченного ранера, розроблена студією *Imangi Studios*. В ній гравець керує шукачем скарбів, який тікає лабіринтом від охоронців скарбу, намагаючись забігти якомога далі.

					КРБ.КІ.1.147-03.3.5	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

За фавулою, шукач скарбів викрав зі стародавнього храму ідола і повинен утекти з ним від охоронців ідола — демонічних мавп. Гравець керує цим шукачем скарбів, спрямовуючи його в безпечному напрямку та збираючи монети й бонуси. Мета гри — пробігти якомога далі та набрати якомога більше очок. Головний герой біжить уперед сам, нахил пристрою відповідає за зміщення на одну з трьох умовних ліній, на які поділений шлях. Місцями шлях звертає вбік і від гравця вимагається вчасно спрямувати героя туди. Перешкоди здебільшого блокують шлях таким чином, що їх можна оминати, перестрибнувши, або просковзнувши під ними.

На шляху трапляються монети й бонуси, що полегшують втечу або дозволяють досягнути більших результатів. Спочатку монети тільки жовті (золоті), але після подолання 1000 метрів зустрічаються більш цінні червоні і сині, вартість яких відповідно вдвічі і втричі більша. Під час збору монет, поповнюється спеціальна шкала і «фреска», розташовані з лівого краю дисплею. За кожен частину фрески гравець винагороджується множителем очок. Щоразу, коли він збирає 100 монет, частина фрески заповнюється і шкала обнуляється, максимум 5 разів. Також вона обнуляється, а фреска руйнується, якщо герой спотикається, звернувши передчасно.

Після забігу виводиться статистика набраних очок, дальність забігу та кількість отриманих монет. Також виводиться ілюстрація того, в яку халепу встряв протагоніст, що супроводжується кумедним підписом. За монети перед наступним забігом купуються подовження дії бонусів, додаткові життя, персонажі. Також їх можна отримати, заплативши реальні гроші.

Зібрані монети можна витратити в магазині на покупку продовжень дії бонусів. Після покупки деяких з цих предметів з'являються на шляху втечі гравця у випадкових місцях і діють деякий час після підбору. Деякі діють постійно після покупки, а деякі активуються самим гравцем.

Випадково розкидані по рівню бонусні предмети:

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

- Мегамонета (*Mega Coin*) — одноразово додає 50 монет. Вдосконалення цього бонуса збільшують кількість монет від нього;
- Монетний магніт (*Coin Magnet*) — найближчі монети автоматично притягуються до героя. Із вдосконаленнями час дії бонусу збільшується і кількість притягнутих монет множитьься на 2 або 3;
- Непомітність (*Invisibility*) — після підбору цього бонусу персонаж не може провалитися в прірву або потрапити у пастку, однак гравцеві все одно потрібно управляти персонажем на поворотах. Час дії бонусу після покупки поліпшень збільшується;
- Прискорення (*Boost*) — персонаж швидко долає 250 метрів, автоматично оминаючи всі перешкоди і звертаючи в безпечному напрямку. Час дії бонусу і дальність після покупки поліпшень збільшується.
- Вартість монет (*Coin Values*) — після покупки збільшує вартість монет, що збираються персонажем, якщо той подолає певну дистанцію від старту. Після покупки поліпшень скорочується дистанція, необхідна для початку дії цього бонусу і додаються нові дистанції, за якими вартість зростає.



Рис.1.4 – Знімок екрана з гри *Temple Run*

					КРБ.КІ.1.147-03.3.5	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.4 Donkey Kong

Відеогра у жанрі платформера, розроблена компанією *Nintendo Research & Development 1* ексклюзивно для *NES* 1983 року. Це перша гра серії *Donkey Kong* та перша, де з'являється прототип персонажа Маріо.

Гравець керує персонажем Стрибуном (в пізніх версіях Маріо), якому належить врятувати свою подругу (в оригіналі безіменна, в пізніх версіях Пауліна), викрадену величезною мавпою Данкі Конгом. Мавпа знаходиться на вершині споруди з кількома поверхами, поєднаними драбинами. Завданням гравця є провести Стрибуна на вершину споруди та підійти до Пауліни. Стрибун може бігти по поверхах, лізти по драбинах і підстрибувати.

Йому перешкоджає Данкі Конг, скидаючи бочки, що котяться поверхами, балки, що падають прямо, або інші предмети, такі як вогняні кулі чи пружини. Стрибун втрачає одне життя, коли торкається бочки, іншого предмета, Данкі Конга, або падає з висоти.

Час від часу на рівнях виникає бонус у вигляді молотка. Поки він діє, Стрибун автоматично знищує бочки перед і над собою, але не може стрибати. Гравцеві нараховуються очки, якщо персонажу вдається перестрибнути бочку чи інший предмет (100 очок), знищити їх (300 або 500), або підібрати бонусні предмети — парасольки та сумочки Пауліни (300). Також бонусні очки присуджуються за швидкість проходження рівня. В запасі є 3 життя, додаткові даються за кожні 20000 очок.

У грі 4 рівні, в кожному з яких Данкі Конг вилізає на вищу споруду: 25, 50, 75 і 100 метрів заввишки. На першому рівні є похилі поверхи, поєднані драбинами, по яких котяться бочки. Коли Стрибун досягає Пауліни, Данкі Конг хапає її та несе далі. На другому рівні знаходяться рівні поверхи з драбинами, але літає вогняна куля і їздять вантажі цементу, а верхні драбини періодично складаються і розкладаються. Сам Данкі Конг переміщується в боки. Третій рівень має також підіймачі, на ньому немає бочок, але літають

					КРБ.КІ.1.147-03.3.5	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

вогняні кулі та стрибають пружини. На четвертому багато вогняних куль і прямого шляху до Пауліни немає. Щоб дістатись до неї, Стрибун повинен знищити всі заклепки (перестрибнути через них, або розбити), після чого споруда розвалюється і Данкі Конг падає, а шлях до Пауліни відкривається. Коли гру пройдено, можна продовжити грати з початку, але швидкість руху перешкод зростає.

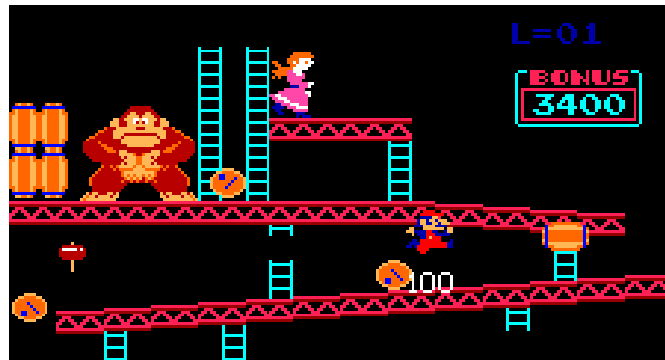


Рис.1.5 – Знімок екрана з гри *Donkey Kong*

1.3 Постановка задачі для розробки

Для досягнення поставленої мети необхідно виконати наступні дії:

- Дослідити предметну область.
- Знайти аналоги програмного продукту.
- Вибрати наступні компоненти:
 - середу розробки гри;
 - середу програмування.
- Розробити програмну оболонку, що міститиме:
 - можливість зберігання всієї інформації про об'єкти в файлах;
 - ігровий інтерфейс;
 - можливість зручного управління та гри (можливість розпочати нову гру, можливість закінчити гру, можливість комфортної гри)
- Перевірити працездатність системи і при необхідності зробити налагодження.
- Розробити інструкцію для користувача.

					КРБ.КІ.1.147-03.3.5	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

1.4 Технічне завдання до дипломної роботи

Повне найменування відеогри і її умовне позначення – «2D гра-платформер *«Manticora's travel»*».

Підстави для розробки:

1. Дані наказу (договору) – № 147-03 від 10.04.2023 р.
2. Розробник відеогри – Вячеслав Бурячек , замовник відеогри – ОНТУ
3. Перелік документів, на підставі яких створюється відеогра – повне завдання дипломного керівника.
4. Планові терміни початку та закінчення роботи: 10.02.2023–03.06.2023

Порядок оформлення і пред'явлення замовникові результатів робіт із створення відеогри, її частин і окремих етапів – виконання завдань та представлення результатів керівнику у встановлений термін часу для перевірки та внесення змін при необхідності.

Призначення розробки – отримати досвід від процесу розробки комп'ютерної гри.

Вимоги до програми або програмного виробу:

- Вимоги до функціональних характеристик:

1. перелік завдань, що підлягають автоматизації – відеогра має бути легкою та інтуїтивно зрозумілою для користувача, відповідати вимогам до гри-платформеру;
2. перелік і критерії відмов – при роботі з програмою іноді можуть виникати помилки, це може бути пов'язано з порушенням посилань до об'єктів.
3. Вимоги до надійності, безпеки, ергономіки, транспортабельності, експлуатації, технічного обслуговування, ремонту, захисту інформації від несанкціонованого доступу, захисту від впливу зовнішніх дій, до патентної чистоти, по стандартизації і уніфікації. Дана гра має дуже зручну транспортабельність.

					КРБ.КІ.1.147-03.3.5	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

4. Умови експлуатації.

Характеристики навколишнього середовища – навколишнє середовище не повинно бути вологим і температура не повинна знижуватися нижче 10 °С.

- Вимоги до персоналу (чисельність користувачів, кваліфікація, режим роботи, порядок підготовки) – працювати з грою може один гравець. Для роботи з програмою достатньо початкового рівня володіння ПК.

– Вимоги до складу й параметрів технічних засобів.

Відомості про умови експлуатації – гра працює у середовищі *Windows7/8/10*, з процесором *Ryzen 2200g*, 8 Гб ОЗУ, та відеокарті *Nvidia GeForce GTX1060*.

– Вимоги до інформаційної й програмної сумісності.

Інформаційне забезпечення (склад, структура і організація даних, обмін даними між компонентами системи, класифікатори, списки) – інформація вся зберігається в базі даних (містять записи про тип зброї та інше).

Лінгвістичному (мови програмування, мови взаємодії користувачів з грою, системи кодування, мови введення-виводу) – мова програмування, що використовувалася – *C#*, мова взаємодії користувача з системою – українська та англійська.

Програмне забезпечення (незалежність програмних засобів від платформи, якість програмних засобів і способи його контролю, використання фондів алгоритмів і програм) – рекомендується використовувати операційну систему *Windows*.

Організаційне забезпечення (структура і функції експлуатації гри, захист від помилкових дій користувача) – гра розрахована на експлуатацію багатьох користувачів.

Методичне забезпечення (склад нормативно-технічної документації) – є повне керівництво по роботі з програмою.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

- Вимоги до маркування й упакуванню – до даної прикладної програми вимоги не встановлені.
- Вимоги до транспортування й зберіганню – вимоги до транспортування та зберігання не вказані.

Вимоги до програмної документації:

- Перелік належних розробці документів – технічне завдання, пояснювальна записка, презентація.
- Перелік документів на машинних носіях – код програми, пояснювальна записка, презентація.
- Вигляд і порядок експертизи технічної документації – документація оформлена відповідно до нормоконтролю і Держстандарту.

Техніко-економічні показники:

Найменування і необхідні значення технічних, технологічних, виробничо-економічних та інших показників об'єкту, які повинні бути досягнуті при впровадженні гри.

Стадії й етапи розробки:

- Перелік стадій і етапів робіт – аналіз, проектування, реалізація, впровадження, супровід.
- Склад організацій – виконавців робіт – на стадії аналізу, а також на всіх початкових стадіях створення системи брали участь розробник і дипломний керівник проекту, далі до розробки приєдналися консультанти по охороні праці і по економічній частині.

Порядок контролю й приймання:

- Загальні вимоги до приймання робіт по стадіях – після закінчення кожної стадії розробки гри дипломному керівнику пред'являється відповідна частина коду прикладної програми і розділ дипломної записки.
- Статус приймальної комісії – дипломний керівник, члени та голова приймальної комісії.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки першого розділу

Після аналізу існуючих розробок, можливо зробити висновок, що ігор, які включають у себе лише один жанр Платформера дуже невелика кількість, через те що інші ігрові жанри також розвивались і платформери перестали відповідати вимогам гравців. Однак не можна говорити, що ігри цього жанру втратили свою популярність. На сьогоднішній день виходить маса різних платформерів, які можна зарахувати і до інших жанрів. Це може говорити про те, що «чисті платформери» стали продуктом направленим на вузький круг споживачів, тому для отримання великих прибутків розробники почали додавати до цього жанру елементи сторонніх ігрових жанрів. Тому можливо зробити заключний висновок, що для успіху гри її потрібно урізноманітнити елементами різних жанрових категорій.

У результаті аналізу предметної області було виявлено основні характеристики ігор жанру Платформер, показано історію його становлення та розвитку. Проаналізовано сучасні аналоги ігор жанру Платформер. Виявлено їх переваги та недоліки. Здійснено постановку задачі з зазначенням бажаного результату.

					КРБ.КІ.1.147-03.3.5	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

ПРОЕКТУВАННЯ ГРИ

2.1 Створення концепт документу

Концепт документу - повний опис майбутньої відеоігри. Від свого попередника, концепт-документа, відрізняється максимально можливою детальністю і опрацюванням кожного аспекту, що торкається - це і механіки, і рівні, і сюжет, і інтерфейс, і взагалі все, що має бути в грі. У певному сенсі дизайн-документ можна назвати докладним планом, в якому зазначено, що треба робити, але не зазначено ким і як (для цього є інші документи).

Концепт документ складають для спрощення роботи та підвищення її ефективності. Більшість відеоігор включають таку кількість компонентів і взаємозв'язків, яка не влізе в жодну голову. Адже є ще колеги, вони потребують актуальної інформації і якщо не зможуть знайти її на папері, то почнуть постійно ставити запитання. При вербальному обміні інформацією спочатку щось забудеться, потім згадається те, чого не було, потім переплутається все, що вже є – і так знову і знову. Результати зазвичай не радісні:

- багаторазові переробки одних і тих самих елементів;
- дублювання роботи;
- хаос у взаємодії із колегами;
- поступове різнопланове зміна загальної структури і, як наслідок, порушення цілісності готової відеоігри.

За допомогою концепт документу всього перерахованого вище легко уникнути або, принаймні, знизити до терпимого мінімуму. Відповідальним за цю роботу призначається ігровий дизайнер (він же геймдизайнер, іноді – гейм-дизайнер (від англійського game designer) – загалом назва професії поширена,

					КРБ.КІ.1.147-03.3.5	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

але не усталена в написанні). Він необов'язково буде один - все залежить від складності гри, що розробляється.

Так, наприклад, над документацією рольової гри кількість трудящих ігрових дизайнерів може рахуватися не в «людях», а у відділах. Невеликі ж аркади, головоломки, платформери, «сайд-скролери» та інші подібні відеоігри менш вимогливі, і для їх розробки часом достатньо не тільки одного ігрового дизайнера, а й взагалі однієї людини.

В концепт документ дуже бажано додати таку інформацію:

- Платформи: ПК, консолі та мобільні пристрої. • Технології: чи використовуватимете ви готовий двигун або напишете свій.
- Мова: українська, англійська.
- Аудиторія: короткий опис людей, для яких гра робиться. Повний опис можна винести у спеціальний розділ.
- Жанр: наприклад, 2D гра жанру Платформер.
- Настрій: похмурий, світлий.
- Емоції: які почуття має викликати гра. При цьому можна спробувати надихнути гравців похмурою грою або змусити відчувати приреченість у світлій.
- Вікове обмеження
- Кількість користувачів, режим: чи буде гравець один чи в команді з іншими гравцями. Чи потрібне підключення до інтернету.
- Час, місце та тривалість гри: де і як довго людина повинна грати у вашу гру, щоб отримати ті емоції, які ви хочете передати. Наприклад, можна грати три хвилини в метро чи черзі чи засісти на кілька годин увечері у себе вдома.
- Головна ігрова механіка: короткий, але ємний опис того, чим займатиметься гравець.

Розробимо концепт документ для дипломного проекту:

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

- Назва: Manticora’s Travel;
- Жанр: Платформер;
- Мова: українська;
- Аудиторія: діти та дорослі;

Гра орієнтована на молодшу та підліткову аудиторію, тому не містить обмежуючого контенту, мінімальний вік гравця – 3 роки. Для молодшої частини цієї категорії характерна потреба у простих іграх. Гра є простою та динамічною, має просте керування та зрозумілі правила гри і безперечно підходить для проведення вільного часу.

- Вид ігрового простору: двовимірна графіка (2D), вид збоку;
- Настрій: бадьорий;
- Емоції: прагнення до перемоги;
- Вікове обмеження: дітям віком від 12 років;
- Кількість користувачів, режим: гра створена для одного ігрока, не потрібне підключення до інтернету;
- Час, місце та тривалість гри: можна грати, находячись удома або пересуваючись у транспорті для скорочення часу;
- Головна ігрова механіка: головні механіки – це біг і подолання перешкод;

Ігрові об'єкти:

1. Персонаж: Мантикора;

1.1 Зовнішній вигляд: лев з крилами;

1.2 Можливості: пересування вліво та вправо, стрибок, підтягування (автоматичне, при взаємодії з кутами поверхонь). Рух із невеликим ковзанням – при зупинці персонаж продовжує рухатися, плавно сповільнюючись. Падіння нерегульоване.

Життя має кількісний вираз: базове значення: 3 одиниці. Можливо знайти декілька одиниць. Життя зменшується при взаємодії з ворогом та об'єктами. При зменшенні кількості життя

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

персонаж переноситься на початок поточного рівня в точку появи.

2. Поверхні:

2.1 Поляна: основна поверхня знаходиться у всіх рівнях;

2.2 Мотузка: героєві потрібно пересуватися по ній.

3. Перешкоди:

3.1 Рухаюча платформа: якщо персонаж потрапить під платформу, то у нього вираховується одне життя;

3.2 Шипи;

3.3 Стріла;

3.4 Леза;

3.5 Ворог та головний бос гри.

4. Предмети: життя – надає можливість відновити недостачу життів, каміння, дерева, цегла – елементи декору.

– Ігрові рівні: гра має 2 рівні та один з героєм-босом.

– Управління: пересування героя відбувається за допомогою стрілок праворуч/ліворук, стрибок – стрілка вгору, вогонь по ворогові – клік мишкою.

Ключові особливості гри:

– растрова графіка;

– атмосферний музичний супровід;

– вороги незвичайного вигляду;

– класична механіка платформерів;

– детально пророблені рівні з декількома варіантами проходження.

Сюжет гри буде розрахований на 0,5 години.

Основною задачею гравця буде знайти боса і знищити його. Гравець вирішує, якою дорогою йому йти та як поступати з ворогами — вбивати або обходити.

Перешкодами на дорозі гравця будуть:

– шипи;

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

- ворог;
- платформа;
- стріли;
- головний бос гри.

Також гравець буде знаходити на своєму шляху серця для поповнення життя персонажа.

Для головного героя будуть реалізовані наступні механіки:

- біг, помірна швидкість;
- стрибок;
- атака(дальня дія).

Головний герой буде мати наступні анімації:

- анімація бездіяльності (коли герой стоїть на місці не рухаючись);
- анімація бігу;
- анімація стрибку;
- анімація смерті.

У грі заплановано реалізувати декілька видів ворогів:

- нерухомі вороги (шипи);
- рухомі вороги з анімацією (платформа та стріли), які наносять шкоду герою, при зіткненні;
 - стріляючий ворог и анімацією (Ворог), який випускає червоний снаряд який наносить шкоду герою при зіткненні.
 - стріляючий ворог з анімацією (Бос гри), який випускає червоний снаряд який наносить шкоду герою при зіткненні.

Для ворогів створити анімацію:

- рухомий ворог — платформа має анімацію пересування по траєкторії вгору/донизу;
- стріляючий ворог — ворог та бос мають анімацію пострілу.

