

Галицький коледж імені В'ячеслава Чорновола
відділення комп'ютерних та видавничих технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділенням
комп'ютерних та видавничих
технологій

Чубей О.О. / _____ /

« ____ » _____ ^{підпис} 2020 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту
освітньо–кваліфікаційного рівня «молодший спеціаліст»
зі спеціальності 122 «Комп'ютерні науки та інформаційні технології»
на тему: «Логічно – розвиваюча гра»

Студент групи К–47

Пісний О.В.

(підпис)

Керівник проєкту

Кульчинська Н.З.

(підпис)

Консультанти:

з техніко – економічного
обґрунтування

Меленчук Л.І.

(підпис)

нормо контролер

Гавришків Н.Г.

(підпис)

Тернопіль – 2020

Галицький коледж імені В'ячеслава Чорновола
відділення комп'ютерних та видавничих технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ
Завідувач відділенням
комп'ютерних та видавничих
технологій

Чубей О.О. / _____ /

підпис

«___» _____ 2019 р.

ЗАВДАННЯ

на дипломне проєктування
на здобуття освітньо – кваліфікаційного рівня «молодший спеціаліст»
студенту Пісному Олегу Володимировичу
(прізвище, ім'я та по – батькові студента)

1. Тема проєкту «Логічно – розвиваюча гра»

затверджена наказом по коледжу від 20.11.2019 р.

2. Термін здачі студентом завершеного проєкту «___» _____ 2020р.

3. Вихідні дані до проєкту – результати аналізу програмної області ігрових програм у жанрі головоломки, результати аналізу попиту на ринку комп'ютерних ігор

4. Перелік питань, які повинні бути розроблені в проєкті:

а) основна частина – опис об'єкту інформатизації, проєктування інформаційної системи, реалізація ігрового продукту, тестування.

б) Техніко–економічне обґрунтування – аналіз ринку, розрахунок витрат на проєктування, обґрунтування необхідності розробки.

5. Перелік графічного матеріалу – діаграма варіантів використання, схема популярності ігрових жанрів, рейтинг ігрових рушіїв.

6. Консультанти проекту: _____

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко— економічного обґрунтування	_____ (вчена ступень, звання П.І.Б. консультанта)		

КАЛЕНДАРНИЙ ПЛАН дипломного проектування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до дипломного проектування	11.11.19	18.11.19
2.	Огляд типових рішень	19.11.19	25.11.19
3.	Дослідження засобів та технологій реалізації проекту	26.11.19	15.12.19
4.	Розробка функціональних вимог до системи та робота над її структурою	16.12.19	27.01.20
5.	Встановлення на налаштування програмного забезпечення	28.01.20	11.02.20
6.	Проектування структури програмного продукту	12.02.20	15.03.20
7.	Розробка програмного продукту.	16.03.20	27.04.20
8.	Доопрацювання модулів, підсистем і т.д.	28.04.20	06.05.20
9.	Опрацювання питань з техніко — економічного обґрунтування	07.05.20	20.05.20
10.	Тестування на налагодження програмного продукту	21.05.20	25.05.20
11.	Робота над основним розділом пояснювальної записки	26.05.06.20	10.06.20
12.	Оформлення пояснювальної записки, додатків	11.06.20	17.06.20
13.	Попередній захист дипломного проекту, доопрацювання над проектом	18.06.20	
14.	Підготовка до захисту дипломного проекту	19.06.20	25.06.20
15.	Захист дипломного проекту	26.06.20	

7. Дата видачі завдання “___” _____ 2019 р.

Керівник _____ / _____ /

Завдання прийняв до виконання _____ / _____ /

Реферат

Логічно – розвиваюча гра. Дипломний проєкт. Пісний Олег Володимирович. Сторінок – 58, рисунків – 38, джерел – 9, додатків – 2.

Об’єкт дослідження – ігрові продукти спроектовані і розроблені в жанрі головоломка та засоби їх реалізації.

Метою дипломного проєкту є розробка ігрового продукту в жанрі головоломка та елементами жанру Adventure на базі ігрового рушія Unity.

Завданням проєкту є розробка ігрового продукту в жанрі головоломка.

Для розробки системи було використано ігровий рушій Unity та програмне забезпечення MagicaVoxel, Visual Studio та Blender.

Для написання скриптів та тригерів які їх використовуються в ігровому продукті використано мову програмування C#.

Результатом виконання проєкту є спроектований та розроблений ігровий продукт на базі ігрового рушія Unity. Гра має простий та зручний користувацький інтерфейс.

СИСТЕМА, ГОЛОВОЛОМКА, ADVENTURE, UNITY, C#, РЕАЛІЗАЦІЯ ПРОЄКТУ, TRIGGER, PLAY.

Abstract

Logic and developmental game. Diploma project. Pisnyi Oleh Volodymyrovych. Pages – 58, figures – 38, sources – 9, appendices – 2.

The object of research – game products designed and developed in the genre of puzzles and means of their implementation.

The aim of the diploma project is to develop a game product in the genre of puzzle and elements of the genre of Adventure based on the game engine Unity.

The task of the project is to develop a game product in the genre of puzzle.

Unity game engine and MagicaVoxel, Visual Studio and Blender software were used to develop the system.

The C # programming language is used to write the scripts and triggers that are used in the game product.

The result of the project is a designed and developed game product based on the Unity game engine. The game has a simple and user – friendly interface.

СИСТЕМА, ГОЛОВОЛОМКА, ADVENTURE, UNITY, C#, РЕАЛІЗАЦІЯ ПРОЄКТУ, TRIGGER, PLAY.

ЗМІСТ

Вступ.....	8
1 Опис об'єкту інформатизації.....	9
1.1 Характеристика об'єкту інформатизації.....	9
1.2 Аналіз існуючих рішень	10
1.3 Дослідження засобів реалізації.....	13
1.4 Обґрунтування доцільності створення ігрового продукту	16
1.5 Постановка задачі.	18
2 Проєктування інформаційної системи	19
2.1 Опис предметної області	19
2.2 Розробка сюжету гри	20
2.3 Обґрунтування технологій та засобів реалізації.....	22
3 Реалізація ігрового продукту	26
3.1 Реалізація інтерфейсу ігрового продукту	26
3.2 Реалізація рівнів гри	29
3.3 Реалізація рухів персонажів	37
4 Тестування	39
5 Техніко – економічне обґрунтування.....	45
5.1 Аналіз ринку	45
5.2 Розрахунок витрат на проєктування	46
5.3 Обґрунтування необхідності розробки	48
Висновки	50
Перелік джерел посилання	52
Додатки.....	53

					ДП.КН.20.416.20.000 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Пісний О.В.			Логічно–розвиваюча гра		Літ.	Арк.
Перевір.		Кульчинська Н.З.						Аркуші
Реценз..		Посвятовська О.Б.					6	58
Н.контр.		Гавришків Н.Г.					ГК. ВКВТ К–47	
Зав.відділ		Чубей О.О.						