Бос матиме наступні механіки:

- пересування стрибками, невелика швидкість;

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

- нанесення шкоди при контакті та снарядами;
- нанесення шкоди снарядами;
- завершення рівня після знищення.

В грі заплановано три рівні, а також рівень с босом. Дизайн рівнів починається з простого і стає дедалі складнішим з проходженням на наступні. Об'єм рівнів стає більшим з проходженням на наступні.

Графіка гри в піксельному стилі, що може викликати ностальгію у людей, які часто грали в класичні ігри.

2.2 Ігрові моделі персонажів та оточення

Основною складовою частиною будь-якої комп'ютерної гри є її візуальна складова. Перед початком опису частини графічного оформлення, слід розібрати основні поняття, що використовуються у геймдизайні, такі як спрайт та тайл.

Тайл – повторюваний фрагмент невеликих розмірів, який служить для створення зображень великих розмірів, а сам цей процес має назву Тайлова графіка. Методи тайлової графіки використовуються для створення рівнів у двовимірних та тривимірних іграх.

Спрайт – у двовимірних іграх є графічним зображенням якогось об'єкту, та може містити у собі декілька зображень, з яких можна зробити анімацію до об'єкту.

Для реалізації графічної складової нашого прототипу було взято вже готовий набір двовимірних спрайтів з магазину Unity Asset Store, який містить у собі спрайти декількох жанрів та типів ігор на вибір розробника, тому було вибрано спеціальні спрайти та текстури, що підходять до нашого проекту. Для роботи зі спрайтами Unity має вбудовану функцію Sprite Editor (Рис.2.1)

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

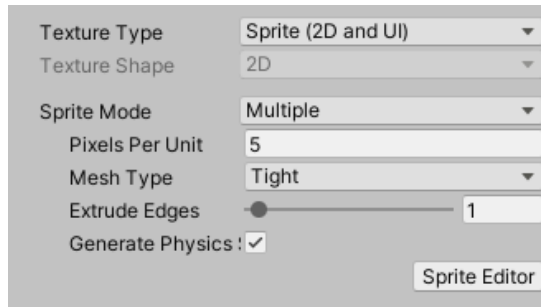


Рис.2.1 – Вікно функції Sprite Editor

Sprite Editor дозволяє відкривати великий спрайт-об'єкт, щоб розділити його на складові, для створення анімацій у наступних кроках.

2.2.1 Спрайт ігрового героя

У якості персонажа у даній грі буде використано спрайти мантикора, який має окремі спрайти для створення анімації простою, бігу, стрибків. (Рис.2.2)



Рис.2.2 – Головний герой Мантикор

2.2.2 Спрайт інтерфейсу

Після ігрового персонажа наступним кроком слід підібрати спрайти для фонового зображення, а також елементи візуальної складової гри.

У якості фонового зображення було підібрано поляну, небо та підземний тунель. (Рис.2.3-2.4)

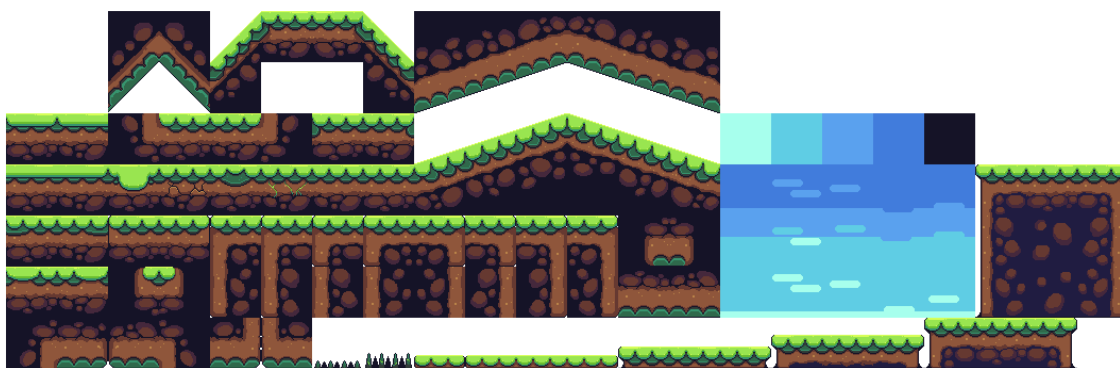


Рис.2.3 – Спрайт поляни та неба

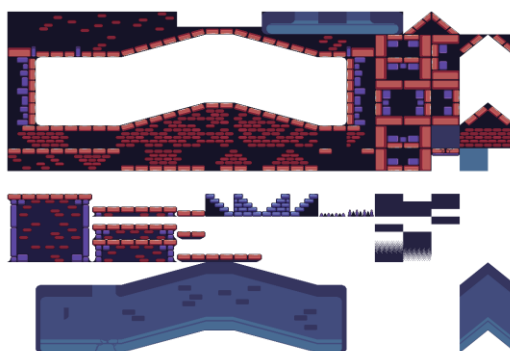


Рис.2.4 – Спрайт підземного тунеля

2.2.3 Спрайт ворогів

Останнім елементом візуальної складової гри є спрайти ворогів, які будуть нападати на ігрового персонажа або наносити йому пошкодження. Було обрано спрайт ворога, головного боса, стріл, леза, платформи та шипи. (Рис.2.5-2.7).



Рис.2.5 – Спрайт ворога

					КРБ.КІ.1.147-03.3.5	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис.2.6 – Спрайт головного боса



Рис.2.7 – Спрайти стріл, леза, платформи та шипа

2.3 Анімація моделей

Кожний спрайт має свої стани, для яких у подальшому буде написано код. Щоб героя заанімувати потрібно додати декілька його станів, наприклад, спокійний стан, біг, падіння та смерть (Рис.2.9-2.12).



Рис.2.9 – Спрайт героя у спокійному стані



Рис.2.10 – Спрайт героя при бігу



Рис.2.11 – Спрайт героя при падінні



Рис.2.12 – Спрайт героя у стані смерті

2.4 Проектування гри

2.4.1 Стартове меню гри

Першим, що побачить перед собою потенційний гравець буде ігрове меню. Ігрове меню має 3 кнопки: Старт, Керування та Вихід. Натиснувши Старт гравець переходить до гри, натиснувши Керування гравець потрапляє до екрану з поясненням керування, натиснувши Вихід закриває ігрове меню. Загальний фон та інтерфейс меню був виконаний з вогнем на фоні. (Рис.2.13)

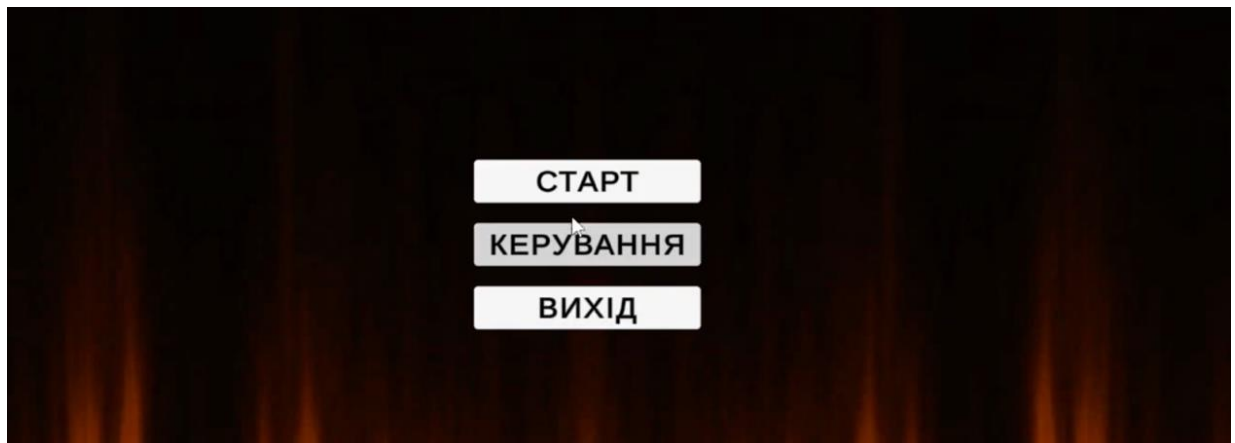


Рис.2.13 – Стартове меню гри

2.4.2 Інтерфейс та ігровий процес

Після натиснення кнопки Старт гра починається. Управління у грі відбувається за допомогою кнопок на клавіатурі стрілка праворуч/ліворуч – пересування, стрілка вгору – стрибати, клік – вогонь.

					КРБ.КІ. 1.147-03.3.5	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Гравець бачить перед собою свого ігрового героя, навколишній світ, платформи, по яким можна стрибати, кількість його життів, які вказують на кількість очок життя. (Рис. 2.14)

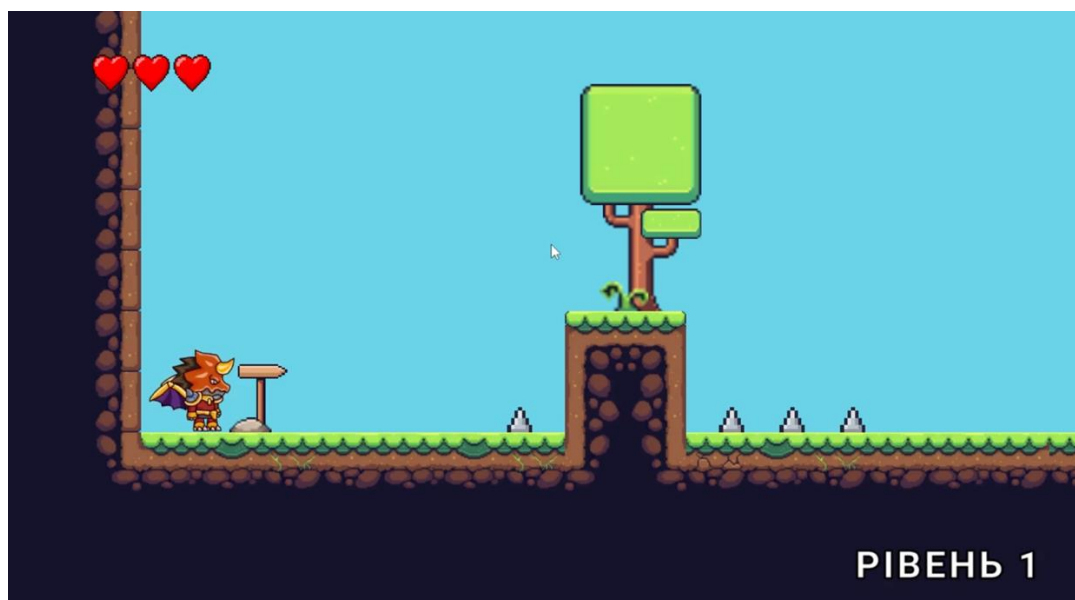


Рис.2.14 – Інтерфейс гри

Гравець починає гру, пересуваючись стрілками направо/наліво, він бачить шипи, які героєві потрібно уникнути. Треба перестрибнути їх, за допомогою стрілки вгору. Якщо герой не перестрибне шип, то він втрачає одне життя. (Рис.2.15)

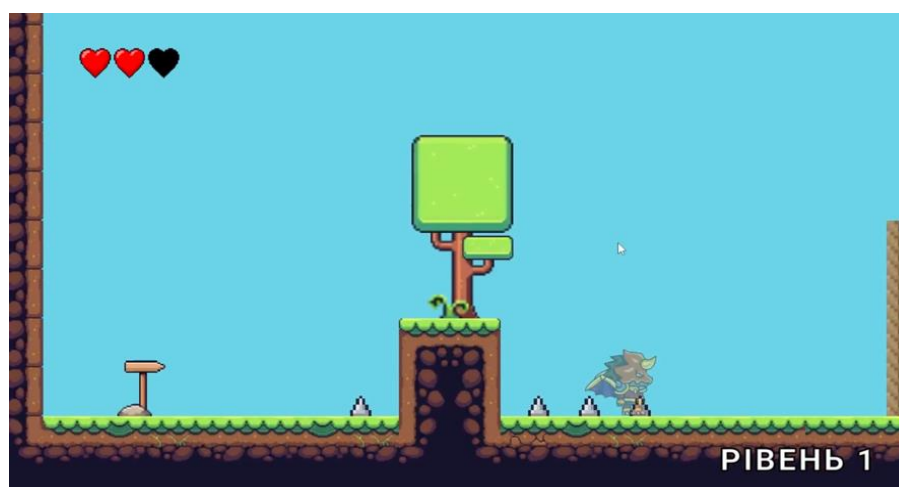


Рис.2.15 – Втрата життя

					КРБ.КІ.1.147-03.3.5	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Наступнім кроком героєві потрібно уникнути літаючої платформи, щоб здобути додаткове життя. (Рис.2.16)

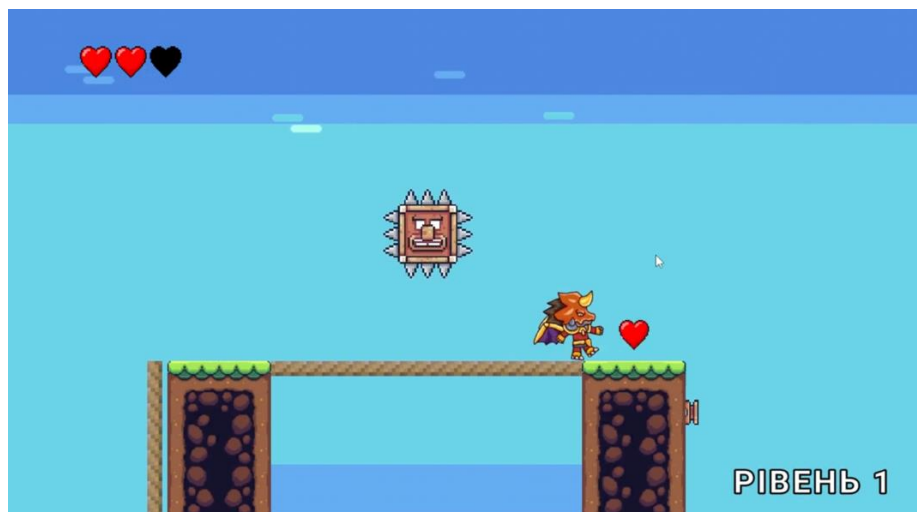


Рис.2.16 – Здобування додаткового життя

Крок за кроком герой проходить перешкоди, щоб дістатися двері, яка дає можливість перейти на наступний рівень. (Рис.2.17)

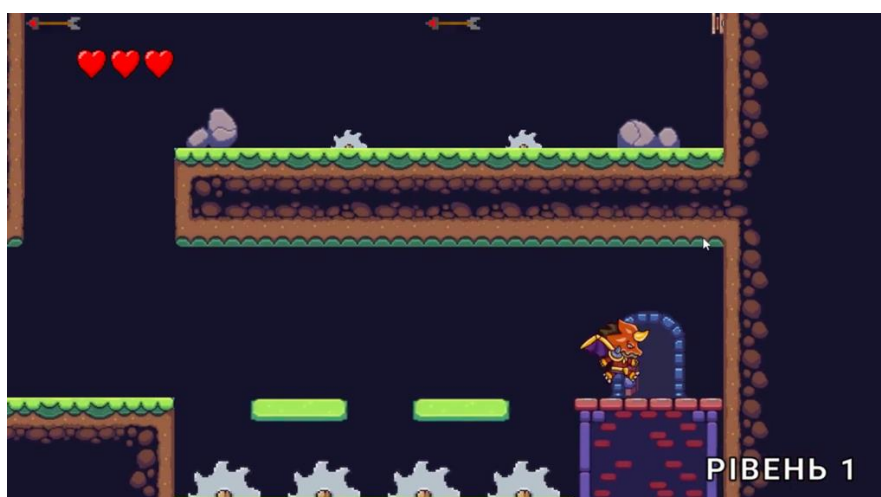


Рис.2.17 – Двері, які дають можливість перейти до наступного рівня

На наступному рівні героєві потрібно пройти усі перешкоди та здолати його ворога, аби мати змогу перейти на наступний рівень, щоб зустрітися з босом гри. Щоб здолати ворога, потрібно клікати мишкою кожного разу, аби з'явився постріл, який буде задавати урон героєві (Рис.2.18)

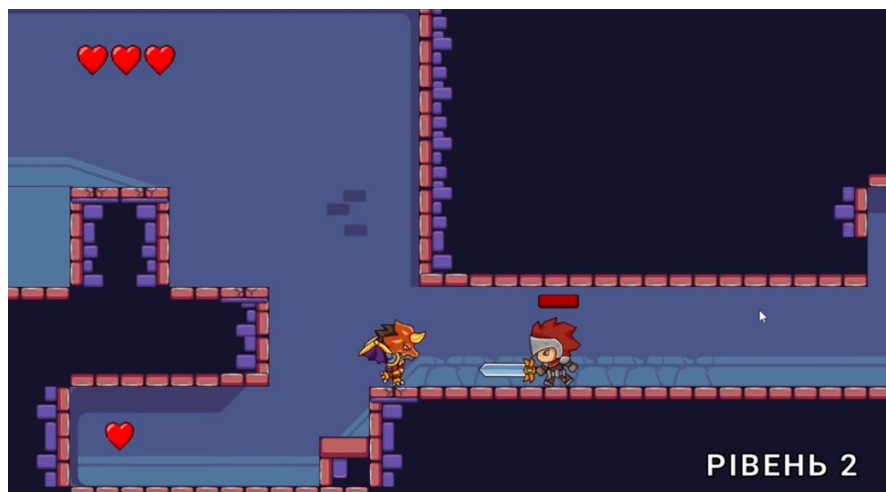


Рис.2.18 – Поява ворога

Останній рівень – рівень з босом, щоб виграти у грі треба його здолати.
(Рис.2.19)



Рис.2.19 – Поява боса гри

2.4.3 Умови та меню завершення гри

Умовою для завершення гри є знешкодження усіх хвиль ворогів, або втрата усіх життів.

Після виконання однієї із умов для перемоги або поразки, гравець бачить перед собою меню завершення гри, де йому пропонуються вибір, спробувати ще раз, або завершити та вийти із гри. (Рис. 2.20)

					КРБ.КІ.1.147-03.3.5	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис.2.20 – Меню завершення гри

2.5 Проектування алгоритмів для імітації штучного інтелекту

Ігровий штучний інтелект (ШІ) (*Game artificial intelligence*) – набір програмних методик, які використовуються в комп'ютерних іграх для створення ілюзії інтелекту в поведінці персонажів, керованих комп'ютером. Ігровий ШІ, крім методів традиційного штучного інтелекту, включає також алгоритми теорії управління, робототехніки, комп'ютерної графіки та інформатики в цілому.

Інтелектуальною поведінкою ігрового персонажа є його здатність досягати поставленої мети. Інтелектуальність поведінки залежить від запасу використовуваних основних принципів і якостей процесів міркування загального вигляду. Використовувані системою основні принципи характеризують широту охоплення ігрової області. Важливим є відображення відомих системі принципів на ступінь відомих їй подробиць. Якість міркувань залежить від доступності фактів, які мають відношення до ситуації, і принципів і повноти процедури виведення та ефективності її реалізації.

					КРБ.КІ.1.147-03.3.5	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

У традиційних дослідженнях в галузі ШІ метою є створення справжнього інтелекту, хоча і штучними засобами. В таких проектах, як *Kismet* Массачусетського технологічного інституту (МТІ) робиться спроба створити ШІ, здатний до навчання і до соціальної взаємодії, до прояву емоцій. Наразі в МТІ ведеться робота над створенням ШІ, що має рівень здібностей маленької дитини, і результати цієї роботи дуже перспективні.

З точки зору ігор справжній ШІ далеко виходить за рамки вимог розважального програмного проекту. В іграх така потужність не потрібна. Ігровий ШІ не повинен бути наділений почуттями і самосвідомістю, йому немає необхідності навчатися чогось за межами рамок ігрового процесу. Справжня мета ШІ в іграх – імітація розумної поведінки і в наданні гравцеві переконливої, правдоподібної мети та постановки відповідної задачі.