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІТ – Інформаційні технології

ЛКМ – Ліва кнопка миші

ПЗ – Програмне забезпечення

ПП – Програмний продукт

AS – Asset Store

CE – CryEngine

FPS – First Person Shooter

GM – GameMode

GML – Game Maker Language

MMO – Massively Multiplayer Online

OS – Operating System

PC – Personal computer

VR – Virtual Reality

VS – Visual Studio

					ДП.КН.20.416.20.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Комп'ютерні відеоігри набули значної популярності за останні десятиліття та посіли почесне місце на ринку програмного забезпечення для розваг та дозвілля. В результаті відбувся бурхливий розвиток індустрії комп'ютерних ігор та інструментальних засобів для їхньої розробки.

Для задоволення потреб розробників було створено величезну кількість інструментальних засобів та продовжується розробка нових. Загальна ціль даних середовищ для створення ігор – полегшення та покращення розробки, використання передових технологій обробки графіки, фізики, забезпечення кросплатформності розроблених проєктів.

Завдяки розвитку ринку інді-ігор, багато творців комп'ютерних ігор отримали можливість працювати над своїми ігровими проєктами без правових та економічних зобов'язань перед ІТ-фірмами – розробниками відповідного програмного забезпечення. Інді-ігри – це комп'ютерні ігри, створені окремими розробниками або маленькими колективами без фінансової допомоги із сторони великих ІТ-компаній. Просування таких ігор відбувається з використанням каналів цифрового постачання.

Головоломка – найменування жанру відеоігор, завданням яких є вирішення логічних задач, вимагаючи від гравця використання логіки, стратегії, інтуїції і спостережливості. Елементи головоломки можуть вводитися в ігри інших жанрів як першорядні компоненти ігрового процесу або для його різноманіття за допомогою міні-гри. Аналогічно, ігри-головоломки можуть містити різні компоненти пригодницького жанру.

Завданням даного дипломного проєкту є реалізація комп'ютерної гри, яка зможе задовільнити потреби користувача в ігровому сюжеті в жанрі головоломки, а також – якісній графічній складовій, розважальному контенті.

Об'єкт дослідження – ігрові продукти спроектовані і розроблені в жанрі головоломка та засоби для проєктування та розробки комп'ютерних ігор.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ОПИС ОБ'ЄКТУ ІНФОРМАТИЗАЦІЇ

1.1 Характеристика об'єкту інформатизації

Засновниками жанру головоломки були настільні, текстові і механічні головоломки – від кубика Рубіка до ребусів які були модні в той час. Перша гра з жанру головоломок у відеоіграх – «Amazing Maze», яка була підготовлена для аркадних автоматів. Основною метою гри було покинути лабіринт швидше ніж це зробить штучний розум комп'ютера. У подальші роки головоломки проєктувалися для розвитку інтелекту дітей.

У 80-х роках вийшла гра "Loco-Motion", де користувачеві потрібно було повертати квадратики з фрагментами залізниці, щоб скласти цілу дорогу і провести по ній потяг. Величезну популярність також здобула гра, де застосовувалася ізометрична графіка «Q * bert» яка була розроблена в 82 році. У ній потрібно було зафарбувати куби, плигаючи по ним ігровим персонажем, при цьому дивлячись щоб вони перефарбувалися на потрібний колір з декількох ймовірних.

Гра «Boulder Dash», розроблена в 84 році об'єднувала кілька цілей, з яких: сформувати для себе комфортний маршрут, а не застрягти надовго у рівні, назбирати певну кількість об'єктів і обходить небезпечні пастки.

Зразком жанру стала модна на той час гра «Тетріс», яка була розроблена в 1984 році і об'єднувала в собі простоту і цікавий ігровий процес.

Від початку 1990-х років комп'ютерні ігри в жанрі головоломок почали розвиватись ще активніше. Цьому також сприяла поява тривимірної графіки на ігрових приставках.

Протягом 2000-х років були створені такі популярні ігри-головоломки, як Pikmin, Polarium, LocoRoco та інші. Вони знову відродили принципи жанру.

Крім цього, ігрові продукти даного жанру, а саму – розвиваючі логічні почали активно використовуватись для розвитку логічного мислення у дітей.

В 2015 році величезний прорив в жанрі таких відеоігор стала гра "Undertale" від творця Тобі Фокс, де гравець маніпулює героєм – дитиною,

					ДП.КН.20.416.20.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

мандрує підземеллям, в якому проживає велика кількість всіляких ворогів, з якими повинен контактувати гравець, щоб вибратися. Вибір гравця надзвичайно сильно впливає на підсумковий результат відеогри, у залежності від рішень змінюються розмови і заключні сцени.

1.2 Аналіз існуючих рішень

Комп'ютерні ігри стають все більш складними та функціональними. На даний момент часу багато компаній займається розробкою відео–ігор та ігрових рушіїв, відповідно, кожна група розробників чи окремий розробник може обирати жанр та платформу для своєї гри.

Ігрова індустрія розвивається завдяки користувачам, оскільки вони платять компаніям за розваги (ігри). Для того, щоб отримати більше прибутку від свого продукту, компаніям потрібно збільшувати зацікавленість користувачів до гри, тому ігрові компанії постійно конкурують між собою, таким чином, створюється більш цікавий ігровий сюжет, більш продуманий ігровий світ та покращення графіки у проєкті.

Проведемо огляд подібних ігрових продуктів в жанрі головоломки, визначимо їх переваги та недоліки.

Отже, Undertale – комп'ютерна гра виконана в жанрах: RPG, головоломка. У Undertale гравець керує дитиною, яка звалилася в «Підземелля» через тріщину в горі «Еботт». Підземелля населене великою кількістю ворогів, з якими гравець мусить побудувати хороші, або погані відносини, щоб вибратися. В залежності від вибору гравця, сюжет може дуже сильно змінюватися, кожне прийняте рішення може вплинути на кінцевий результат гри, діалоги та фінальні сцени.

Головний персонаж гри, який досліджує світ підземелля, населений ворогами, має змогу як знищити їх, так і подружитись. Гравець бачить світ видом зверху, основний його рівень називається «Зовнішнім підземеллям», в якому є відділені місця, як магазини чи будинки з ігровими персонажами в них. Гравець має вирішувати різні головоломки, а також взаємодіяти з різними неігровими персонажами для просування сюжетом.

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

У зовнішньому підземеллі зустрічаються точки збереження, в яких зберігається прогрес гри і відновлюється здоров'я гравця. Існує єдиний слот збереження, тобто нове збереження перезаписує старе і змінити попередній вибір в сюжеті у рамках того самого проходження неможливо (рисунок 1.1).