1.5.1 Базове сприйняття навколишнього середовища

Щоб штучний інтелект міг приймати осмислені рішення, йому необхідно якимось чином сприймати середовище, в якому він знаходиться. У простих системах таке сприйняття може обмежуватися простою перевіркою положення об'єкта гравця. У більш складних системах потрібно визначати основні характеристики і властивості ігрового світу, наприклад можливі маршрути для пересування, наявність природних укриттів на місцевості, області конфліктів.

При цьому організатори повинні придумувати спосіб виявлення і визначення основних властивостей ігрового світу, важливих для системи ШІ. Наприклад, укриття на місцевості можуть бути заздалегідь визначені дизайнерами рівнів або заздалегідь обчислені при завантаженні або компіляції карти рівня. Деякі елементи необхідно обчислювати на льоту, наприклад карти конфліктів і найближчі загрози.

У багатьох іграх зі стріляниною було б добре, якби герой або вороги вміли ховатися за укриттями, якщо вони є поблизу, а не просто стояти під

					<i>КРБ.КІ. 1.147-03.3.5</i>	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

вогнем противника на відкритому місці. Яким же чином вони зможуть визначати, чи є поруч якісь відповідні укриття?

Ця проблема насправді складається з двох задач: по-перше, потрібно правильно розпізнати укриття на основі геометрії навколишнього світу; по-друге, потрібно правильно розпізнати укриття на основі об'єктів навколишнього світу. Щоб визначити, чи здатне укриття захищати від атак, можна просто один раз порівняти розмір граничної межі агента з розмірами можливого укриття. Потім слід перевірити, чи зможе ваш об'єкт поміститися за цим укриттям. Для цього потрібно провести промені від відмінностей в положеннях вашого стрілка і укриття. За допомогою цього променя можна визначити, чи є вільним місце, що знаходиться за укриттям (якщо дивитися з боку стрілка), а потім позначити це місце як наступну мету переміщення агента.

Прикладом є: Коли головний герой потрапляє на другий рівень, то йому потрібно уникнути вогонь ворога, тому було створено розгалуження нижче ворога, аби герой зміг сховатися від нього. (Рис.2.21)

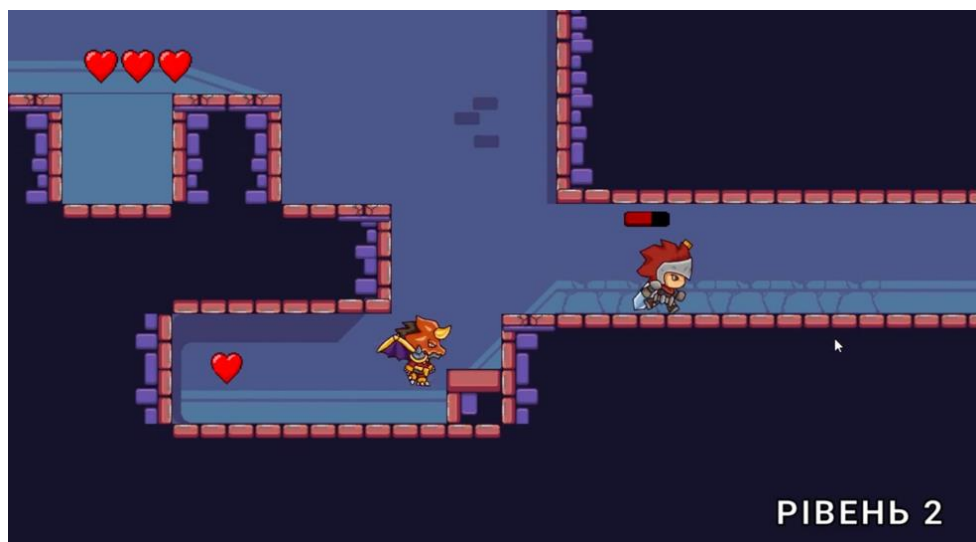


Рис.2.21 – Розгалуження

Аби для того, щоб штучний інтелект розумів, що потрібно робити, було створено алгоритм, на якому наш герой визначив, що в місці, позначеному

					КРБ.КІ.1.147-03.3.5	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

зірочкою, можна буде сховатися в безпеці (Рис.2.22):

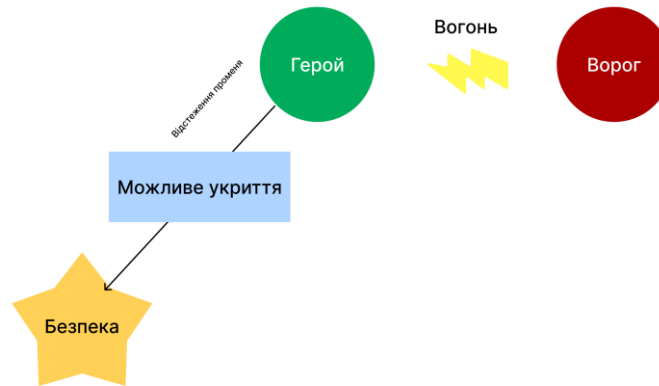


Рис.2.22 – Алгоритм безпеки

1.5.2 Навігація штучного інтелекту

Прийнявши рішення, герой повинен зрозуміти, як рухатися з точки А в точку Б. Для цього можна використовувати різні підходи, вибравши оптимальний залежно від характеру гри і від потрібного рівня продуктивності.

Алгоритм під умовною назвою «Зіткнутися та повернути», є одним з найпростіших способів формування маршруту руху об'єкта. Ось як це працює.

- Герою потрібно рухатися в напрямку мети.
- Якщо герой зіткнеться зі стіною, йому треба повернутися в напрямку, при якому він опиниться найближче до мети. Якщо жоден з доступних для вибору варіантів не має очевидних переваг, вибір робиться довільним чином.

Якщо ж повороти в усіх можливих напрямках не привели до успіху, герой зможе повернутися назад і вибрати інший маршрут руху. Таким чином, герой буде виробляти систематичний пошук шляху до мети. Ось як це працює.

- Герою потрібно рухатися в напрямку мети.
- Якщо шлях розгалужується, треба обрати один з можливих напрямків.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

- Якщо шлях приводить в глухий кут, треба повернутися до останнього розгалуження і вибрати інший напрямок.
- Якщо всі можливі шляхи пройдені безрезультатно, відмовляйтеся від подальшого пошуку.

1.5.3 Пошук шляхів

Пошук шляхів можна вважати вже давно і успішно вирішеною проблемою в розробці ігор.

Для визначення шляхів руху використовується алгоритм під назвою A^* (вимовляється а-стар). З його допомогою можна знаходити оптимальний шлях між двома будь-якими точками в графі (в даному випадку – на карті). Простий пошук в Інтернеті видає чистий алгоритм, який використовує вкрай «зрозумілі» описові терміни, такі як F , G і H .

Спочатку потрібно створити два списки: список вузлів, які ще не перевірені (*Unchecked*), і список вже перевірених вузлів (*Checked*). Кожен список включає вузол розташування, передбачувану відстань до цілі і посилання на батьківський об'єкт (вузол, який помістив даний вузол в список). Спочатку списки порожні.

Тепер додамо початкове розташування в список неперевірених вузлів, не вказуючи нічого в якості батьківського об'єкта. Потім вводимо алгоритм.

- Обираємо в списку найбільш підходящий вузол.
- Якщо цей вузол є метою, то пошук завершено.
- Якщо цей вузол не є метою, додаємо його до списку перевірених.
- Для кожного вузла, сусіднього з цим вузлом:
 1. якщо цей вузол непрохідний, ігноруємо його;
 2. якщо цей вузол вже є в будь-якому зі списків (перевірених або неперевірених), ігноруємо його;

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

3. в іншому випадку додаємо його в список неперевіраних, вказуємо поточний вузол в якості батьківського і розраховуємо довжину шляху до мети (досить просто обчислити відстань).

Коли об'єкт досягає поля мети, можна побудувати шлях, відстеживши батьківські вузли аж до вузла, у якого немає батьківського елемента (це початковий вузол). При цьому ми отримуємо оптимальний шлях, по якому може переміщатися об'єкт.

Висновки другого розділу

Було створено концептуальний документ гри, необхідний для презентації видавцям чи інвесторам.

Рішення, прийняті на етапі проектування відеогри, представлені в цьому розділі, стосуються структурних, поведінкових та інтерфейсних аспектів застосування. Вони дають змогу перейти до практичної реалізації ігри з використанням відповідних засобів в обраному середовищі розробки. Взагалі, при використанні спіральної моделі проектування, можливий перегляд цих рішень з наступним внесенням змін до реалізації.

					<i>КРБ.КІ. 1.147-03.3.5</i>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3

РОЗРОБКА ГРИ

3.1 Вибір та обґрунтування програмних засобів

Ігровий рушій – програмне забезпечення, призначене для розробки комп'ютерних ігор та інших інтерактивних програм, що обробляють графіку у режимі реального часу.

Для аналізу було взято на розгляд декілька ігрових рушіїв – *Unity 3D*, *CryEngine*, *Unreal Engine 4*.

3.1.1 Unity 3D

Unity – більше, ніж движок, це середовище розробки комп'ютерних ігор, у якій об'єднані різні програмні засоби, використовувані під час створення ПЗ – текстовий редактор, компілятор, відладчик тощо. При цьому, завдяки зручності використання, *Unity* робить створення ігор максимально простим і комфортним, а мультиплатформеність движка дозволяє розробникам охопити якнайбільшу кількість ігрових платформ та операційних систем.

Насамперед, движок *Unity3D* дає можливість розробляти ігри, не вимагаючи для цього якихось спеціальних знань. Тут використовується компонентно-орієнтований підхід, у якого розробник створює об'єкти (наприклад, головного героя) і до них додає різні компоненти (наприклад, візуальне відображення персонажа та способи управління ним). Завдяки зручному *Drag & Drop* інтерфейсу і функціональному графічному редактору движок дозволяє малювати карти і розставляти об'єкти в реальному часі і відразу ж тестувати результат.

Цей движок також відомий як середовище розробки інді та мобільних ігор. Деякі статистичні дані: станом на вересень 2019 року 52% з 1000

					КРБ.КІ.1.147-03.3.5	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

найкращих мобільних ігор були створені на базі *Unity*, а також 60% всього контенту *AR/VR*, за даними компанії.

Гравці в ігри, створені у *Unity* знаходяться в 195 країнах – це буквально кожна країна на планеті. Платформа для розробки в режимі реального часу дозволяє створювати *2D*, *3D*, *VR* і *AR*-контент і досягла 3 мільярдів пристроїв за останні 12 місяців.

Основна причина популярності *Unity*, особливо для невеликих проектів: це безкоштовно. Персональна ліцензія движка дозволяє розробникам створювати комерційні ігри безкоштовно, за умови, що вони не заробили або не отримали більше 100 000 доларів США доходу або фінансування за останні 12 місяців.

Unity дозволяє легко створювати інтерактивний *3D*-контент. Цей ігровий движок сьогодні вибирають багато великих організацій через його відмінну функціональність, високоякісний контент і можливість використання для будь-якого типу гри. Він підтримує як *2D*-, так і *3D*-контент

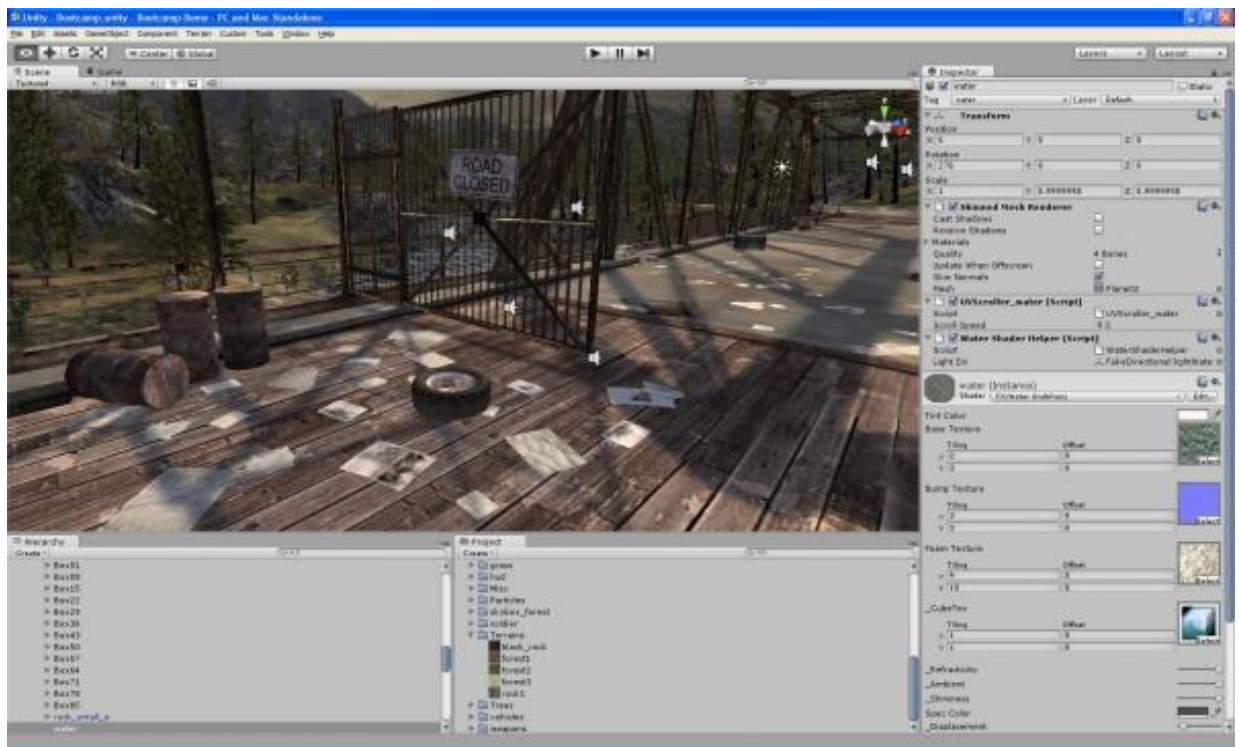


Рис.3.1 – Інтерфейс середовища розробки *Unity3D*

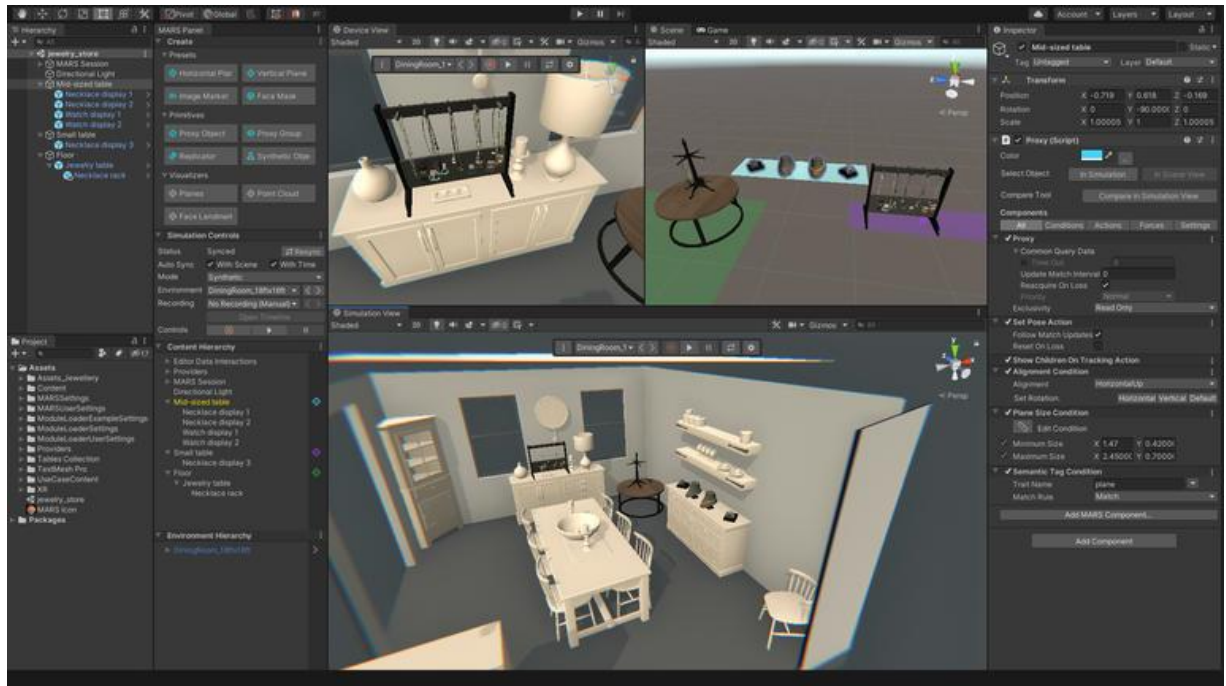


Рис.3.2 – Створення гри з використанням VR технології

Завдяки універсальному редактору *Unity* сумісний із *Windows, Mac, Linux, IOS, Android, Switch, Xbox, PS4, Tizen* та іншими платформами. Дружній інтерфейс спрощує розробку та знижує потребу у навчанні. *Unity Asset Store* зберігає велику колекцію інструментів та контенту, які створюються щодня.

Сьогодні *Unity* є одним з найпопулярніших ігрових движків у всьому світі. Як повідомляється на офіційному сайті *Unity*, понад 50% мобільних ігор розроблено на движку *Unity 3D*. Також, 34% найкращих ігор AAA, створених за допомогою *Unity*. Крім того, на движку *Unity 3D* встановлено близько 28 мільярдів ігор, що втричі більше, ніж населення світу. А це говорить про неймовірний попит серед користувачів.

Переваги *Unity*:

- *IDE*: поєднання редактора сцен (у комплексі загального редактора) із редактором ігрових об'єктів та редакторів скриптів. Додатково додаються генератори дерев та терейнів;
- покращені можливості скриптингу, а саме в *unity* доступні три мови: *JavaScript, C#* і різновид *Python's boo*;

					КРБ.КІ.1.147-03.3.5	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

- кроссплатформенність – як згадувалося вище, підтримуються Windows, MacOS, Wii, iPhone, iPod, iPad, Android, PS3 і Xbox 360, не всі з яких доступні в безкоштовній ліцензії;
 - сучасний рівень графіки, здатний конкурувати з іншими двигунами. *Unity*, безумовно, програє *unrealengine* за кількістю реалізованих можливостей. Однак *Unity* має такі можливості, як deferred освітлення, стандартний набір постпроцесингових ефектів, ssao, прискорене опрацювання лайтмапів;
 - гідним чином опрацьований фізичний двигун;
 - масштабованість та продуктивність. Більшість простих процесів двигун обробляє на чудовому рівні;
 - запуск будь-якої програми на *Unity* у веб-плагіні;
 - невисока ціна за повну ліцензійну версію для великого веб-розробника.
- Недоліки *Unity*:
- закритість коду, неможливість отримання вихідних кодів двигуна навіть за ліцензією;
 - неможливість доповнення фізики двигуна сторонніми можливостями, наприклад сторонню фізику, або *speedtree*.

3.1.2 CryEngine

Остання версія ігрового рушія розробленого та виданого компанією *Crytek*. Даний рушій має відмінну графіку. Першою грою, яка була випущена на *Cry Engine* стала популярна *Far Cry* (Рис.3.3)

					КРБ.КІ.1.147-03.3.5	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис.3.3 – Знімок екрану з гри, створеної в Far Cry

Хоча *Cry Engine 5* заявлено як кросплатформерний рушій, він підтримує лише 4 платформи – *Windows, Xbox One, PlayStation 4* та *Oculus Rift*.

Для роботи з ігровим рушієм потрібні відносно невисокі характеристики комп'ютера:

- Персональний комп'ютер з операційною системою *Windows 7/8/10 64x*;
- Процесор *Intel Dual-core* (або інший від 2GHz);
- Оперативна пам'ять 4 GB;
- Відеокарта *NVIDIA GeForce 450, AMD Radeon HD 5750* (або вище з підтримкою *DirectX 11*).