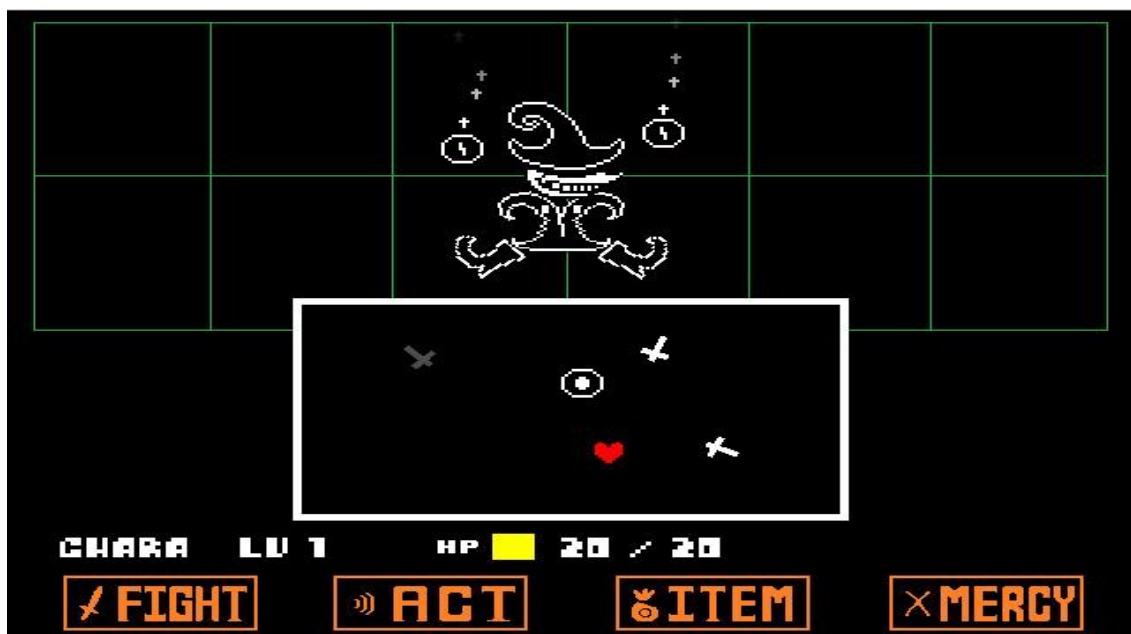


Рисунок 1.1 – Знімок екрана з гри (момент бою з чудовиськом)

Перехід між локаціями в цій грі реалізований за допомогою «Cut Scene» тобто користувач доходить до точки в якій спрацьовує функція, яка запускає відео-фрагмент (рисунок 1.2).



Рисунок 1.2 – Знімок екрана з гри (рівень з баром)

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Наступним прикладом обрано відеогру The Witness. В грі The Witness дія відбувається від першої особи, гравець повинен досліджувати дивний острів, де є багато як природних так і штучних споруд. Головним завданням гравця є вирішення головоломок–лабіринтів, як доволі простих представлених на звичайних дошках, так і значно складніших які потребують взаємодії з навколишнім середовищем (рисунок 1.3).



Рисунок 1.3 – Знімок екрана з гри (панель головоломки)

Також на кожній новій місцевості гравцю доведеться вирішувати нові види головоломок і дуже часто з першого погляду проста головоломка може виявитися значно складнішою ніж здавалася, хоча можливо і зворотнє, коли складна і незрозуміла головоломка насправді має просте та банальне вирішення.

Локації гри також відрізняються своїми стилями, представляючи минуле, теперішнє та майбутнє. Таким чином на кожній новій локації, антураж гри буде різний, а отже гравця чекає різноманітність та чергування локацій, від технологічних та футуристичних, до сучасних та середньовічних.

Окрім того, для більшості головоломок є підказки, які можуть допомогти тим гравцям хто зможе їх знайти та зрозуміти (рисунок 1.4).

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12



Рисунок 1.4 – Знімок екрана з гри (рівень «замок»)

Загалом, досліджені ігри є яскравими представниками свого жанру, але Undertale є менш цікавою в плані комп'ютерної графіки та цікавих локацій, що, очевидно приваблює користувача. Але, все ж таки, основним завданням комп'ютерних ігор в жанрі головоломки є забезпечення гравця цікавими логічними завданнями, які вимагають уваги, мислення та зосередженості в процесі гри.

1.3 Дослідження засобів реалізації

Для створення відео ігор використовуються різні середовища розробки, вони можуть бути як платними так і розповсюджуватися у вільному доступі. Платні середовища переважно надають більшу гнучкість та функціонал яких немає у безкоштовних версіях програм. [3]

При підготовці до створення свого програмного продукту було проаналізовано декілька середовищ розробки крім Unity 3D, такі як:

- Game Maker;
- 3D Rad;
- CryEngine.

Першим на розглядання середовищем був Game Maker, цей рушій доволі легкий у використанні та зрозумілий новачкам тому що це фактично конструктор

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

відео гри, розробник завантажує у середовище переважно 2D спрайти або повноцінні 3D моделі. Створення гри відбувається завдяки візуальному програмуванню але можна обійтися без нього якщо вивчити GML тобто вбудовану мову програмування. Завдяки GML ігри які будуть створюватися стануть більш якісними та частково цікавішими. Інтерфейс цієї програми можна побачити на рисунку 1.5.

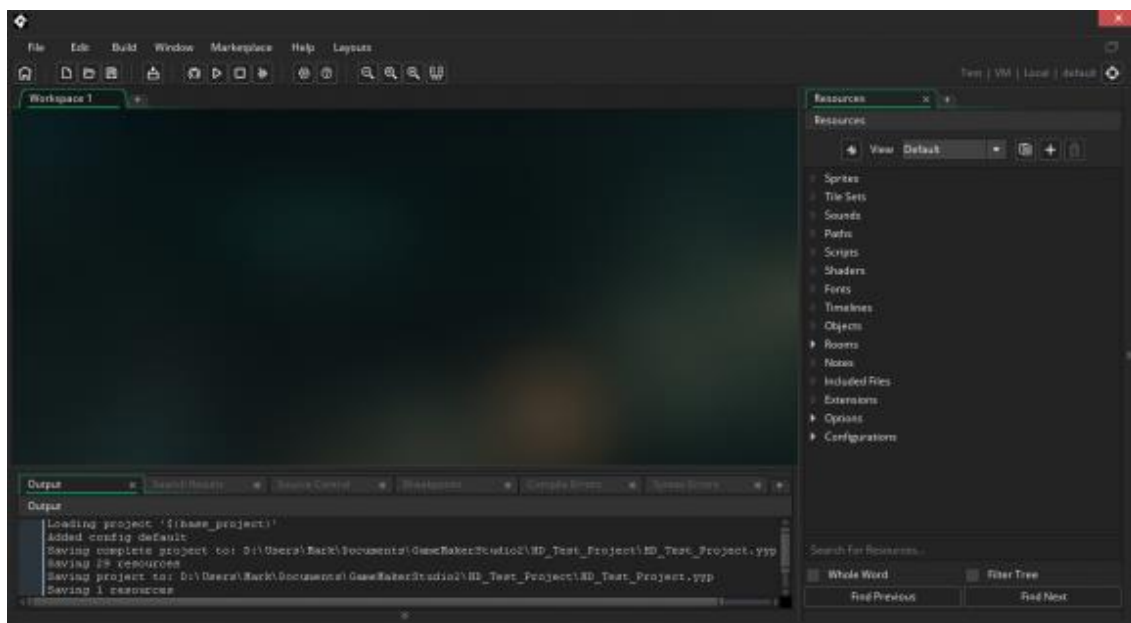


Рисунок 1.5 – Інтерфейс програми Game Maker

Game Maker невибагливий до системних та апаратних характеристик комп'ютера і має можливість розробки гри під різні операційні системи.