У *Cry Engine* зроблено акцент на візуальну складову проекту, тому у ньому присутній різний функціонал на інструментарій для досягнення найкращої якості зображення. Також рушій підтримує останню версію *DirectX 12*, набору *API* що відповідає за візуальну складову гри. *Physically Based Rendering* – система у *Cry Engine*, що дозволяє імітувати взаємодію світла з матеріалами та використовує фізику реального світу, що надає правдоподібності вигляду об'єктів. Також за допомогою динамічної обробки водних каустик досягається висока якість зображення води. Даний рушій має інструменти для ефективного

					КРБ.КІ.1.147-03.3.5	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

згладжування та зниження пікселізації зображення. Присутня глобальна настройка освітлення, яка дозволяє ви-користовувати більший тональний діапазон, ніж інші рушії, за допомогою якого можна поліпшити розпізнавання світлих, середніх та темних тонів без особливих зусиль.

Для роботи з матеріалами рушій має вбудований редактор *Cry Engine SandBox*, який пропонує великий інструментарій для створення рівнів та ігрових світів.

За допомогою інструменту *Designer Tool* є можливість редагувати моделі та експортувати створені матеріали у зовнішні інструменти. Також існує редактор *TrackView*, що дозволяє створювати відеозаписи та інтерактивні сцени з об'єктами.

Так само як і у *Unreal Engine 4*, у *Cry Engine* присутня система візуальних скриптів *FlowGraph*, яка дозволяє створювати та контролювати ігрову логіку без написання коду та скриптів вручну.

Рушій має велику кількість налаштувань для анімації персонажів, в тому числі параметричну кісткову анімацію. Є вбудована розширена система штучного інтелекту, що дозволяє налаштовувати реалістичну поведінку персонажів.

Cry Engine має можливість аналізувати продуктивність прямо під час гри, а вбудований інструмент *Statoscope* (Рис.3.4) дозволяє у графічному вигляді слідкувати за витратою ресурсів комп'ютера для майбутньої оптимізації.

					КРБ.КІ.1.147-03.3.5	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис.3.4 – Інструмент Statoscope

3.1.2 Unreal Engine 4

Одним з найпопулярніших і широко використовуваних ігрових двигунів є *Unreal Engine*, що належить *Epic Games*. По суті, це мультиплатформний двигун для розробки ігор, призначений для підприємств будь-якого розміру, який допомагає використовувати технології реального часу для перетворення ідей у привабливий візуальний контент.

Створений розробниками ігор для розробників ігор цей висококласний ігровий движок більше не призначений тільки для студій з мільйонним доходом. Геймдизайнери будь-якого рівня підготовки тепер мають змогу втілити свої ідеї у власну епічну гру.

Початкова версія була випущена ще в 1998 році, і через 19 років вона продовжує використовуватися для деяких з найбільших ігор. Головна перевага *Unreal Engine* полягає в тому, що він може бути досить модифікований, щоб у іграх можна було створити унікальний ігровий досвід. Однак для цього потрібні кваліфіковані розробники з великим досвідом.

Написаний мовою C++, рушій дозволяє створювати ігри для більшості

операційних систем і платформ: *Microsoft Windows, Linux, Mac OS i Mac OS X, консолей Xbox, Xbox 360, PlayStation 2, PlayStation Portable, PlayStation 3, Wii, Dreamcast i Nintendo GameCube*. У грудні 2009 Марк Рейн продемонстрував роботу рушія *Unreal Engine 3* на *iPod Touch* і *iPhone 3GS*. У березні 2010 робота рушія була продемонстрована на комунікаторі *Palm Pre*, що базується на мобільній платформі *webOS*.

Для спрощення портування движок використовує модульну систему залежних компонентів: підтримує різні системи рендерингу (*Direct3D, OpenGL, Pixomatic*; раніше підтримувалися *Glide API, S3 Metal, PowerVR SGL*), відтворення звуку (*EAX, OpenAL, DirectSound3D*; раніше підтримувалися *A3D*), засоби голосового відтворення тексту, розпізнавання мовлення (тільки для *Xbox360, PlayStation 3, Nintendo Wii i Microsoft Windows*, також планувалося для *Linux i Mac*), модулі для роботи з мережею й підтримка різних пристроїв вводу.

Рушій має зручний та простий інтерфейс, що вдосконалюється з кожною новою версією. (Рис.3.5)

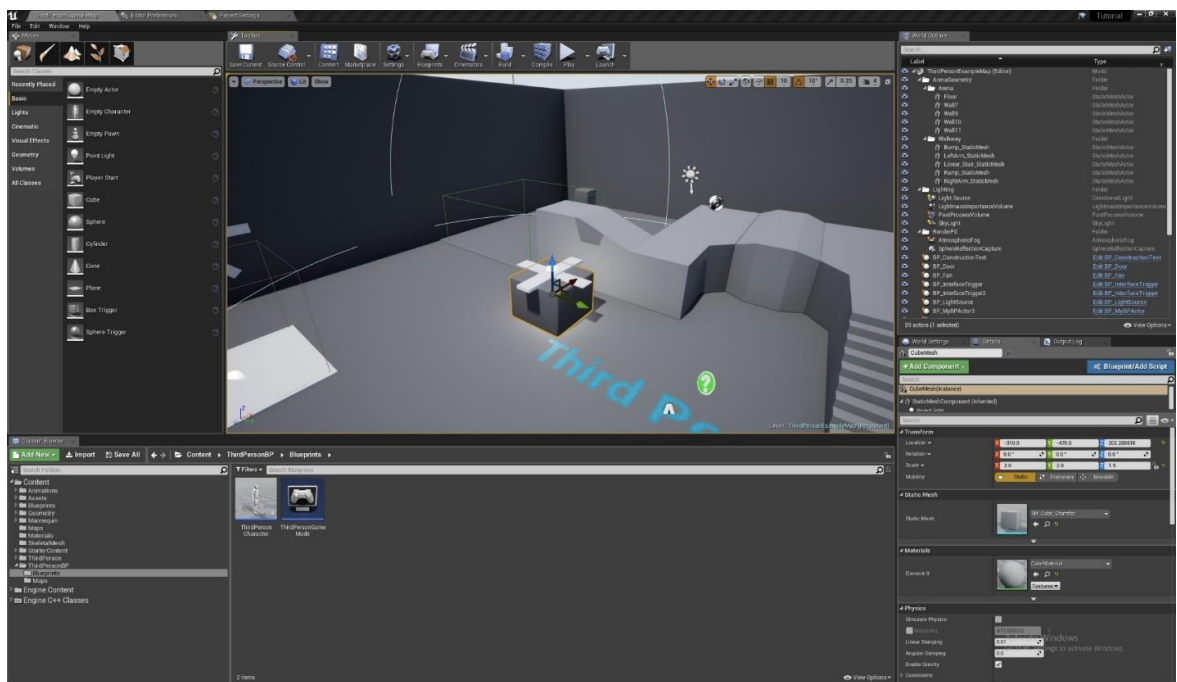


Рис.3.5 – Інтерфейс *Unreal Engine 4*.

					КРБ.КІ.1.147-03.3.5	Арк. 54
Змн.	Арк.	№ докум.	Підпис	Дата		

Для гри у мережі підтримуються технології *Windows Live*, *Xbox Live*, і *GameSpy*, що дає можливість підключити до 64 гравців (клієнтів) одночасно. Попри те, що офіційно засоби розробки не містять у собі підтримки великої кількості клієнтів на одному сервері, рушій використовувався для створення *MMORPG*-ігор. Один з найвідоміших представників жанру, *Lineage II*, використовує рушій *Unreal Engine*.

Крім того, *Epic Games* придбала *Quixel*, що має величезну бібліотеку «фотограмметричної» тобто реалістичної графіки, яку можна використовувати для створення анімації та відеоігор. Користувачі *Unreal Engine* можуть безкоштовно використовувати інструменти, що входять до комплекту *Quixel (Bridge, Mixer)* та всі ресурси бібліотеки *Quixel Megascans* (Рис.3.6).



Рис.3.6 – Набір зразків бібліотеки *Quixel*

За допомогою вбудованого редактору візуальних ефектів *Cascade*, з можливістю налаштування системи частинок з використанням різних модулів. Також рушій має редактор матеріалів, який використовує штучне затінення та дає повний контроль над зовнішнім виглядом об'єктів та персонажів. Використовується широкий набір анімаційних інструментів для редагування сі-

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

тки та анімації. Отриманий результат роботи можливо відразу подивитись та при потребі редагувати.

Для створення відеороликів та різних сцен з анімацією вбудовано *Sequencer*, інструмент, що дозволяє налаштовувати персонажів, камеру та освітлення.

Unreal Engine також працює з режимом віртуальної реальності, з розширеними настройками управління рухом. Компанія-розробник *Epic Games* співпрацює із світовими лідерами розробки програмного та апаратного забезпечення, тому *Unreal Engine* має високу якість взаємодії з системами віртуальної та доповненої реальності. Розширена система штучного інтелекту дозволяє налаштовувати реалістичну поведінку персонажів гри.

Офіційний сайт рушія має велику кількість документації та навчальної інформації для починаючих розробників. Також присутній магазин який має різні матеріали для розробки: моделі персонажів на об'єктів, скрипти, звуки, анімації та різні додаткові плагіни до рушію.

З 2015 року *Unreal Engine* має вільну ліцензію та може бути використаний ким завгодно, але якщо проект розроблений на базі рушію приносить прибуток більш ніж 3 тисячі доларів за квартал, розробник має перераховувати 5% від прибутку компанії *Epic Games*.

Рушій *Unreal Engine* більш підходить до глобальних розробок ігор у стилі 3D, ніж 2D, але все ж таки має широкий вибір платформ для готових проектів та розробки нових.

Так як для реалізації даного нашого проекту головними критеріями вибору середі розробки є вільна ліцензія, навчальна документація, зручний та зрозумілий інтерфейс, низькі системні вимоги та інструменти для розробки 2D проектів, вибором стане ігровий рушій *Unity*.

					КРБ.КІ.1.147-03.3.5	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Реалізація програмного коду у *Unity* відбувається за допомогою *Microsoft Visual Studio*, яка має можливість інтеграції додаткового функціоналу для *Unity*.

Visual Studio Community 2019 – інтегрована середа розробки програмного забезпечення розроблена та видана компанією *Microsoft*. *Visual Studio* включає у себе редактор вихідного коду та має можливість рефакторінгу. Також існує редактор форм для створення графічного інтересу додатків. Дана середа розробки є зручним інструментом програмування та відмінно підходить для написання коду.

3.2 База даних

Для реалізації поставленої мети був обраний двигун *Unity*, тому для створення програмного продукту не обиралося жодної реляційної бази даних. Всі файли та компоненти гри структуровані та зберігаються в робочій папці з назвою гри, яка створюється на комп'ютері через *Unity*, та вміщує у себе усі файли гри.

Представлено вигляд робочої папки, де зберігаються усі файли гри. (Рис.3.7)

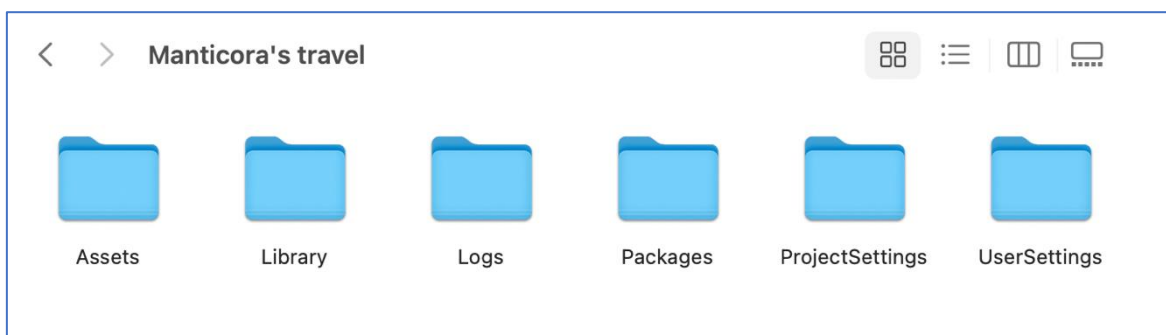


Рис.3.7 – Вигляд бази даних ПК

Також представлено, як виглядає ієрархічна структура усіх файлів, що використовуються в грі, якщо проглядати її в середовищі розробки *Unity*. (Рис.3.8)

					КРБ.КІ.1.147-03.3.5	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

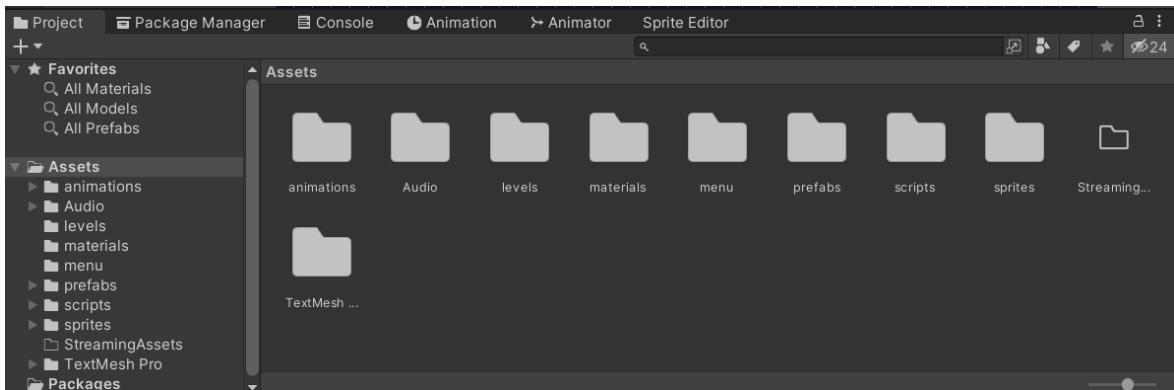


Рис.3.8 – Вигляд бази даних в *Unity*

Тут усі файли сортировано у папки по своєму функціоналу, наприклад, у папці *Assets* зберігаються усі папки та файли, які відносяться до спрайтів персонажів, ворогів, рівнів та інш.

Також варто уточнити, що всі папки, які можна побачити в БД - стандартні і присутні в будь-якому проекті на даному движку. Використовувати їх чи ні повністю залежить від користувача.

Так як усі елементи БД мають дуже зрозумілі назви та сортировку, то будь-який користувач, який зайде в БД, зможе зрозуміти, що до чого і де знаходиться, відкрита папка з спрайтами для побудови героя. (Рис.3.9)

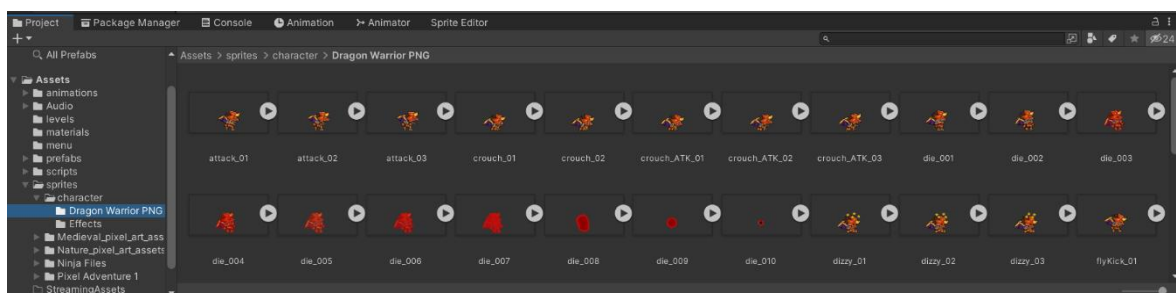


Рис.3.8 – Вигляд папки *Sprites* з відкритою папкою спрайтів героя

У папці *Sprites* також є папка, яка містить у собі все для створення рівнів. (Рис.3.10)

					КРБ.КІ.1.147-03.3.5	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

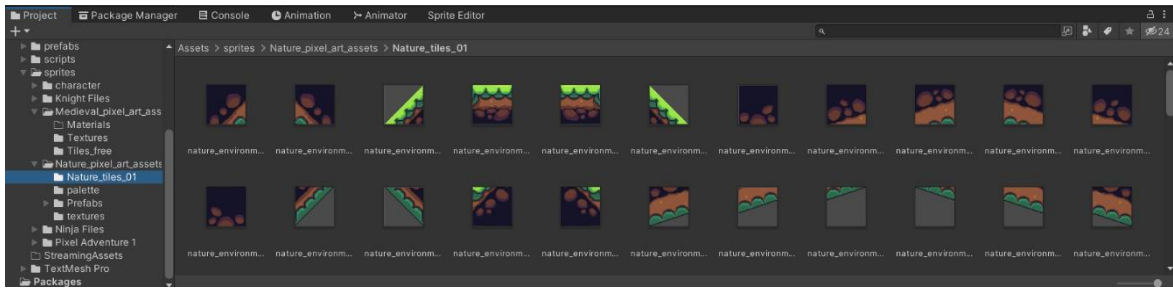


Рис.3.10 – Вигляд папки *Sprites* з відкритою папкою спрайтів для побудови рівнів

У папці *Scripts* зберігаються усі скрипти, які застосовані у цьому проєкті. Вигляд папки можна побачити нижче (Рис.3.11).

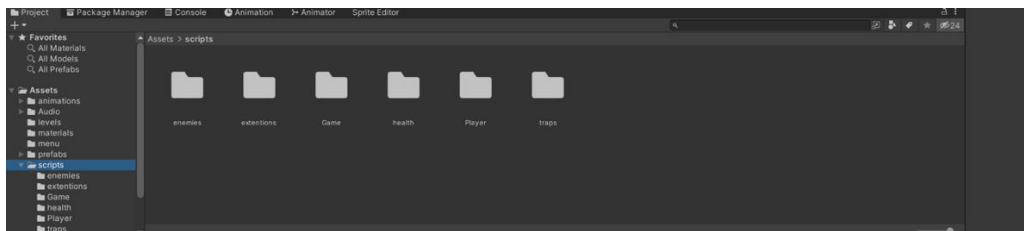


Рис.3.11 – Вигляд папки *Scripts*

3.3 Створення основної сцени

Спочатку створимо задній фон. Для цього на сцену додамо *Canvas* (у вікні *Hierarchy* вибираємо *Create -> UI -> Canvas*) і перейменуємо його на *Tiles*. Щоб він був постійно на фоні, у вкладці *Inspector* міняємо *Render Mode* на *Screen Space – Camera* і перетягуємо камеру (*Main Camera*) на *Render Camera* в вікні *Inspector*.

Додамо на *Tiles* картинку (у вкладці *Hierarchy* натискаємо праву кнопку мишки (ПКМ) на *Tiles -> UI -> Image*) і перейменуємо на *Platforms*. Щоб картинка розтягувалася по всій площі камери, вибираємо у вікні *Inspector* вкладку *Middle* і затискаючи *Alt* тиснемо на квадратик праворуч внизу.

Виділяємо об'єкт *Platforms* і перетягуємо зображення першого шару з папки *Assets - Tiles* в вікно *Inspector* на поле *Source Image*. Копіюємо цей шар, перейменовуємо на *Bg* і переносимо на нього зображення 2-го шару. Так само робимо з іншими шарами.

					КРБ.КІ.1.147-03.3.5	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

Перш за все, необхідно додати до сцени платформу, за якою буде пересуватися герой і його вороги. Спочатку додамо спрайт платформи, для цього перетягуємо його з папки в вікно проекту, після чого він повинен з'явитися серед всіх об'єктів вже доданих в гру. Після додаємо платформу в поточну сцену. Це можна зробити перетягнувши з вікна Проекту в вікно Сцени спрайт платформи. Далі додамо до об'єкта 2D коллайдер. Він необхідний для того, що б двигунець розумів межі об'єкта і гравець міг з ним взаємодіяти. Щоб додати коллайдер до об'єкта, необхідно виділити його у вікні ієрархії, а потім додати компонент коллайдера у вікні *Inspector*.

Для налаштування розмірів коллайдера потрібно скористатися кнопкою *Edit Collider*. Скріншот налаштувань платформи представлено нижче. (Рис.3.12)

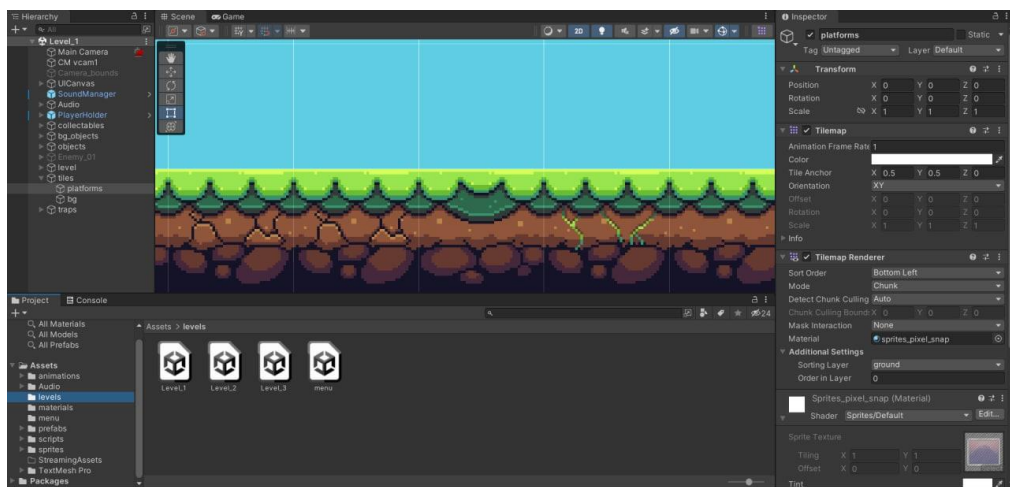


Рис.3.12 – Параметри сцени

3.4 Створення головного героя

Для створення головного герою додаємо двовимірний об'єкт, і додаємо до нього зображення героя. Для реалізації героя нам знадобиться коллайдер, а також компонент *Rigidbody 2D*, через нього буде налаштовуватися фізична модель об'єктів. Встановлюємо прапорець компонента *Rigidbody 2D Freeze Rotation* по осі Z, щоб гравець не завалювався в сторони. Скріншот

					КРБ.КІ.1.147-03.3.5	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

налаштувань головного героя представлено нижче. (Рис.3.13)

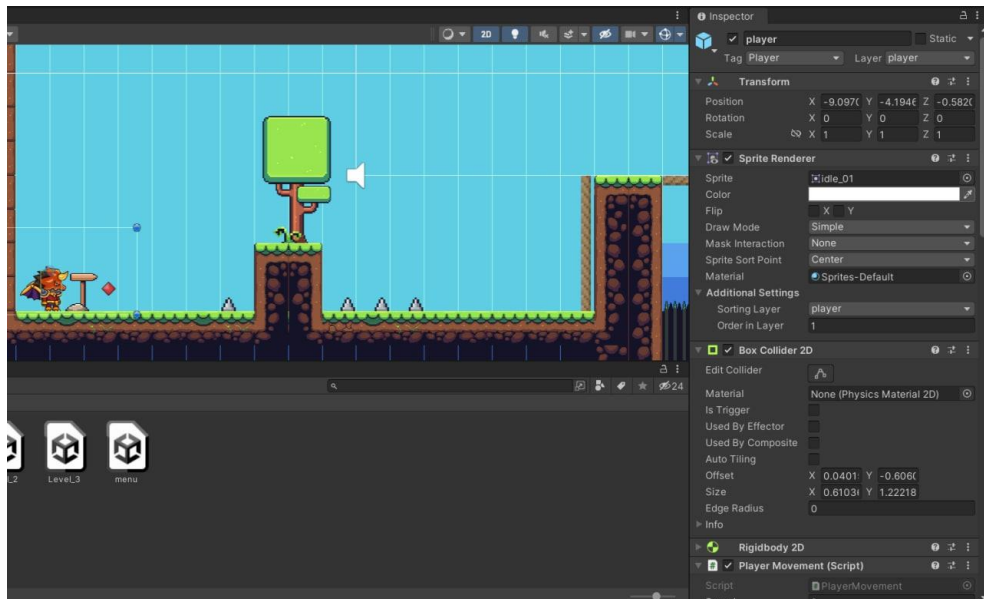


Рис.3.13 – Параметри героя

Створюємо клас *Unit* (батьківський клас для всіх класів персонажів даного курсового проекту). В папці *Assets* → *Scripts* створюємо сценарій *Unit*:

Так як головний герой повинен рухатися і здійснювати різні дії, необхідно створити сценарій на мові *C#*. Це можна зробити декількома способами: створити у вікні Проекту, або створити відразу на необхідному об'єкті у вікні Інспектора. Коли сценарій буде створений, відкриваємо його і після цього повинен запуснитися *Microsoft Visual Studio*.

У коді вже будуть підключені основні бібліотеки, а так же створений стандартний метод *Update*. Записуємо код для руху персонажем. Після чого додаємо його до нашого героя.

Для реалізації механіки стрибку було написано такий код:

```
private void Jump()
{
    if (isGrounded())
    {
        body.velocity = new Vector2(body.velocity.x, jumpPower);
        anim.SetTrigger("jump");
    }
    else if (onWall() && !isGrounded())
    {

```

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (horizontalInput == 0)
        {
            body.velocity = new Vector2(-
Mathf.Sign(transform.localScale.x) * 10, 0);
            transform.localScale = new Vector3(-
Mathf.Sign(transform.localScale.x), transform.localScale.y,
transform.localScale.z);
        }
        else
            body.velocity = new Vector2(-
Mathf.Sign(transform.localScale.x) * 3, 6);

        wallJumpCooldown = 0;
    }
}

```

Створюємо анімацію героя. У *Hierarchy* вибираємо *Character* і відкриваємо вікно *Animation (Window -> Animation)*. Натискаємо *Create*, обираємо шлях *Animation -> Character* (з'являється вікно вибору папки, в папці *Assets* створюємо папку *Animation*, в ній папку *Character* і зберігаємо туди налаштування анімації спокою персонажа *Idle*).

Натискаємо *Add Property -> Sprite -> Spite Renderer* і натискаємо на + навпроти *Sprite*.

З папки *Assets -> Character -> Idle* переносимо спрайт в вікно *Inspector*, вкладка *Sprite Renderer-> Sprite*. У *Animation* в середині шкали часу додаємо кадри по черзі, на останній ставимо 1й кадр. Аналогічно створюємо анімацію *Run i Jump* для бігу та стрибків персонажа.

Далі необхідно налаштувати переходи між анімаціями. Відкриваємо вікно *Animator*. Створюємо параметр переходу з назвою *State*. Задамо код для кожного з станів персонажа: *Idle = 0, Run = 1, Jump = 2*. Зі стану *Any State* задаємо перехід в стан *Idle* (ПКМ миші по *Any State-> Make Transitions*), тиснемо на стрілку і вибираємо умова переходу – *Conditions*, ставимо *State Equals = 0* і прибираємо галочку у вкладці *Settings -> Can Transition To Self*. Далі робимо такий же перехід від *Any State* до *Run, Equals = 1*, та до *Jump, Equals = 2*.

					КРБ.КІ.1.147-03.3.5	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаємо герою можливість стріляти. Створимо кулю. Створюємо порожній об'єкт з назвою *Bullet*, додамо до нього дочірній об'єкт спрайт (ПКМ → *2D Object-> Sprite*) і додаємо до нього картинку з папки *Sprites*. Додамо об'єкту *Bullet* колайдер *CircleCollider* (*Add Component-> Physics 2D-> Circle Collider*). Активуємо для колайдера тригер (Встановлюємо чекбокс *Is Trigger*). Створюємо сценарій для роботи кулі. Повний сценарій наданий у додатку. Перенесемо сценарій *Bullet* в вікно *Inspector* на об'єкт *Bullet*. Робимо з кулі префаб: перенесемо об'єкт *Bullet* в папку *Resources*.

Видимість і перекриття ігрових об'єктів визначається шарами, на яких вони розташовані. Чим більше значення у властивості компоненти *Sprite Renderer* → *Order in Layer* об'єкта *Sprite*, тим ближче будуть відображатися об'єкти. Поставимо у персонажа властивість *Order in Layer* компонента *Sprite* в значення 100, у кулі в 200.

В ході гри об'єкти *Bullet* будуть створюватися з префаба при натисканні клавіші пострілу.

3.5 Налаштування камери

Налаштовуємо управління камерою. Щоб камера переміщалася разом з персонажем, створюємо сценарій *Assets* → *Scripts-> CameraController*. Перенесемо сценарій *CameraController* на *Main Camera* і в вікні *Inspector* → *Camera Controller (Script)* → *Target* виберемо *Character*. Скріншот налаштувань камери представлено нижче. (Рис. 3.14)

					КРБ.КІ.1.147-03.3.5	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

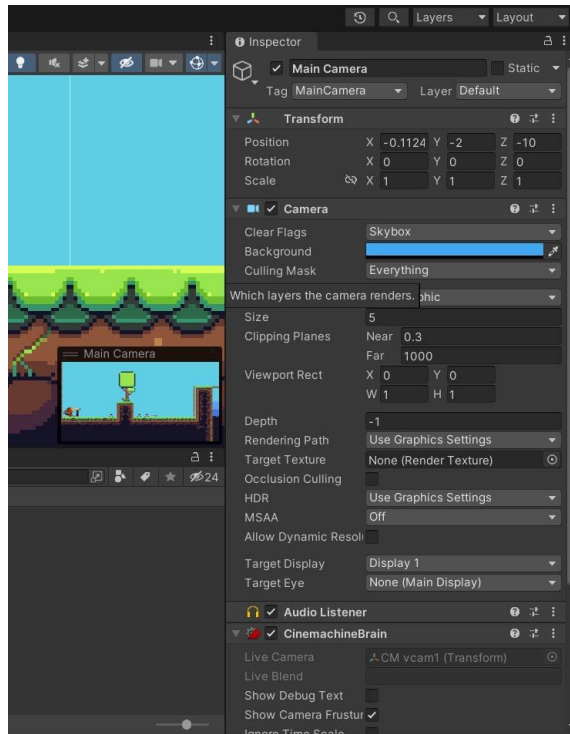


Рис.3.14 – Параметри налаштування камери

3.6 Створення перешкод, монстрів та боса

Створюємо перешкоду. Створюємо об'єкт з назвою `Trap_Damage`, додаємо йому спрайт з папки `Assets-> Sprites-> Obstacles` і додамо об'єкту `Obstacle` коллайдер `BoxCollider 2D`, у коллайдера вмикаємо опцію `Is Trigger`. Для об'єкта `Obstacle` створюємо сценарій `Assets-> Scripts-> Obstacle.cs` і переносимо сценарій на об'єкт. З об'єкта `Obstacle` робимо префаб. Скріншот налаштувань перешкоди представлено нижче. (Рис.3.15).

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

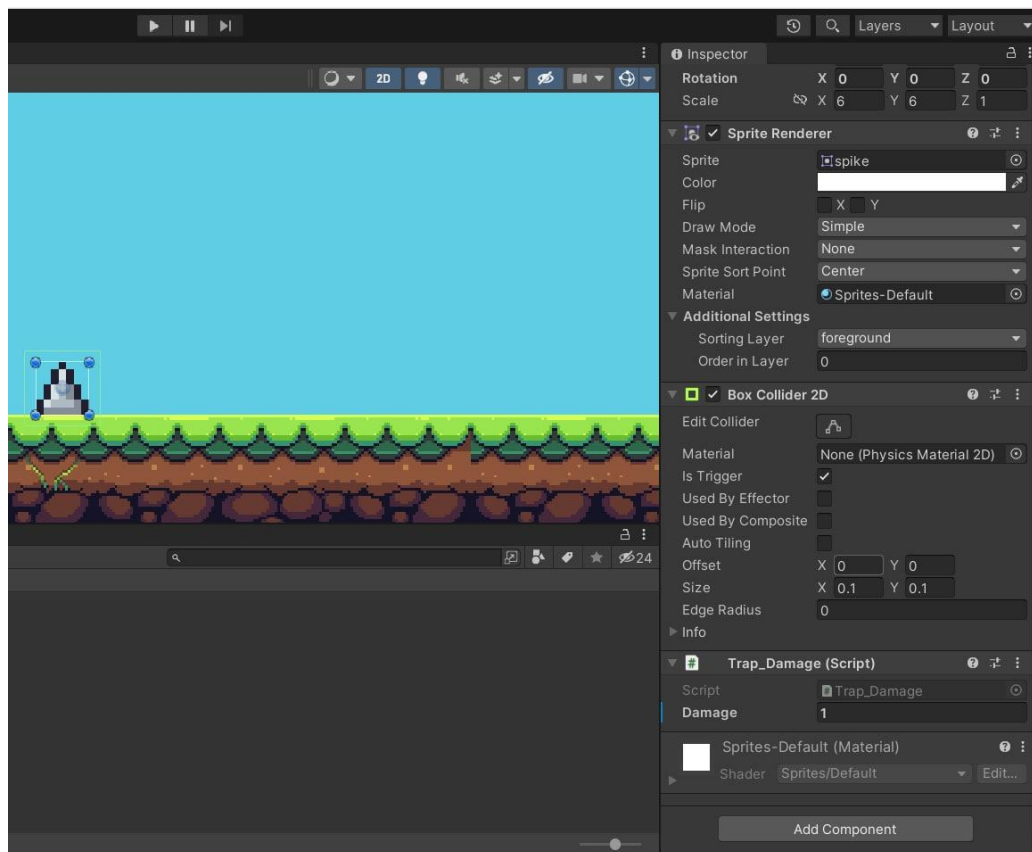


Рис.3.15 – Перешкоди

Для реалізації взаємодії з об'єктами було написано наступний вихідний

КОД:

```
public class Trap_Damage : MonoBehaviour
{
    [SerializeField] protected float damage;
    protected void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.tag == "Player" && damage > 0)
            collision.GetComponent<Health>().TakeDamage(damage);
    }
}
```

Створимо рухомого ворога. Створюємо сценарій *Assets-> Scripts->Spike_head, Saw*. Повний сценарій наданий у додатку. Створюємо об'єкт *Monster*, додаємо йому спрайт з папки *Assets*. Додаємо об'єкту компоненти *Rigidbody 2D* і *BoxCollider 2D*. У вікні *Inspector* натискаємо *IsTrigger*. У компоненти *Rigidbody2D* властивість *BodyType* ставимо в *Kinematic*, пересувати цього противника будемо з коду. Переносимо на об'єкт сценарій *Spike_head, Saw*.

					КРБ.КІ.1.147-03.3.5	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Створюємо анімацію спокою. Для цього виділяємо об'єкт *Spike_head*, *Saw*, відкриваємо вікно *Animation* (*Window-> Animation*), створюємо кнопку *Create*. Для збереження анімації у вікні запиту збереження створюємо папку *Assets-> Animation-> Spike_head, Saw* і зберігаємо в файл *Idle*. Натискаємо *Add Property*, вибираємо

Sprite-> Sprite Renderer-> Sprite. Перетягуємо спрайт руху на часову шкалу в вікні *Animation*. Скріншот налаштувань представлено нижче. (Рис.3.16-3.17)

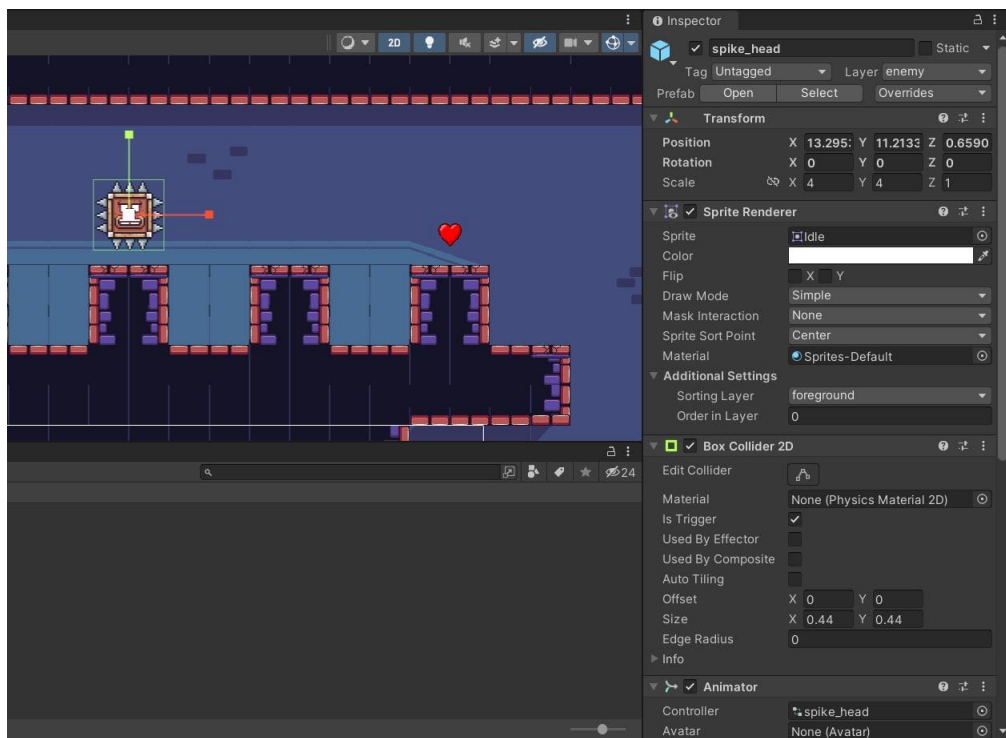


Рис.3.16 – Налаштування платформи

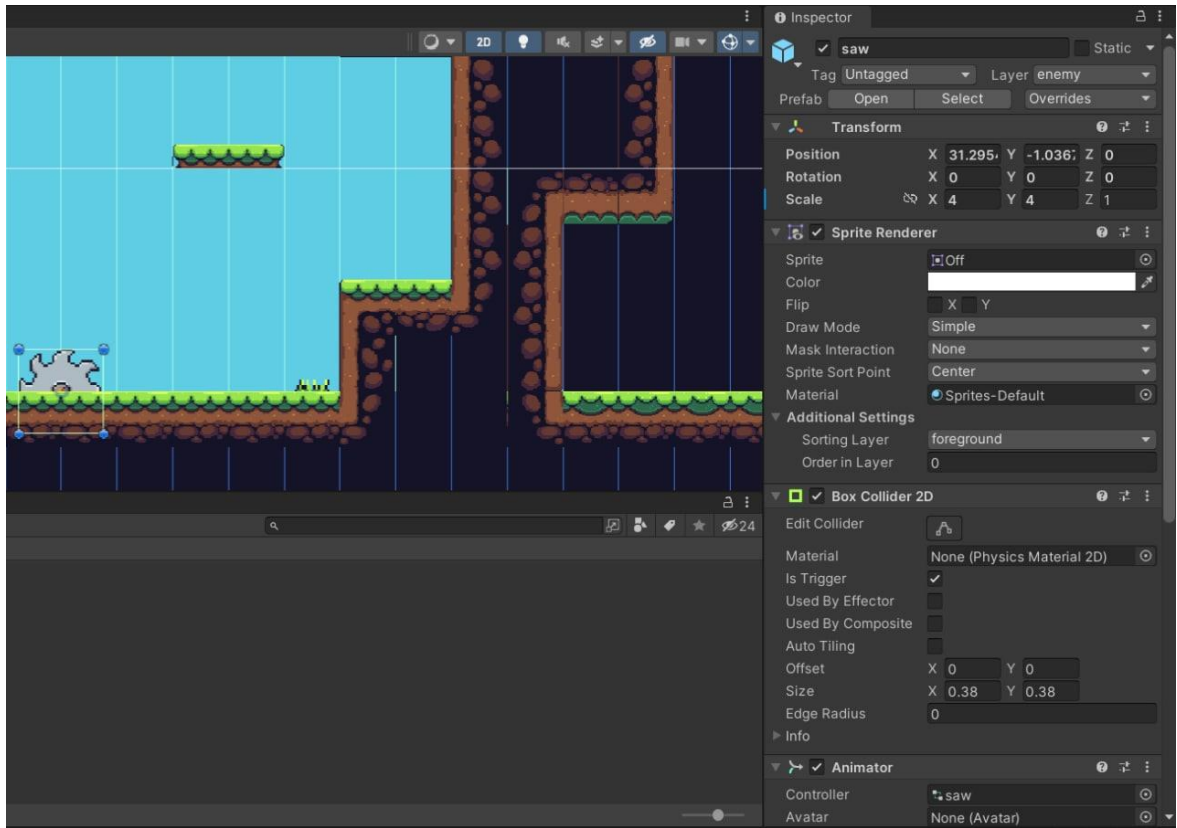


Рис.3.17 – Налаштування леза

Код виглядає так:

```
private void Update ()
{
    if (movingLeft)
    {
        if (transform.position.x > leftEdge)
        {
            transform.position = new
            Vector3 (transform.position.x - speed * Time.deltaTime,
            transform.position.y, transform.position.z);
        }
        else
            movingLeft = false;
    }
    else
    {
        if (transform.position.x < rightEdge)
        {
            transform.position = new
            Vector3 (transform.position.x + speed * Time.deltaTime,
            transform.position.y, transform.position.z);
        }
        else
            movingLeft = true;
    }
}
```

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

Створимо наступного стріляючого рухомого ворога. Створюємо об'єкт *MeleeEnemy*, додаємо йому спрайт з папки *Assets*, робимо анімацію *Run*. Додаємо об'єкту *BoxCollider 2D*.

У вікні *Inspector* натискаємо *IsTrigger*. Переносимо на об'єкт сценарій *MeleeEnemy*. Щоб монстр стріляв іншими кульками, в об'єкті *MeleeEnemy* міняємо колір у вікні *Inspector* -> *Bullet Color*. Скріншот налаштувань рухомого ворога представлено нижче. (Рис.3.18)

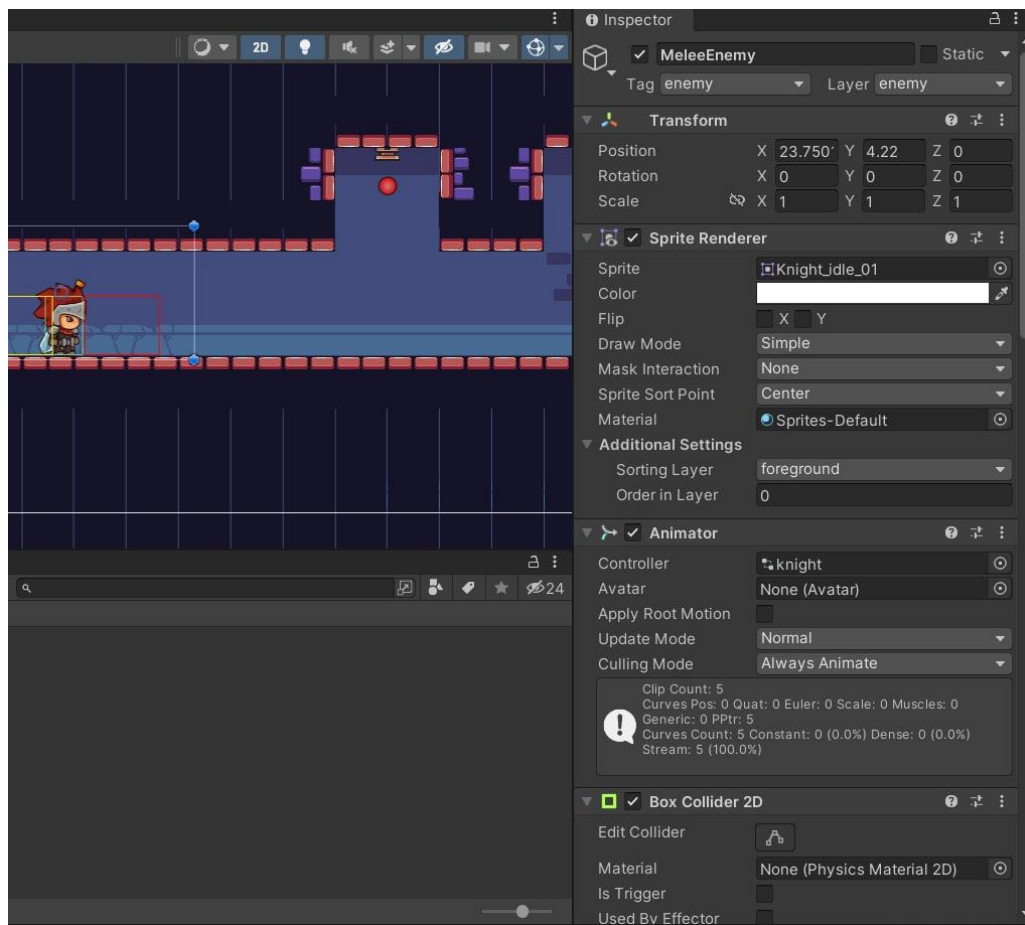


Рис.3.18 – Налаштування ворога

Для реалізації атаки ворога було написано наступний вихідний код:

```
private void Attack()
{
    SoundManager.instance.PlaySound(fireballSound);
    anim.SetTrigger("attack");
    cooldownTimer = 0;

    fireballs[FindFireball()].transform.position = firePoint.position;
}
```

					КРБ.КІ.1.147-03.3.5	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