Наступним було розглянуто рушій 3D Rad

3D Rad – простий рушій який бу розроблений у країнах СНГ програма призначена для вибудовування 3D–ігор, додатків різного ступеня трудомісткості та фізичних емуляцій. Йдеться про процес без використання коду. Інтерфейс програми інтуїтивно зрозумілий тому труднощів з використанням інструментів у новачка не повинно бути.

У цьому рушії є доволі хороші переваги в плані фізики, так вона доволі реалістична і дає можливість створювати хороші авто симулятори. В програмі є вбудовані різні ефекти:

- відблиск;
- вода;

					ДП.КН.20.416.20.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

- туман;
- вогонь;
- плями світла.

Інтерфейс цього рушія зображено на рисунку 1.6.

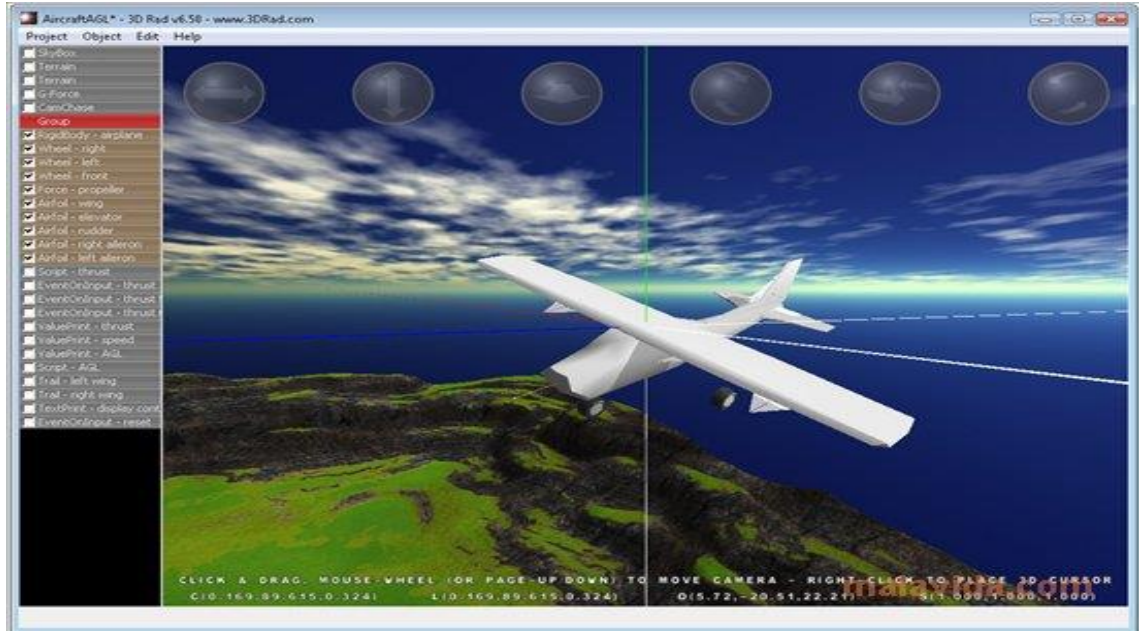


Рисунок 1.6 – Інтерфейс рушія 3D Rad

Останнім був проведений рушій CryEngine. Цей рушій зарекомендував себе надзвичайно добре у 2004 році, саме тоді вийшла відома гра “Far Cry” від студії Crytek. CryEngine потужний рушій який може показувати високі результати у графічному плані.

У цьому середовищі дуже велика кількість інструментів для створення своєї відеогри, завдяки ним можна швидко створити ескіз ігрової моделі у редакторі або навіть на самій сцені.

Хоча ігри які створюють на цьому рушії і виглядають прекрасно але за цю красу потрібно платити, тка є і безкоштовна версія цього двигуна але вона значно урізана і призначена тільки для ознайомлення. Інтерфейс рушія зображено на рисунку 1.7.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

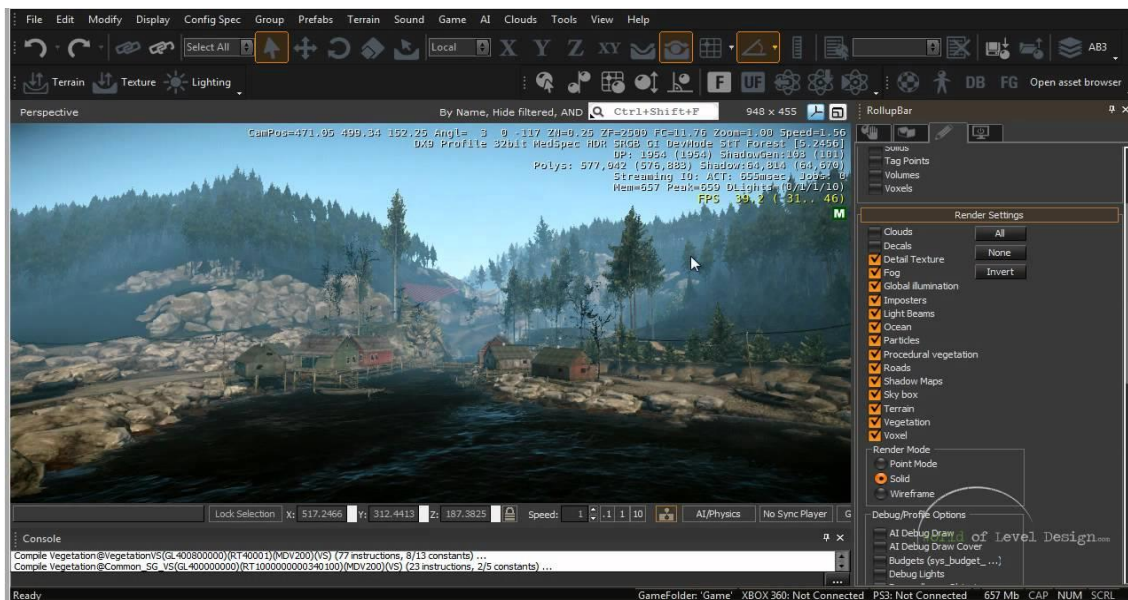


Рисунок 1.7 – Інтерфейс рушія CryEngine

1.4 Обґрунтування доцільності створення ігрового продукту

Головоломки як жанр відеоігор було створено у кінці 70-х і до сьогоднішнього часу вони не втрачають свої популярності, актуальності та користуються величезним попитом на ринку ігрових продуктів.