```

fireballs[FindFireball()].GetComponent<Projectile>().SetDirection(Mathf.Sign(
transform.localScale.x));
}
private int FindFireball()
{
    for (int i = 0; i < fireballs.Length; i++)
    {
        if (!fireballs[i].activeInHierarchy)
            return i;
    }
    return 0;
}
}

```

Створимо головного боса. Створюємо сценарій *Assets-> Scripts-> RangeEnemy*. Повний сценарій наданий у додатку. Створюємо об'єкт *Boss*, додаємо йому спрайт з папки *Assets*. Додаємо об'єкту компоненти *Rigidbody 2D* та два *CircleCollider 2D*. У вікні *Inspector* натискаємо *IsTrigger*. У компоненті *Rigidbody2D* властивість *BodyType* ставимо в *Dynamic*, пересувати цього противника будемо з коду. Переносимо на об'єкт сценарій *RangeEnemy*.

Створюємо анімацію руху для *Boss*. Для цього виділяємо об'єкт *RangeEnemy*, відкриваємо вікно *Animation (Window-> Animation)*, створюємо кнопку *Create*. Для збереження анімації у вікні запиту збереження створюємо папку *Assets-> Animation-> RangeEnemy* і зберігаємо в файл *Run*. Натискаємо *Add Property*, вибираємо *Sprite-> Sprite Renderer-> Sprite*.

Перетягуємо спрайт руху на часову шкалу в вікні *Animation*. Скріншот налаштувань рухомого ворога представлено нижче. (Рис.3.19)

					КРБ.КІ.1.147-03.3.5	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

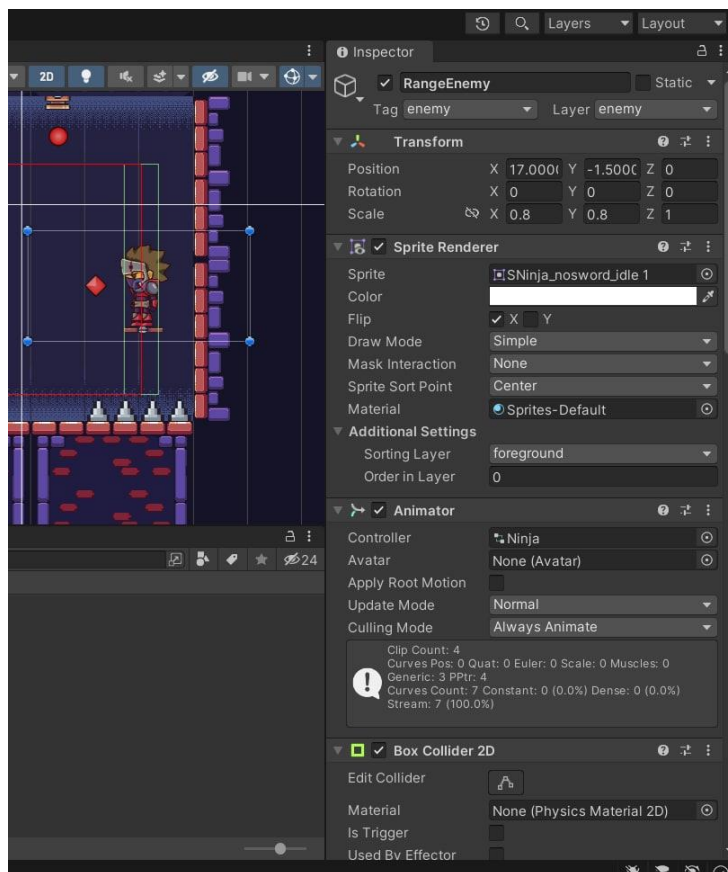


Рис.3.19 – Налаштування боса

3.7 Створення бонусів та переходів між рівнями

Далі створюємо життя. Для цього переносимо на сцену спрайт енергії, додамо йому *Box Collider 2D* (*Add Component* -> *Physics 2D*->*Box Collider2D*). Створюємо сценарій *Assets-> Scripts-> Heat*. Повний сценарій наданий у додатку. Скріншот налаштувань бонусних життів представлено нижче. (Рис.3.20)

Для переходу між сценами створимо простий об'єкт (у вікні *Hierarchy* вибираємо *Create* -> *Empty*). Додати йому *Box Collider2D* (*Add Component* -> *Physics 2D*->*Polygon Collider2D*) та встановити прапорець *Is Trigger*. Створюємо сценарій *Assets-> Scripts-> Door*. Скріншот налаштувань переходу між рівнями представлено нижче. (Рис.3.21)

					КРБ.КІ.1.147-03.3.5	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

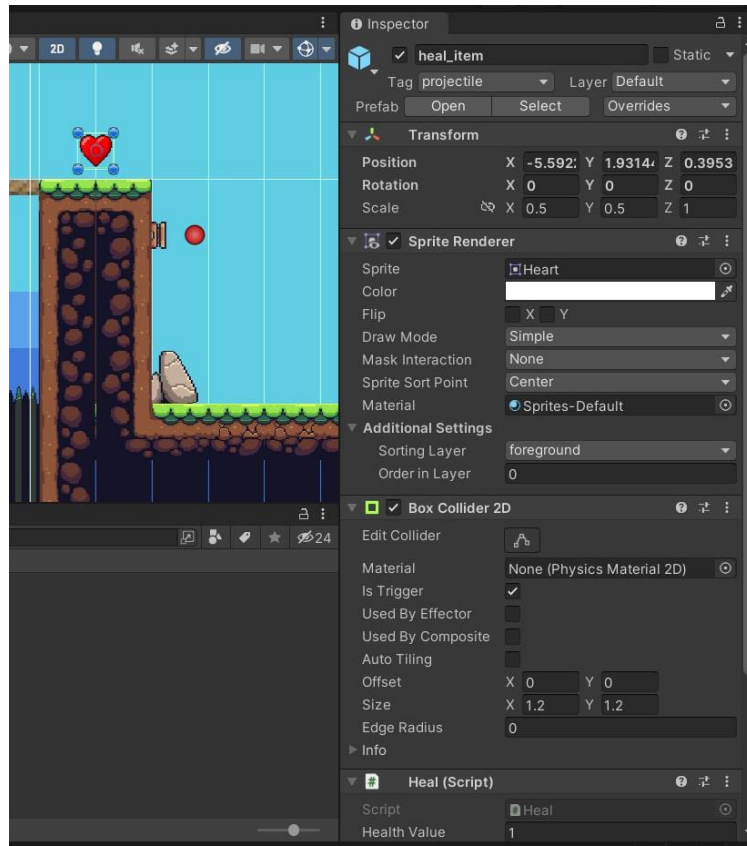


Рис.3.20 – Налаштування життя

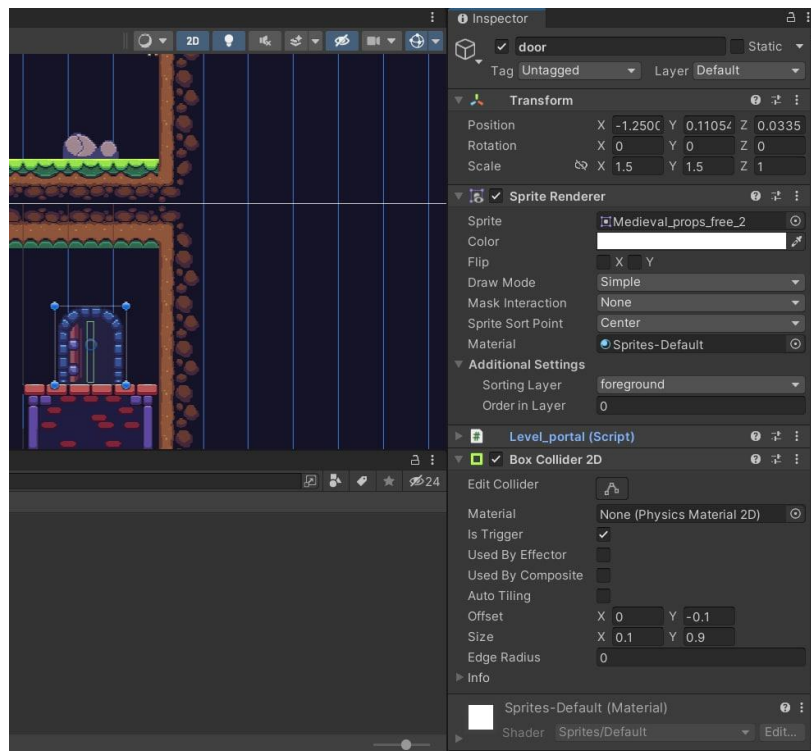


Рис.3.21 – Налаштування об'єкту переміщення між рівнями

3.8 Додавання аудіоефектів

Для приєднання фонової музики необхідно на кожну сцену додати компонент *Audio Source*(*Audio* ->*Audio Source*, та додати у поле *AudioClip* музичний файл. Скріншот представлено нижче. (Рис.3.22)

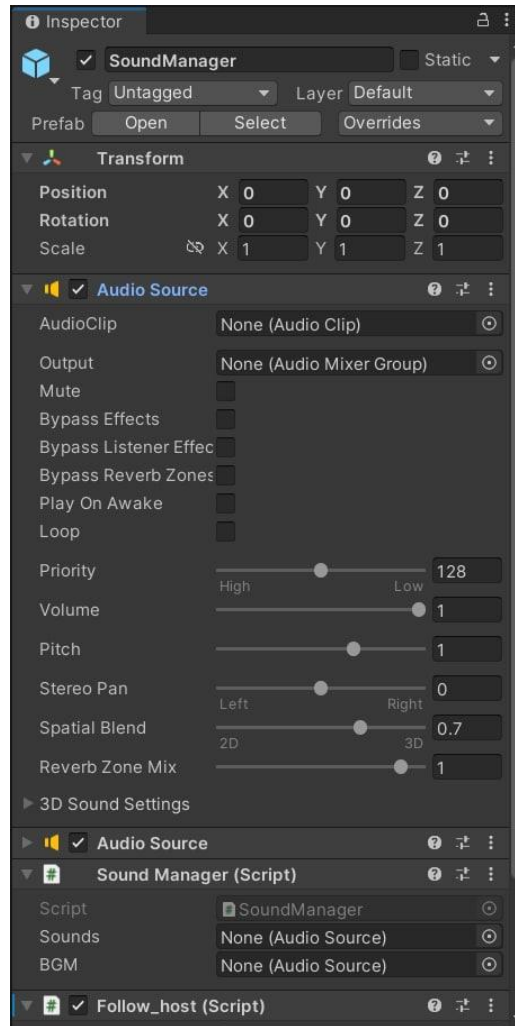


Рис.3.22 – Налаштування музичного супроводу

3.9 Створення інтерфейсу користувача та системи управління грою

Для початку створюємо *Canvas* (у вікні *Hierarchy* вибираємо *Create* -> *UI* -> *Canvas*). Додамо на *Canvas* дві панелі *Menu* (у вікні *Hierarchy* вибираємо *Create* -> *UI* -> *Panel*). Створюємо і додаємо їм сценарій *Assets*-> *Scripts*->

Menu. В *Menu* створюємо *Panel* і в цій панелі створюємо об'єкт *Text* (у вікні *Hierarchy* вибираємо *Create -> UI -> Text*) та перейменовуємо його в *Menu*, робимо налаштування за бажанням. Створюємо кнопку *Start Game* (у вікні *Hierarchy* вибираємо *Create -> UI -> Button*), *Controls* та *Exit*. (Рис.3.23-3.25).

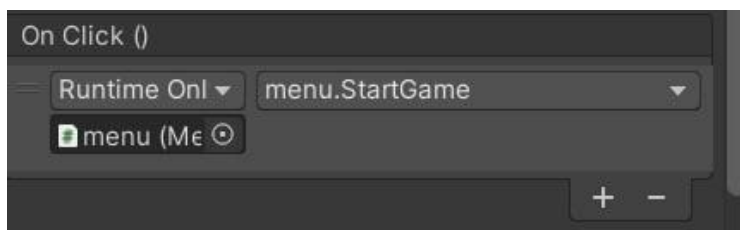


Рис.3.23– Налаштування кнопки *Start Game*

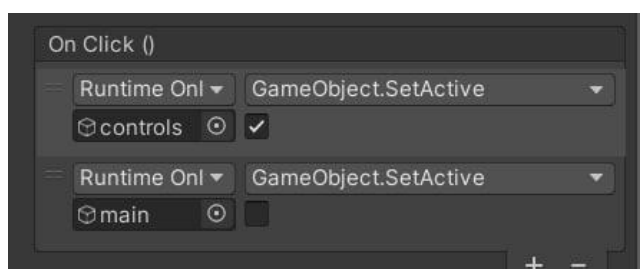


Рис.3.24– Налаштування кнопки *Controls*

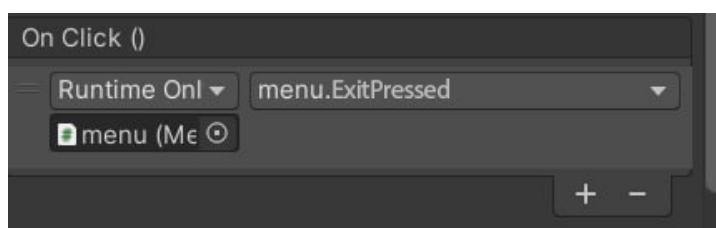


Рис.3.23– Налаштування кнопки *Exit*

3.10 Опис рівнів

Перший рівень має простий дизайн и містить в собі шипи, платформа та леза. Зображення першого рівня наведено нижче. (Рис.3.24)

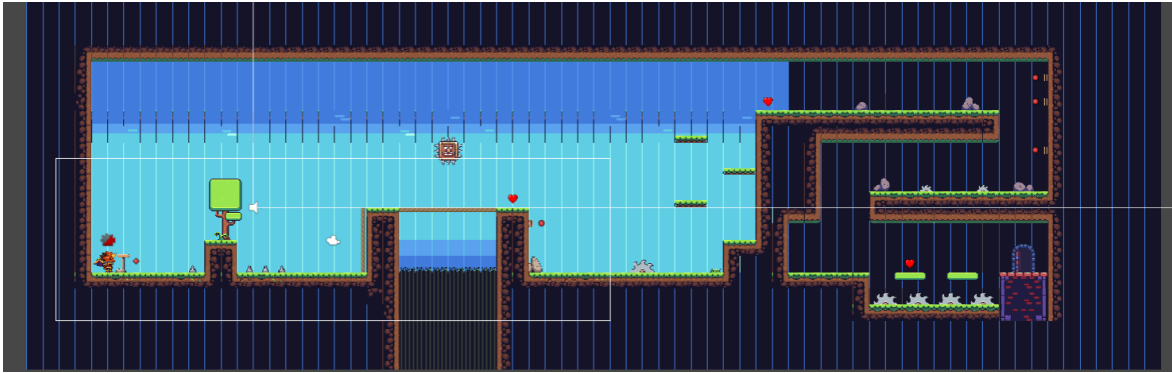


Рис.3.24 – Перший рівень

Другий рівень має дещо ускладнений дизайн, тут з'являється перший ворог. Схему рівня представлено нижче. (Рис.3.25)

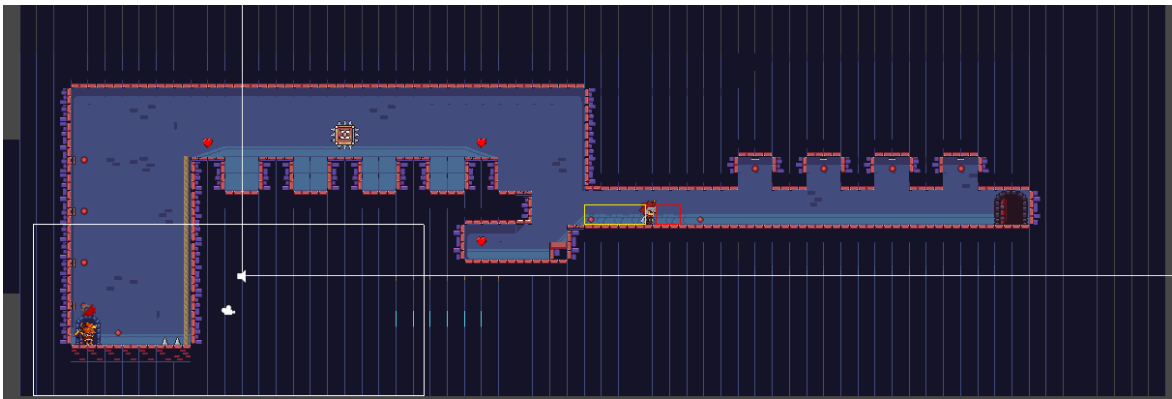


Рис.3.25 – Другий рівень

Третій рівень представляє собою тунель, що являє собою арену, потрібно здолати боса, щоб виграти гру. Схему рівню представлено нижче. (Рис.3.26)

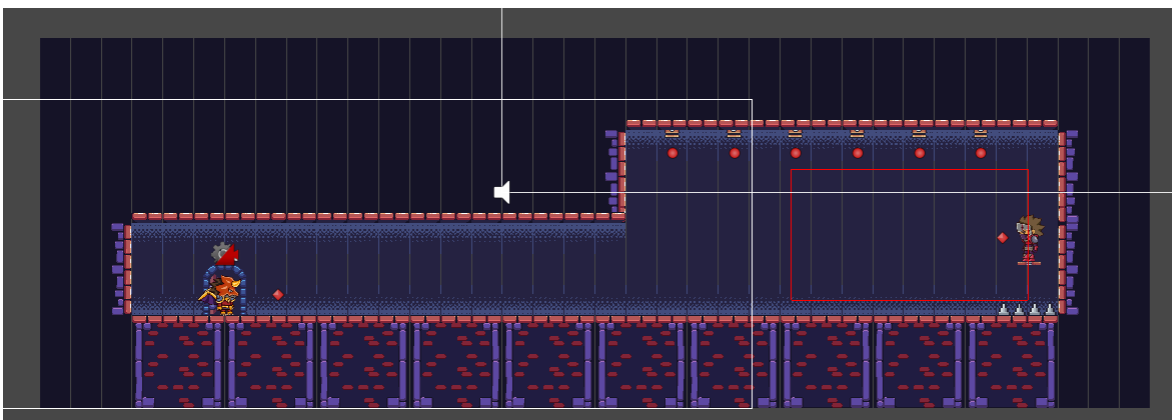


Рис.3.26 – Третій рівень з Босом

3.11 Тестування та відлагодження гри

Основною функцією тестування ігор є виявлення та документування дефектів програмного забезпечення (або помилок). Тестування інтерактивного програмного забезпечення для розваг – це високотехнічна галузь, яка вимагає комп’ютерних знань, аналітичної компетентності, навичок критичного оцінювання та витривалості.

Забезпечення якості є найважливішим компонентом у розробці ігор, хоча індустрія відеоігор не має стандартної методології. Натомість розробники та видавці мають власні методи. Маленькі розробники, як правило, не мають персоналу з контролю якості; однак великі компанії можуть наймати команди з контролю якості на повний робочий день. Гучні комерційні ігри професійно та ефективно перевірені відділом контролю якості видавця.

Тестування починається відразу після написання першого коду і збільшується в міру просування гри до її завершення. Основна команда контролю якості буде стежити за грою з моменту її першого подання до контролю якості до пізньої публікації. На початку процесу розробки ігор команда тестування невелика і зосереджується на щоденних відгуках про новий код. У міру наближення гри альфа етапі, працевлаштовується більше членів команди і складаються плани випробувань. Іноді функції, які не є помилками, повідомляються як помилки, а іноді програміст не вдається вирішити проблеми вперше. Хороша система повідомлення про помилки може допомогти програмістам працювати ефективно. По мірі надходження проектів бета-версія етап, команда тестування матиме чіткі завдання на кожен день. Відгуки тестувальників можуть визначати остаточні рішення про виключення або включення остаточних ознак. Представлення тестерів із новими перспективами може допомогти виявити нові помилки. На цей момент провідний випробувач щодня спілкується з виробником та керівниками відділів

					КРБ.КІ.1.147-03.3.5	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

Щоб вдосконалитися, що гра працює без помилок потрібно скласти тест-план:

Таблиця 3.1

Тест-план

№	Призначення	Дія	Очікуваний результат	Отриманий результат	Підсумок
1	Перевірка на коректне реагування персонажу на команди, що надходять, про пересування	Використовуючи клавіші руху, пробігти частину рівня	Персонаж буде рухатися в належному напрямку, коректне програвання анімації	Персонаж пересувався в потрібному напрямку, коректне відображення анімації	Пройдено
2	Перевірка на коректний стрибок	Використовуючи клавішу стрибка, підстрибнути кілька разів поспіль	Гравець може зробити стрибок тільки з землі, коректне програвання анімації	Персонаж здатний зробити стрибок тільки з землі, вірно виконання анімації	Пройдено
3	Перевірка поведінки ворогів	Підійти до ворога на досить близьку відстань, утекти від ворога	Поки гравець перебуває поза увагою ворога, патрулює місцевість. Якщо герой попадає в поле зору противника, ворог починає його преследувати, якщо під час переслідування гравець втікає (ховається,) зупиняється на короткий час, потім повертається до точки старту.	Патрулює місцевість, якщо знаходиться в пасивному стані, коли помічає гравця, переслідує його, якщо гравець втік, зупиняється на короткий час і потім повертається на стартову точку.	Пройдено
4	Перевірка на коректну реалізацію події програшу гравця	Дозволити противнику атакувати гравця	Коли ворог атакує героя, програється анімація смерті персонажу і анімація атаки у противника.	Ворог атакує героя, анімація програється правильно	Пройдено
5	Перевірка на коректну реалізацію події виграшу гравця	Дійти до кінця рівня	При досягненні кінця рівня з'являється меню з вибором дії	При досягненні кінця рівня зупиняється час та появляється меню	Пройдено

6	Перевірка на коректну роботу меню програшу	1) Програти та перезапустити рівень 2) Програти і вийти в головне меню	Є можливість вийти в головне меню, коли програєш	Вихід в головне меню	Пройдено
7	Перевірка на коректну роботу меню виграшу	1) Виграти та перезапустити рівень 2) Виграти і вийти в головне меню	Є можливість вийти в головне меню, коли віграєш	Вихід в головне меню	Пройдено
8	Перевірка на коректну роботу головного меню	1) Запустити рівень 2) Вийти з гри 3) Керування	Є можливість запустити рівень, або вийти з гри	Залежно від вибору запускається рівень, чи закривається гра	Пройдено
9	Перевірка на коректний збір об'єктів, що збираються.	Зібрати колекціоновані об'єкти	При зборі колекціонованих об'єктів, зібраний об'єкт знищується, життя збільшується на 1 очоко	Життя відновлюється	Пройдено

Дане тестування показує, що на даному етапі реалізації проекту, гра відповідає більшості функціональних вимог, виділених на етапі постановки задачі.

Висновки третього розділу

У ході проведеної роботи в третій частині було

- обрано ігровий рушій *Unity 2021*, як засіб для створення гри. Для створення програмного коду гри було обрано редактор *Visual Studio 2022*.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

- було описано детальну розробку головного персонажу гри. Показані головні компоненти головного та розробка основних можливостей.

Проект був зкомпільований та при його тестуванні не було виявлено жодних проблем та помилок.

За час роботи над третьою частиною проекту було реалізовано заплановані механіки для наступних ігрових об'єктів:

- персонаж (біг, стрибки, постріли);
- монстри (рух, постріли);
- інтерфейс користувача (LivesBar, Головне меню);
- інтерактивні об'єкти(SWlight, Health, Shoot);
- не інтерактивні об'єкти(фон, платформи).

Особливістю гри є мінімалістичний піксельний дизайн і різноманітні рівні з атмосферою музикою.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						78
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 4

ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ПРОЄКТУ

4.1 Основні завдання організаційно-економічного та маркетингового обґрунтування проєкту

Ігрова індустрія багатогранна та відкрита для нових ідей, технологій та досягнень. На сьогоднішній день ігри – це не тільки спосіб цікаво провести час, це яскраві емоції, враження, незабутній досвід, це спільноти, що стрімко розвиваються, це професія і це спорт. Розробка ігор – це мистецтво.

Люди грають в ігри не стільки для самої гри, скільки для досвіду, який гра створює: захоплюючий прилив адреналіну, загадкові пригоди, психологічний виклик. Люди грають в ігри, щоб створити короточасні емоції та переживання, чи подолають вони складні ігрові виклики, шукаючи позбавлення від щоденних турбот.

Метою дипломної роботи є проектування ігрового процесу і здійснення етапів розробки комп'ютерної 2D гри-платформера. Для розробки гри було обране середовище розробки Unity3D з використанням мови програмування C#. Весь процес створення можна поділити на наступні етапи:

- Постановка загальної ідеї ігри, техніко-економічне обґрунтування проєкту, вибір мови програмування та середовища розробки.
- Розробка рівнів та сетінгу. Розробка головного герою та ворогів, написання скриптів та сюжету гри.
- Програмування штучного інтелекту, налагодження ігрових механік та об'єктів на рівнях. Комплексне налагодження головного меню та підготовка до експлуатації. Налагодження та виправлення помилок (bugs).
- Тестування ітогової версії 2D гри-платформера.
- Підготовка і передача дизайнерської документації для супроводу.

По масштабу проєкт, що розробляється, можна віднести до середніх

					КРБ.КІ.1.147-03.3.5	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

проектів, які невеликі по масштабу, прості і з обмеженими об'ємами. По термінах реалізації проект буде короткостроковий (термін реалізації до 3 років). По складності проект є проектом середньої складності.

Основні завдання організаційно-економічного та маркетингового обґрунтування проекту включають в себе аналіз технічної та економічної доцільності проекту, визначення потенційних споживачів продукції, оцінку ринкового потенціалу проекту, визначення стратегії входження на ринок, розрахунок вартості проекту та оцінку його ефективності.

Організаційно-економічне обґрунтування проекту передбачає розробку організаційної структури проекту, планування витрат та прибутків, визначення ефективності використання ресурсів, розрахунок фінансових показників та визначення можливостей для залучення фінансування проекту.

Маркетингове обґрунтування передбачає дослідження ринку та конкурентного середовища, визначення потреб споживачів та їх готовності купувати продукцію, розробку маркетингової стратегії та плану маркетингових заходів.

Усі ці завдання допомагають забезпечити успішну реалізацію проекту, максимізувати його прибутковість та ефективність використання ресурсів.

Для проекту з розробки програмного продукту основні задачі організаційно-економічного обґрунтування полягають у визначенні витрат на придбання необхідного програмного забезпечення, навчання та підготовку персоналу для роботи з обраним програмним забезпеченням, оплату послуг фахівців з проектування та розробки, тестування, установки та впровадження гри, оплату витрат на електроенергію, інтернет-підключення та інші операційні витрати, пов'язані з періодом розробки, розрахунку прибутковості та ефективності проекту в довгостроковій перспективі, а також у підготовці фінансових та бізнес-планів проекту.

Щодо маркетингового обґрунтування, його головна мета - визначення потенційного попиту на 2D гру-платформера, розробка стратегії маркетингу

					КРБ.КІ.1.147-03.3.5	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

та реклами, а також визначення конкурентоспроможності гри на ринку ігрового контексту.

Крім того, організаційно-економічне та маркетингове обґрунтування проєкту включає в себе аналіз ризиків та можливостей, що пов'язані з реалізацією проєкту.

У проєкті з проєктування та розробки 2D гри-платформера, витрати в економічній частині можуть бути спрямовані на наступні напрямки:

1. Придбання необхідного програмного забезпечення для виконання розробки.
2. Навчання та підготовка персоналу для роботи з обраним програмним забезпеченням, що включає в себе курси навчання, семінари та тренінги.
3. Оплата послуг фахівців з проєктування та розробки, тестування, установки та впровадження гри
4. Оплата витрат на електроенергію, інтернет-підключення та інші операційні витрати, пов'язані з періодом розробки.

Ці витрати потрібно обґрунтовувати на етапі планування проєкту, проводячи відповідні розрахунки та аналіз ринку, щоб забезпечити максимальну ефективність витрачених коштів і забезпечити успішне виконання проєкту.

Ще одним важливим аспектом економічної частини проєкту є оцінка його прибутковості. Для цього потрібно провести аналіз витрат на проєкт та очікуваних доходів. Доходи можуть складатись з доходів від реалізації готового продукту або надання додаткових платних послуг у грі.

4.2 Обґрунтування ефективності проєкту

В умовах відкритої ринкової економіки розширюється діапазон оцінки ефективності науково-технічних розробок, а отже, збільшується кількість основних видів ефективності НДДКР, які необхідно визначити з метою цієї

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

оцінки. До них належать:

– маркетинговий ефект полягає в аналізі існуючих аналогічних розробок і порівнянні з грою, що розробляється. На сьогоднішній день виходить маса різних платформерів, які можна зарахувати і до інших жанрів. Це може говорити про те, що «чисті платформери» стали продуктом направленим на вузький круг споживачів, тому для отримання великих прибутків розробники почали додавати до цього жанру елементи сторонніх ігрових жанрів. Тому можливо зробити висновок, що для успіху гри її потрібно урізноманітнити елементами різних жанрових категорій;

– економічний ефект проєктування та розробки гри полягає в забезпеченні економічних результатів для кожного виробничого суб'єкта та в цілому для економіки країни. Економічна ефективність проєкту має відображати вплив його результативності на розвиток галузей, регіонів та підприємств, що беруть участь у впровадженні новітніх технологічних рішень;

– соціальний ефект, що відображає зміни умов діяльності людини в суспільстві. Його прояв спостерігається в змінах характеру та умов праці, підвищенні життєвого рівня населення, поліпшенні побутових умов, розширенні можливостей духовного розвитку особистості, у змінах стану довкілля. Позитивний соціальний ефект від комп'ютерних ігор, помічений у ході досліджень, полягає в наступному: чим активніше люди грають у відеоігри, тим більший інтерес виявляють до знань і навичок у галузі цифрових технологій. Так, переважна більшість геймерів, які щодня приділяють час іграм (80%) [8], виявляють бажання в майбутньому пройти додаткове навчання по роботі з інформаційними технологіями – тобто підвищити власний рівень цифрової грамотності.

Науково-технічну ефективність (НТЕ) результатів прикладних робіт визначають на основі показників науково-технічного рівня. Оцінка науково–

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

технічної ефективності НДДКР відбувається на основі показника ($O_{НТЕ}$), який представляє собою ступінь досягнення максимально можливого рівня, значення якого дорівнює 1 (одиниці):

$$O_{НТЕ} = K^{\Phi}_{НТЕ} / K^{\Pi}_{НТЕ}, \quad (4.1)$$

де $K^{\Phi}_{НТЕ}$ – показник (коефіцієнт) фактичного рівня науково-технічної ефективності;

$K^{\Pi}_{НТЕ}$ – показник (коефіцієнт) потенціально можливого рівня науково-технічної ефективності (дорівнює одиниці).

Значення показника $K^{\Phi}_{НТЕ}$ визначають на основі шкали експертних оцінок.

Таблиця 4.1

Шкала експертних оцінок для виміру рівня науково-технічної ефективності проєктів

№	Групи показників	Характеристика показників	Інтервал рейтингового числа	Коефіцієнт значущості показників
1	Науково-технічний рівень	Перевищує кращі світові аналоги	10	0,30
		Відповідає світовому рівню	7 – 9	
		Нижче кращих світових аналогів	5 – 6	
		Перевищує кращі вітчизняні аналоги	3 – 4	
		Відповідає вітчизняному рівню	1 – 2	
		Нижче вітчизняного рівня	0	
2	Перспективність	Першочергова значущість	8 – 10	0,25
		Значущий	5 – 7	
		Корисний	1 – 4	
3	Потенційний масштаб практичного використання	Світовий ринок	10	0,20
		Галузі національної економіки	7 – 9	
		Галузь (регіон)	3 – 6	
		Окремі підприємства (об'єднання)	1 – 2	
4	Ступінь вірогідності досягнення позитивних результатів	Великий	10	0,25
		Середній	5 – 9	
		Малий	1 – 4	

Примітка: об'єкт оцінки і аналог(и), які порівнюють за однаковими показниками, наведеними у співставленому вигляді відхилення в значеннях

кожного з показників, мають бути однаковими для варіантів, що порівнюються.

4.3 Оцінка науково-технічного рівня розробки

Визначають $K^{\Phi}_{НТЕ}$ на основі експертної оцінки науково-технічного рівня розробки.

З цією метою:

- розробляють перелік специфічних показників, необхідних для виміру науково-технічного рівня розробки;
- формують групу аналогів, які реалізовані на світовому і вітчизняному ринках;
- здійснюють відповідні розрахунки для співставлення показників і визначення балів по Таблиці 4.1.

До числа специфічних показників відносять:

- для нової техніки: продуктивність, споживання інженерних ресурсів на виробітку одиниці продукції, потреба в робочих, які обслуговують обладнання, експлуатаційні витрати на одиницю продукції;
- для нових матеріалів і речовин: вміст корисних речовин для виробітки готової продукції, питома вага відходів у загальному обсязі переробленої сировини, вартість одиниці нового матеріалу, додаткові витрати на екологічну компенсацію;