Наприклад, у 1984 році була створена дуже популярна гра Тетріс, яка заповнила ринок всіма її варіаціями і утримує високі позиції до теперішніх часів, розроблені її версії як для ігрових автоматів, так і для всіх програмних платформ та сучасних мобільних пристроїв.

Комп'ютерні ігри в жанрі головоломки посідають досить високі позиції в загальному рейтингу найпопулярніших ігрових жанрів, відповідно до даних аналітичної компанії Newzoo (рисунок 1.8).

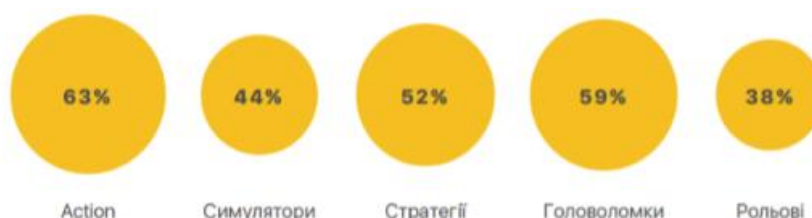


Рисунок 1.8 – Найпопулярніші ігрові жанри

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Часто розробники відеоігор використовують головоломки як додатковий елемент гри, наприклад, щоб пройти далі користувач повинен розгадати деякий пазл або знайти предмети для відкриття проходу, дверей, чи, навіть, наступного рівня гри.

Для створення комп'ютерних ігор також розроблено та активно використовується ряд ігрових рушіїв:

- Unity – вважається одним з лідерів у галузі. Є безкоштовна версія яка може бути використана для створення 2D і 3D ігор. Серед відомих ігор, зроблених в Unity: Temple Run, Deus Ex: The Fall, Assassins Creed: Identity;

- Unreal Engine 4 є ще одним з найбільш просунутих рушіїв для розробки ігор. Він доступний безкоштовно, але потрібно оплачувати 5% податку як тільки ваш дохід перевищить планку в 3000 \$;

- Phaser – це HTML5 рушій з відкритим вихідним кодом, останнім часом набуває популярності. Непоганий варіант для тих, хто бажає створювати веб–ігри;

- CryEngine – один з найпопулярніших ігрових движків, про що свідчить кількість «великих ігор» створених на ньому: Sniper II: Ghost Warrior, Cabal II, Ryse: Son of Rome, Far Cry. Він привернув до себе увагу розробників після виходу Crysis.

Unity дозволяє програмувати на декількох мовах, а саме C #, JavaScript. Існує рішення для спільної розробки – Asset Server.

Відповідно до останніх статистичних даних, обсяг світового ринку комп'ютерних ігор невинно збільшується. Останні дані, опубліковані аналітичною компанією Newzoo подано на діаграмі (рисунок 1.9).

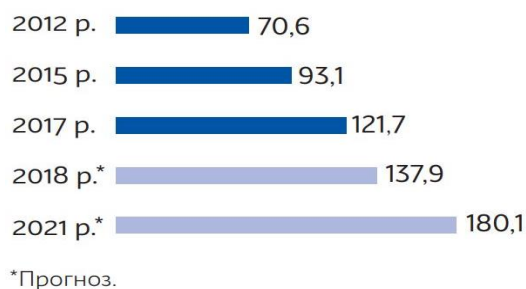


Рисунок 1.9 – Обсяг світового ринку комп'ютерних ігор у 2012–2021 рр.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Протягом останніх років тенденцією серед технологічного суспільства є зміна якості та насиченості ігрового контенту. Люди більше не хочуть купувати однотипні ігри, тому розробникам потрібно кожного разу вигадувати нові ідеї для їх проєктів. Саме жанр головоломок надає як користувачу, так і розробнику ігор широке поле для діяльності в плані хорошої графіки, цікавого сюжету та розвитку логічних здібностей.

1.5 Постановка задачі.

Проаналізувавши наявні ігрові продукти на ринку та виходячи з техніко–економічної характеристики об’єкту інформатизації, ігрова програма що розробляється повинна володіти певними особливостями та відповідати таким вимогам:

- простий і зрозумілий інтерфейс користувача;
- наповнення цікавими задачами–головоломками, в кращих традиціях жанру;
- наявність декількох рівнів складності;
- хороше та продумане графічне оформлення рівнів;
- зручна система управління персонажем.

Функції які повинен виконувати ігровий продукт :

- управління головним персонажем: дозволить користувачеві контролювати дії персонажа в грі, рухатися по ігровим локаціям;
- коректна взаємодія між головним персонажем гри та головоломками;
- стабільна робота ігрового продукту при некоректних діях користувача;
- в міру освітлені рівні для комфортного ігрового процесу.

Використовуючи клавіші для управління персонажем, користувач має змогу не тільки переміщатися по локації а й взаємодіяти з предметами оточення, такими предметами є різні панелі управління у яких користувач повинен вводити данні, артефакти які використовуються для та міні головоломки.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Опис предметної області

Головоломка – найменування жанру відеоігор, завданням яких є вирішення логічних проблем, що просять від гравця використання спостережливості, вдумливості, логіки і стратегії. Ігри цього жанру виникають за прикладами настільних і механічних загадок: ребуси, пятнашок, маджонга, кросвордів і т. п. Загадки можуть інтегруватися в ігри других жанрів як важливі компоненти ігрового процесу або для його різноманіття. [6]

Отже, в результаті аналізу комп'ютерних ігор–головоломок, було спроектовано основні функції гри. На рисунку 2.1 подано діаграму варіантів використання.