- для нових технологій: якість виробленої продукції, енергоємність і трудомісткість продукції, собівартість одиниці продукції.

З метою спрощення визначення $K^{\Phi}_{НТЕ}$ у Таблиці 4.2 не введено показника витрат на одиницю продукції.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.2

Порівняльні показники для виконання оцінки НТЕ

ПОКАЗНИКИ	Варіанти технології	
	розробленої	співвідносної (аналога)
Рівень новізни	Високий	Середній
Якість продукції	Висока	Середня
Споживання на 1 т продукції		
– тепла, Гкал	10	12
– електроенергії, кВт·годину	2500	2000
– води, м ³	3	4
Трудомісткість виробництва, людино-годин/ тонну	50	70

На основі співставлення даних таблиці встановлюють бали по характеристиках чотирьох груп і на цій основі розраховують значення інтегрального показника НТЕ:

$$\text{НТЕ} = \sum B_i \times K_i^3, \quad (4.2)$$

де $i = 1 \div 4$,

B_i – бали (рейтингове число),

K – коефіцієнт значущості показників.

Рівень науково-технічної ефективності НДДКР розраховано на основі наведених даних прикладу (Таблиця 4.3).

Таблиця 4.3

Експертна оцінка і розрахунок величини інтегрального показника НТЕ

№	Групи показників	Рейтинг експертів			Середня за експертними оцінками	НТЕ
		1	2	3		
1	Науково-технічний рівень	8	7	9	8	2,4 (8 x 0,30)
2	Перспективність	8	6	8	7,33	1,83 (7,33 x 0,25)
3	Потенційний масштаб практичного використання	6	7	7	6,66	1,33 (6,66 x 0,20)
4	Ступінь вірогідності досягнення позитивних результатів	9	8	8	8,33	2,08 (8,33 x 0,25)
В С Ъ О Г О						7,64

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дата		

$$НТЕ = 8 \cdot 0,30 + 7,33 \cdot 0,25 + 6,66 \cdot 0,20 + 8,33 \cdot 0,25 = 2,4 + 1,83 + 1,33 + 2,08 = 7,64$$

Отриманий результат слід порівняти з максимально можливим значенням, яке дорівнює 10 балам ($10 \cdot 0,30 + 10 \cdot 0,25 + 10 \cdot 0,20 + 10 \cdot 0,25$).

Отже, оцінка рівня НТЕ може бути зроблена за допомогою інтегрального коефіцієнта оцінки НТЕ ($K_{НТЕ}$):

$$K_{НТЕ} = \frac{НТЕ}{10} \cdot 100 \% \quad (4.3)$$

На основі даних Таблиці 4.3, можна дійти до висновку, що $K_{НТЕ}$ відповідає 76,4 %, тобто:

$$\frac{7,64}{10} \times 100 = 76,4$$

В тому випадку, коли значення $K_{НТЕ}$ перевищує середнє значення, яке дорівнює 5,0, має бути зроблено висновок про достатній рівень НТЕ:

- цілком достатній 5,0 – 6,0;
- достатній 6,1 – 8,0;
- достатньо високий 8,1 – 9,0;
- високий 9,1 – 10.

Таким чином, рівень НТЕ технології можна визнати достатнім. Отже, розроблену технологію пропонується впроваджувати у виробництво.

4.4 Розрахунок економічної ефективності проєкту

Розрахунок економічної ефективності проєкту - це процес визначення потенційного доходу та витрат проєкту, а також оцінювання того, наскільки доцільно вкладати гроші у реалізацію проєкту.

Для розрахунку економічної ефективності проєкту необхідно визначити наступні показники:

1. Вартість проєкту: це загальна вартість всіх витрат, пов'язаних з реалізацією проєкту, включаючи затрати на розробку, впровадження та експлуатацію.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						86
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Прибуток від проєкту: це дохід, отриманий від реалізації продукту або послуги, яку реалізовує проєкт.
3. Термін окупності: це період, за який витрати на реалізацію проєкту повернуться від прибутку.
4. Рентабельність інвестицій (PI): це відношення чистого дисконтованого доходу до вартості інвестицій.

Для розрахунку економічної ефективності проєкту, потрібно мати інформацію про витрати та прибуток проєкту.

Припустимо, що витрати на проєкт складають 100 000 грн. Ця сума складається з оплати студента як розробника, керівника проєкту та консультанта за 3 місяці, податків на зарплату – єдиний соціальний внесок – 22 % від зарплати, оплати за програмне забезпечення, амортизації оргтехніки, канцтоварів, оплати Інтернет та зв'язку, оплати електроенергії, води, опалення.

А очікуваний прибуток становить 200 000 грн. Тоді ціна проєкту буде розрахована за формулою:

$$\text{Ціна} = \text{Витрати} + \text{Прибуток (30\%-300\%)} \quad (4.4)$$

Проведемо розрахунок:

$$\text{Ціна} = 100\,000 + 200\,000 \cdot 30\%$$

Ціна буде дорівнювати = 160 000 грн

Рентабельність проєкту можна розрахувати за формулою:

$$R = \text{Прибуток} / \text{Витрати} * 100\% \quad (4.5)$$

Підставляю свої значення:

$$R = \frac{200\,000}{100\,000} \times 100$$

Рентабельність проєкту дорівнює $R = 200\%$

Срок окупності проєкту можна розрахувати за формулою:

$$T = \frac{1}{R} \quad (4.6)$$

Підставляю свої значення:

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дата		

$$T = \frac{1}{200\%} = 0,5$$

Час, за який проєкт окупиться дорівнює $T = 0,5$ року або 6 місяців.

Висновок: Представлений аналіз дозволяє зробити висновок, що впровадження даного продукту є економічно вигідним, тому розробка цього програмного продукту є актуальною. Проєкт з витратами 100 000 грн та очікуваним прибутком 200 000 грн при реалізації за ціною 160 000 грн матиме рентабельність 200% і буде окупатися за 6 місяців.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						88
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 5

ОХОРОНА ПРАЦІ

5.1 Шкідливі та небезпечні фактори в роботі користувача ПК

Життєдіяльність безпосередньо пов'язана з виконанням певного виду робіт та продуктивністю праці, яка визначається як людським чинником, так і засобами виробництва, а також технологічними й організаційними умовами праці. Сьогодні діяльність більшості працівників сучасних професій у виробничій сфері пов'язана з використанням комп'ютерної техніки. Працюючи за комп'ютером, людина потрапляє під вплив різноманітних факторів: електромагнітних полів, інфрачервоного та іонізуючого випромінювання, шуму і вібрацій, статичної електрики.

Робота за комп'ютером вимагає значної розумової напруги і супроводжується нервово-емоційним навантаженням операторів, високою напругою зорової роботи і досить значним навантаженням на м'язи рук під час роботи з клавіатурою ПК. Велике значення має раціональна конструкція і розташування елементів робочого місця, що є важливим для підтримки оптимальної робочої пози в процесі праці.

У процесі роботи за комп'ютером необхідно дотримуватись правильного режиму роботи і відпочинку. В протилежному випадку в персоналу спостерігається незадоволеність роботою, головний біль, роздратування, порушення сну, втома і больові відчуття в очах, попереку, у ділянці шиї та рук.

Обчислювальна техніка є джерелом тепловиділення, що може спричинити підвищення температури і зниження відносної вологості у приміщенні.

Рівень шуму на робочому місці адміністратора та касирів не має перевищувати встановлених норм. Для зниження рівня шуму стіни і стеля

					КРБ.КІ.1.147-03.3.5	Арк.
						89
Змн.	Арк.	№ докум.	Підпис	Дата		

приміщення, де встановлені комп'ютери, мають бути облицьовані звукопоглинаючими матеріалами.

Оптичне випромінювання включає: ультрафіолетове (УФ), світлове та інфрачервоне.

УФ-випромінювання впливає, як правило, на шкіру та очі людини.

Світлове випромінювання впливає, в основному, на очі і провокує їх втому та запалення райдужної оболонки. Однак ці симптоми швидко минають і не викликають патологічних змін.

Електромагнітні випромінювання (ЕМВ) радіочастотного діапазону. Джерелом ЕМВ є монітор. Тому, обираючи робоче місце для комп'ютера, необхідно пам'ятати, що його задня і бокові стінки можуть бути джерелом значно більшого ЕМВ, ніж екран.

Для зменшення впливу перерахованих видів випромінювання рекомендується застосовувати монітори зі зниженою випромінювальною здатністю, а також дотримуватись регламентованого режиму праці та відпочинку.

5.2 Методи зниження впливу шкідливих та небезпечних факторів при роботі за ПК

Заходи щодо усунення небезпеки ураження електричним струмом зводяться до правильного розміщення устаткування та електричних кабелів. Інші заходи щодо забезпечення електробезпеки, збігаються з загальними заходами пожежо- та електробезпеки.

Екран дисплея повинен бути розташованим перпендикулярно до напрямку погляду. Якщо він розташований під кутом, то стає причиною сутулості. Відстань від дисплея до очей повинна трохи перевищувати звичну відстань між книгою та очима. Перед екраном монітора, особливо старих типів, повинен бути спеціальний захисний екран. При його відсутності треба

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

сидіти на відстані витягнутої руки від монітора. Ще одним моментом, який стосується зору, є необхідність створення неоднорідного поля зору. Для цього можна розвісити на поверхнях (стінах) плакати та картини, виконані у спокійних тонах. Наприклад, пейзажі.

Важливою є форма спинки крісла, яка повинна повторювати форму спини. Висота крісла повинна бути такою, щоб користувач не почував тиску на куприк або стегна. Крісло бажано обладнати бильцями. Його потрібно встановити так, щоб не треба було тягтися до клавіатури. Періодично користувачу необхідно рухатися, вчасно змінювати положення тіла і робити перерви у роботі.

При напруженій роботі за комп'ютером щогодини необхідно робити перерву на 15 хвилин через кожну годину і треба займатися іншою справою. Декілька разів на годину бажано виконувати серію легких вправ для розслаблення.

Наслідками регулярної роботи за комп'ютером без застосування захисних засобів можуть бути: захворювання органів зору, хвороби серцево-судинної системи, захворювання шлунково-кишкового тракту, шкірні захворювання, різноманітні пухлини.

Якщо у приміщенні експлуатується більше одного комп'ютера, то треба врахувати, що на користувача одного комп'ютера можуть впливати випромінювання від інших, в першу чергу бокових, а також і задньої стінки сусіднього дисплея. Тому необхідний захист спеціальними фільтрами і щоб користувач розміщався від бічних і задніх стінок інших дисплеїв на відстані не ближче одного метра.

Отже, щоб запобігти негативним впливам необхідно знати й небезпечні сторони самого комп'ютера і правила безпечної роботи, знати засоби запобігання небезпек. Вони пов'язані перед усім із загально відомими небезпечними факторами - поразками електричним струмом, пожежонебезпечністю.

5.3 Техніка безпеки при роботі за ПК (та при роботі з програмним продуктом)

Перед початком роботи на ПК користувач повинен:

- пересвідчитися у цілості корпусів і блоків (обладнання) ПК;

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дата		

- перевірити наявність заземлення, справність і цілість кабелів живлення, місця їх підключення.

Забороняється вмикати ПК та починати роботу при виявлених несправностях.

Під час роботи, пересвідчившись у справності обладнання, увімкнути електроживлення ПК, розпочати роботу, дотримуючись умов інструкції з її експлуатації.

Забороняється:

- замінювати різноманітні деталі або вузли та проводити ремонт при ввімкненому ПК;
- з'єднувати і роз'єднувати вилки та розетки первинних мереж електроживлення, які знаходяться під напругою;
- знімати кришки, які закривають доступ до струмопровідних частин мережі первинного електроживлення при ввімкненому обладнанні;
- користуватися паяльником з незаземленим корпусом;
- замінювати запобіжники під напругою;
- залишати ПК у ввімкненому стані без нагляду.

По закінченні робочого дня:

- кнопкою "ВИМК" відключити електроживлення ПК згідно з інструкцією експлуатації, вийнявши вилку кабелю живлення з розетки;
- впорядкувати робоче місце користувача ПК, прибравши використане обладнання та матеріали у відведені місця;
- про виявлені недоліків у роботі ПК протягом робочого часу необхідно повідомити відповідним посадовим особам та спеціалістам.

5.4 Вимоги безпеки після закінчення роботи з комп'ютером, принтером, ксероксом, сканером та іншою оргтехнікою

- Вимкнути комп'ютер, ноутбук, телевізор, плазмову панель, LCD-екран,

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дата		

принтер, ксерокс, сканер, колонки та іншу оргтехніку від електромережі, для чого необхідно вимкнути тумблери, а потім акуратно витягнути штепсельні вилки з розетки.

- Протерти зовнішню поверхню комп'ютера чистою вологою тканиною. При цьому не допускайте використання розчинників, одеколону, препаратів в аерозольній упаковці.
- Прибрати робоче місце. Скласти диски у відповідне місце зберігання.
- Ретельно провітрити приміщення з персональним комп'ютером та іншою оргтехнікою.

5.5 Вимоги техніки безпеки та безпеки життєдіяльності в аварійних ситуаціях при роботі з комп'ютером та іншою оргтехнікою

- Якщо на металевих частинах обладнання виявлено напругу (відчуття струму), заземлюючий провід обірваний, необхідно вимкнути обладнання, негайно доповісти керівникові про несправності електрообладнання і без його вказівки до роботи не приступати.
- При припиненні подавання електроенергії, вимкнути обладнання.
- При появі незвичного звуку, запаху паленого, мимовільного відключення комп'ютера та оргтехніки, негайно припинити роботу і поставити до відома керівника.

Висновки: Темпи зростання числа користувачів ПК неухильно зростають. Одночасно з цим з'являється можлива небезпека для здоров'я працюючих за ПК. Під час роботи за комп'ютером найбільшому ризику піддаються зорова, опорно-рухова, нервово-психічна системи і репродуктивна функція у жінок. Крім того, відеодисплейний термінал порушує рівновагу між позитивно і негативно зарядженими іонами в повітрі. А це означає що потрібно намагатися відпочивати від роботи за комп'ютером та проходити мед. огляд при перших симптомах прояву вище перерахованих проблем. При дотриманні всіх вимог і правил техніки безпеки користувач збереже своє життя і здоров'я.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						93
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗАГАЛЬНИЙ ВИСНОВОК

Робота складається з п'яти розділів. Перший розділ містить загальні відомості про комп'ютерні ігри та їх жанрову класифікацію. В ході виконання проектного аналізу мною було розглянуто аналоги ігрових продуктів різних компаній. Такий аналіз допомагає виявити переваги і недоліки ігор-аналогів і оцінити конкурентну перевагу гри, що розробляється, а також поглибити знання про створення оригінального концепту гри та виділити найбільш істотні переваги та недоліки цих аналогів. Базуючись на отриманій інформації, була поставлена наступна мета роботи – створення комп'ютерної гри для платформи *PC* у вибраному мною жанрі.

Другий розділ містить у собі концепт документу проекту, а також спрайти героя, його станів, анімації. Крім цього було описано вороги головного героя, перешкоди та спрайти основної сцени.

Третій розділ – етап проектування, була надана увага щодо структурних, поведінкових та інтерфейсних аспектів гри. Вони дають змогу перейти до практичної реалізації гри з використанням відповідних засобів в обраному мною середовищі розробки. Взагалі, при використанні спіральної моделі проектування, можливий перегляд цих рішень з наступним внесенням змін до реалізації.

Не менш важливими є і засоби реалізації подібних продуктів. Мною було обрано середовище розробки під назвою *Unity* оскільки я вже мав опит його використання. В *Unity* було побудовано рівні гри, а також усі елементи та персонажі. Грунтуючись на сучасних вимогах до ігор, та технологій було створено програмне забезпечення, а саме гру-платформер. Протестовано основні дії користувача в грі, не було виявлено жодних відхилень.

Були проведені економічні розрахунки з метою довести економічну доцільність проекту. Виконані розрахунки показали, що впровадження даного

					КРБ.КІ.1.147-03.3.5	Арк.
						94
Змн.	Арк.	№ докум.	Підпис	Дата		

програмного продукту є економічно доцільним.

Дуже важливо під час користування комп'ютерною технікою дотримуватися правил пожежної та електробезпеки. Цим правилам було надано особливої уваги у п'ятому розділі.

Завершаючим результатом проектної роботи є робочий прототип гри у жанрі Платформер, який не має високих технічних вимог, має просте меню, управління, інтерфейс, мету гри, а також умови для її завершення. Даний прототип гри може бути доповнений новими деталями і елементами та офіційно випущений як гра на базі якогось інтернет-ресурсу.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						95
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Комп'ютерна гра [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/PC_game.
2. Жанр Платформер [Електронний ресурс] – Режим доступу до ресурсу: en.wikipedia.org/wiki/Platform_game.
3. Класифікація ігор за жанрами [Електронний ресурс] – Режим доступу до ресурсу: <https://gamesisart.com/>.
4. Донован Т. Играй! История видеоигр / Тристан Донован; пер. И. Воронина. — М.: Белое Яблоко, 2014
5. Wolf, Mark J. P. The Video Game Explosion: A History from PONG to Playstation and Beyond / Mark J. P. Wolf. ABC-CLIO, 2008
6. Офіційний вебсайт серії Sonic the Hedgehog Архівовано 13 січня 2016 у Wayback Machine.
7. Donkey Kong Архівовано 21 травня 2019 у Wayback Machine. на MobyGames
8. Як штучний інтелект може допомогти освіті [Електронний ресурс]. – Режим доступу: <https://osvitoria.media/experience/yak-shtuchnyj-intelekt-mozhe-dopomogty-osviti/> (дата звернення 18.03.2023).
9. Найкращі ігрові рушії [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gamedesigning.org/career/video-game-engines>.
10. Unreal Engine [Електронний ресурс] – Режим доступу до ресурсу: <https://www.unrealengine.com/>.
11. Cry Engine [Електронний ресурс] – Режим доступу до ресурсу: <https://www.cryengine.com/>.
12. Unity Game Engine [Електронний ресурс] – Режим доступу до ресурсу: <https://unity.com/ru>.
13. Unity Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.unity3d.com/Manual/index.html>.

					КРБ.КІ.1.147-03.3.5	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дата		

14. David B. Hands-On Game Development Patterns with Unity / Baron David., 2019. – 116 с.
15. Створення 2D на Unity [Електронний ресурс] – Режим доступу до ресурсу:
https://skillbox.ru/media/code/kak_sozdat_prostuyu_2d_igru_na_unity/.
16. 2D на Unity, докладний посібник [Електронний ресурс] – Режим доступу до ресурсу: <http://websketches.ru/blog/2d-igra-na-unity-podrobnoye-rukovodstvo-p1>.
17. Пояснення про спрайти, тайли [Електронний ресурс] – Режим доступу до ресурсу: <https://www.econdude.pw/2017/08/chto-takoe-graficheskij-sprajt-sprite.html>.
18. Hocking J. Unity in Action. Multiplatform game development in C# with Unity 5 / Joseph Hocking., 2015. – 352 с.
19. Smith G. Basic Math for Game Development with Unity 3D / G. Smith, K. Sung., 2018. – 279 с.
20. Thorn A. Mastering Unity Scripting / Alan Thorn., 2015. – 380 с.
21. А. Хейлсберг, М. Торгерсен, С. Вилтамут, П. Голд , Язык программирования C#. Классика Computers Science. 4-е изд.
22. Бьёрн Страуструп, Программирование: принципы и практика использования. 2-е изд.
23. Э. Стилмен, Дж. Грин, Изучаем C#. 3-е изд.
24. Герберт Шилдт, C# 4.0. Полное руководство.
25. Ребекка М. Райордан «Основы реляционных баз данных».: Ред.: Русская Редакция, 2001.- 384 с.
26. Бойчик І.М., Харів П.С., Хопчан М.І., Економіка підприємства: Ред.: Сполум – 1998 – 212 с.
27. Закон України «Про охорону праці» від 14.10.1992 № 2694-ХІІ
28. ДНАОП 0.00-1.31-99 «Правила охорони праці під час експлуатації

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						97
Змн.	Арк.	№ докум.	Підпис	Дата		

електронно-обчислювальної техніки»

- 29.ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. – Чинний з 10.12.1998. – К.: Держспоживстандарт України, 1998. – 25 с. <http://pidruchniki.com>
- 30.Жидецький В. Ц. Охорона праці користувачів комп'ютерів. - Львів: Афіша, 2000. - 176 с.

					<i>КРБ.КІ.1.147-03.3.5</i>	Арк.
						98
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		