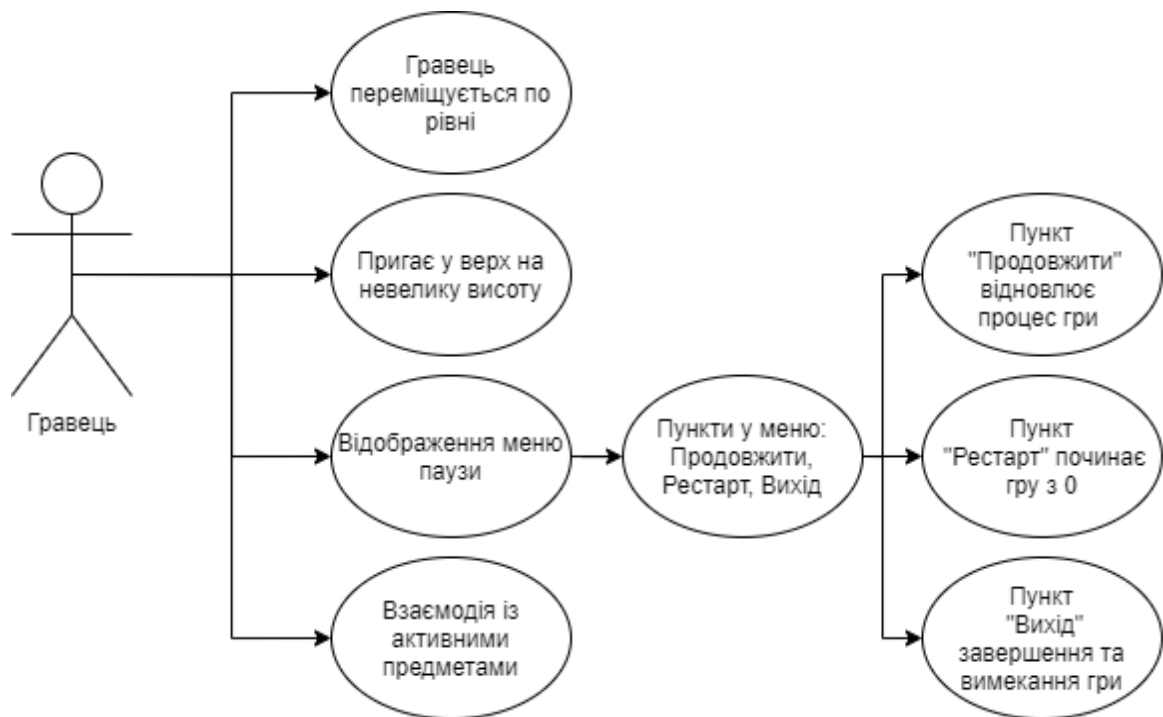


Рисунок 2.1 – Діаграма варіантів використання

Ігровий процес відбувається шляхом проходження персонажем локаціями та рівнями гри, розгадуючи при цьому цікаві задачі–головоломки. Перехід на

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

наступний рівень відбувається після розгадування набору головоломок на попередньому. Керування персонажем здійснюється стандартними засобами – з допомогою миші або клавіш на клавіатурі. Персонаж в процесі гри може взаємодіяти з предметами та іншими героями гри. Можливе збереження стану гри при досягненні певних точок збереження.

2.2 Розробка сюжету гри

Загалом, створення сюжету гри є важливим елементом проєктування та розробки, оскільки саме від вдалого сюжету залежить, чи зацікавиться користувач грою, чи вибере її серед подібних, на скільки вона буде популярною на ринку. Отже, створення сюжету гри можна розділити на декілька основних кроків.

Крок 1. Збираємо інформацію для майбутнього сюжету.

Проходження першого кроку методики тісно пов'язане з вхідними даними (жанр гри, літературно–художній стиль, цільова аудиторія та ін.) І складається з трьох дій:

- вивчення вхідних даних;
- виявлення типових характеристик цих даних;
- складання власного списку асоціацій, пов'язаних з вхідними даними.

В результаті успішного здійснення трьох цих дій розробник отримає набір елементів для будівництва сюжету. Виявлені розробником типові характеристики вхідних даних, а також особисті асоціації з ними – зведені всі разом в один список і складуть цей набір.

Крок 2. Формуємо сюжетний ланцюжок.

Найважливіше, що отримає розробник, завершивши другий крок, – це структура сюжету. Якщо серед вхідних даних присутній ключовий задум сюжету, то потрібно використовувати його. Якщо немає – необхідно цей сценарій створити самостійно.

Тепер, в залежності від ряду факторів, найважливіший з яких – величина майбутньої гри, необхідно створити структуру сюжету. Чим об'ємніший

					ДП.КН.20.416.20.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

передбачається сюжет, тим з більшої кількості частин він буде складатися. Для короткого сюжету маленької гри підійде проста структура – по одній–дві частини на зав'язку, основна дія, кульмінація і розв'язка.

Дотримуючись описаної вище логіки, потрібно розбити структуру сюжету на певну кількість частин. Виконуючи крок 3, рекомендовано дотримуватися цієї структури.

Крок 3. Детальний опис сюжету

На початку кроку 3 в розпорядженні обов'язково є:

- набір елементів, виділених в ході кроку 1;
- ключовий задум і структура сюжету (результат кроку 2).

Готовий сюжет – ось що розробник отримаєте після закінчення кроку 3.

Отже по сюжету головного героя звать "Том", самі дії відеоігри будуть починатися з того що головний персонаж прибуває на дачу до свого діда, прибувши на дачу вже настала ніч тому діставши з сумки ліхтар він пішов в один з будинків де горіло світло але у будинку нікого не було, оглянувши всю місцевість у пошуках діда Том помічає дивне світіння каменів та завалений прохід у стару шахту, на одному із каменів було розташовано пристрій схожий на вибухівку, біля якої був розміщений уривок якогось коду, з цього моменту і починаються пригоди гравця.

Всього у грі є 5 рівнів пройшовши які гравець попадає у фінальну кімнату де знаходиться "мудрець" головного героя та скарби. На першому рівні гравцеві потрібно знайти і ввести усі кусочки паролю у правильній послідовності щоб відкрити прохід на наступний рівень.

У другому рівні "Том" попадає у невеликий тунель в якому розміщені кристали які світяться синім кольором та невелику головоломку у якій потрібно повернути фрагменти пазлу так щоб вийшло цільне зображення.

Третій рівень відрізняється тим що він заповнений густим туманом та великою кількістю рослин, єдиний шанс пройти рівень можна слідкуючи за зеленими кристалами які будуть виводити гравця із туману.

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

Четвертий рівень доволі великий але зовсім не складний так він виглядає як великий хвилястий каньйон на стінах яких ростуть дерева та кущі а по центру висять у повітрі невеликі платформи які світяться синім кольором, коли персонаж попадає на платформу та в свою чергу через деякий час падає у прірву, тому гравець мусить як найшвидше пройти його.

П'ятий рівень подібний на третій з тими відмінностями що відсутні зелені кристали та ціллю його є розміщення усіх "артефактів" на свої місця після чого гравець переноситься у фінальний рівень. У фінальному рівні "Том" попадає у невелику кімнату в якій знаходиться велика кількість скарбів та " мудрець " головного персонажа. Метою гри є знайти " мудреця" головного персонажа після чого гра буде пройдена.

2.3 Обґрунтування технологій та засобів реалізації

Основним засобом реалізації вищевказаного проекту обрано ігровий двигун Unity тому, що – це мультиплатформенний інструмент для реалізації двох і тривимірних додатків та ігор, функціонує він під операційними системами OS X і Windows. Створені з підтримкою Unity додатки функціонують під операційними системами OS X, Windows, Android, Windows Phone, Apple iOS, Linux, а також на ігрових приставках PlayStation 3, Wii і Xbox 360. Є здатність створювати інтернет – додатки з підтримкою додаткового модуля до браузеру Unity. Додатки, створені з підтримкою Unity, підтримують OpenGL і DirectX.

Цей ігровий рушій популярний серед інді-розробників. Перше, що цілком очевидно, в одному середовищі ви можете створити додаток, який буде запускатися майже на всіх пристроях. І, звичайно ж, подібні плюси дуже ефективно привертають шанувальників серед юних розробників. [4]

А також майже що вірусне поширення цього рушія. Тут грає свою роль той факт, що продукт дуже якісний та простий для розуміння. Грає роль і те, що кожен розробник вважає своїм обов'язком викласти прототип своєї гри на улюблений форум, файлообмінник і торрент-трекер, додавши і свою краплю в морі ігор на Unity.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Актуальні статистичні дані щодо популярності ігрових рушіїв подано на рисунку 2.2.

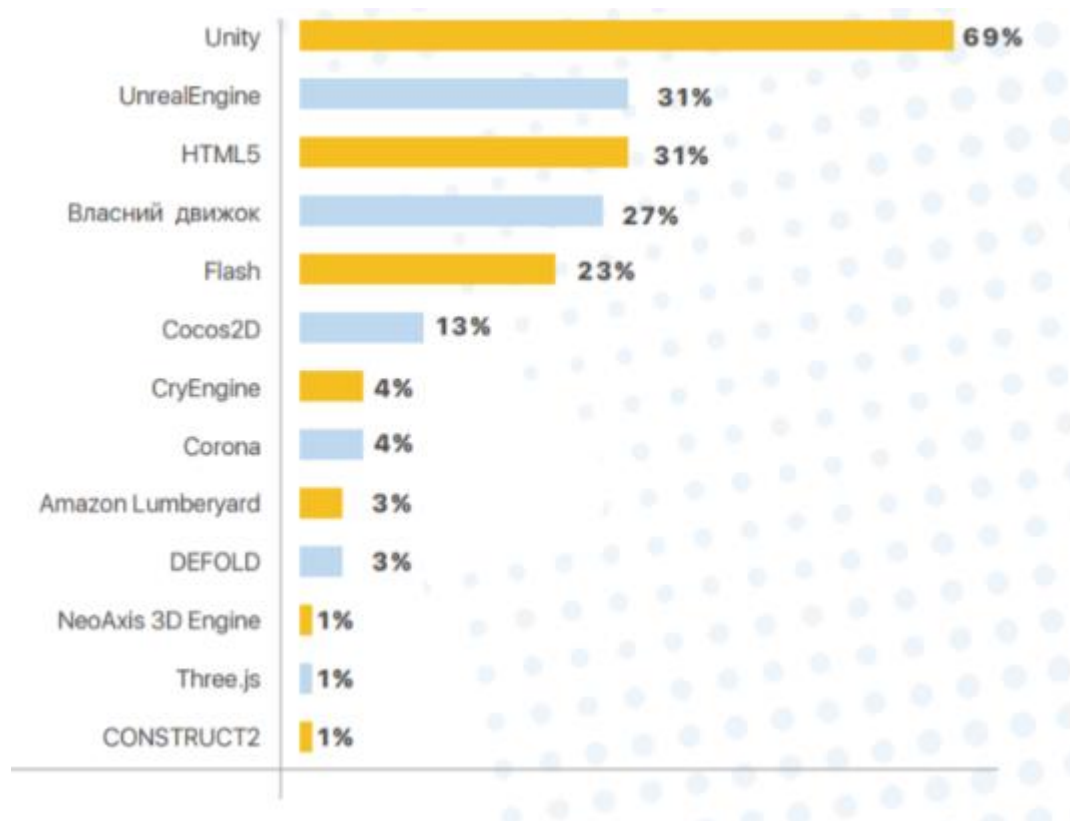


Рисунок 2.2 – Рейтинг популярності технологій створення комп'ютерних ігор

Переваги використання Unity:

- мова C#. Дана мова програмування є простою для розуміння і дозволяє програмісту легко увійти в розробку гри. Це важливий момент, тому що на відміну від інших «двигунів», де використовується мова C++, у C# є багато елементів і прийомів, які вже реалізовані, і програмісту потрібно тільки скористатися ними;
- Asset Store, де є величезна кількість різних доповнень і ресурсів для створення гри. Одні з них безкоштовні, а інші платні, але всі вони зібрані в одному місці зі зручним пошуком та можливістю завантажити, інтегрувати і отримати відразу робочий функціонал;
- кросплатформеність, тобто один і той же код, написаний на «двигуні» Unity, з мінімальними змінами може бути перенесений на різні платформи (PC,

					ДП.КН.20.416.20.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

Mac, Android, iOS, Web, ігрові консолі). Це величезний плюс, який скорочує час на розробку гри в кілька разів;

- Community. Це означає, що у різних функцій «двигуна» є чіткий опис з прикладами на сайті розробника, звернутися до якого можна в будь-який момент.

Для створення деяких графічних елементів у грі використано редактор MagicaVoxel.

MagicaVoxel це воксельний Open Source редактор. Він містить інструменти для воксельного моделювання та хороший рендер, який дозволяє налаштувати матеріали і експортувати їх разом з текстурами і тінями. [9]

Основні фактори, які вплинули на вибір даного редактора:

- редактор доступний на системах Windows і Mac OS і дозволяє експортувати результат роботи в популярні 3d формати і навіть 2d спрайт;
- крім стандартних інструментів редагування, реалізована підтримка введення команд, що дозволяє створювати дійсно складні структури.

Середовище програмування Visual Studio.

Microsoft Visual Studio – добірка товарів фірми Microsoft, що мають можливість інтегрованого середовища для реалізації програмного обслуговування і перелік інших вельми корисних інструментів. Цей продукт дозволяє розробити як програми з візуальним інтерфейсом так і консольні програми, а також веб-додатки, веб-сайти, веб-служби в керованих кодах для всіх платформ, які підтримуються Microsoft. [1]

Графічний редактор Blender. [8]

Blender використовується для створення високоякісних 3D моделей та графіки яка включає такі засоби як:

- моделювання;
- анімація;
- розкадрування;
- обробка відео;
- розробка відеоігор.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

Особливості програми це невеликий об'єм, який вона займає у пам'яті, висока швидкість створення полігонів і деталізації та функціонування під вуликою кількістю операційних систем, таких як: Mac OS, Linux, Sun Solaris, Microsoft Windows, Pocket PC, MorphOS та ін. [7]

Програма містить певний функціонал: динамічність рідини, твердих та м'яких тіл, систему швидких кнопок, величезна кількість різноманітних безкоштовних та платних додатків, розроблених на мові програмування Python.

Переваги цієї програми:

- підтримка різних математичних алгоритмів, включаючи низько або високополігональні моделі, криві «Безьє», обрізання лишніх полігонів, шрифти векторів та metaballs;

- інструмент розкадровки, між якими Kinematics, скелетна та деформація по сітці, нелінійна розкадровка, Weighting Vertex, Timeline, динамічність м'яких тіл, включаючи опис колізії структури об'єкта при взаємодії, динамічна рідина, Bullet, динамічність твердих тіл, системи керування волосся та концепції частинок і система частинок при описі колізії об'єкту;

- Game Blenders – окреми проєкт від Blender, який дає інтерактивні можливості, наприклад визначення колізії об'єкта, двигун динаміки із програмованою логікою. Отже ця програма може перетворювати у реальному часі додатки починаючи від архітектурної матеріалізації до відео ігор.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ІГРОВОГО ПРОДУКТУ

3.1 Реалізація інтерфейсу ігрового продукту

Для створення інтерфейсу у грі було застосовано мову програмування C# та програмне середовище для створення ігор Unity.

Створення інтерфейсу починається з дизайну та підбору шаблонів для головного меню гри. Кнопки головного меню створюються за допомогою діалогового вікна, створивши дві сцени, одна з яких буде першим рівнем гри, а друга сцена вже буде являти собою головне меню гри. [5]

На сцені гри яку обрано стартовим екраном при запуску гри, перейшовши в режим 2D розміщаєм кнопки для переходу в різні розділи меню. Показано на рисунку 3.1.

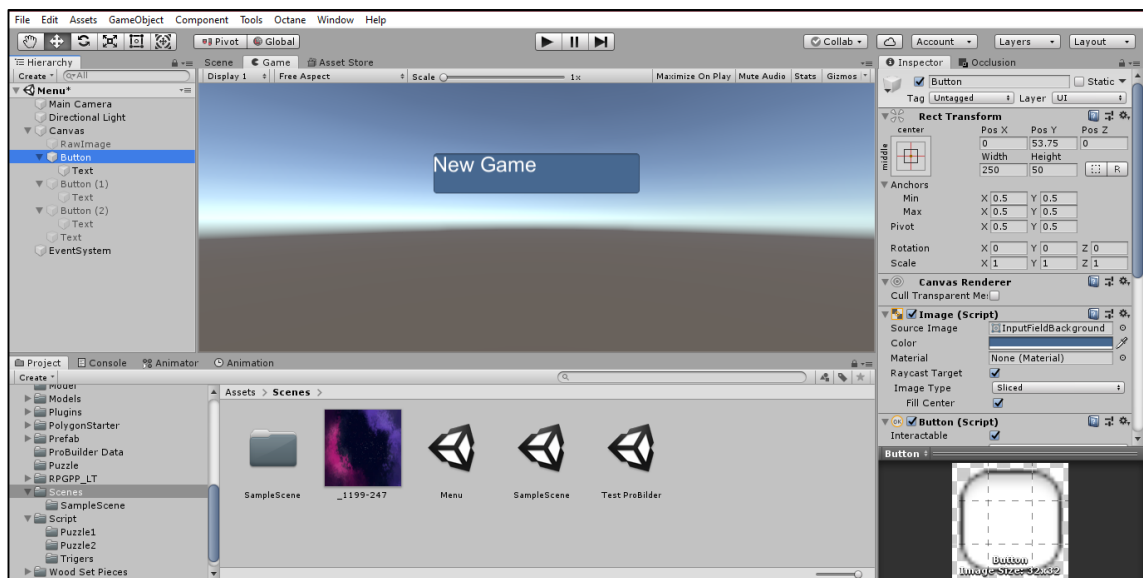


Рисунок 3.1 – Розміщення кнопок головного меню

Потім потрібно змінити дизайн та шрифт кнопки. Щоб змінити дизайн кнопок для більш гарного вигляду необхідно у полі Inspector → Image → Source Image обрати необхідний стиль для кнопки розробивши власний дизайн для кнопки, або ж є можливість простого використання файлів картинок які є можливість скачати з інтернету, показано на рисунку 3.2.

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

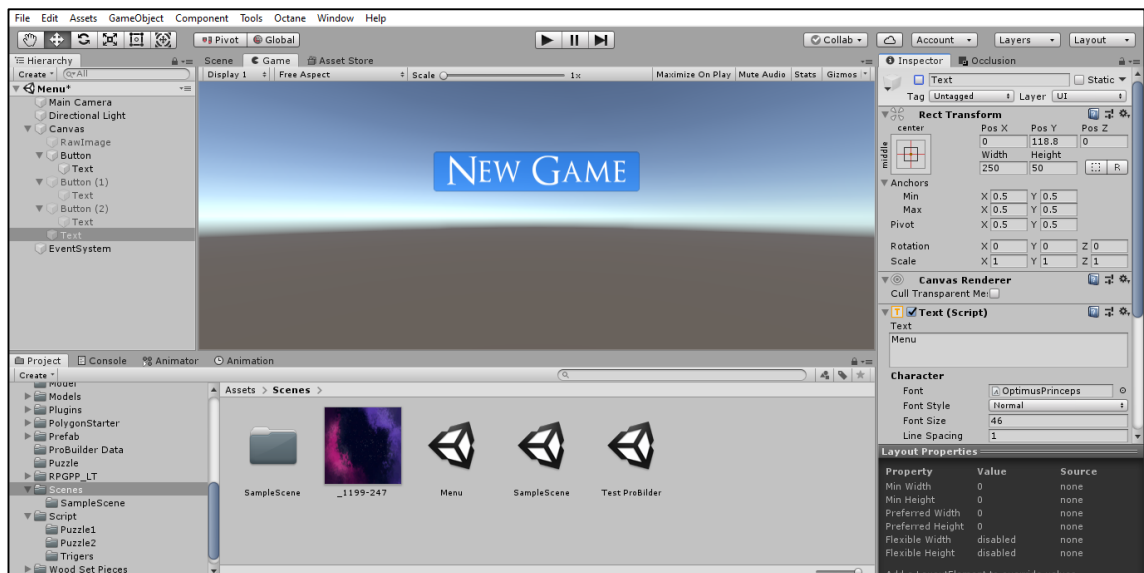


Рисунок 3.2 – Вибір стилю для кнопок меню

Для найкращого вигляду стартового екрану є можливість додавання різних анімацій і картинок на фон. Для додавання іншого фону і анімації в Unity передбачені наступні дії. У папці з файлами гри – Assets , додаємо елемент UI інтерфейсу “RawImage” і беремо перетягуємо фон в параметр “Texture”, показано на рисунку 3.3.

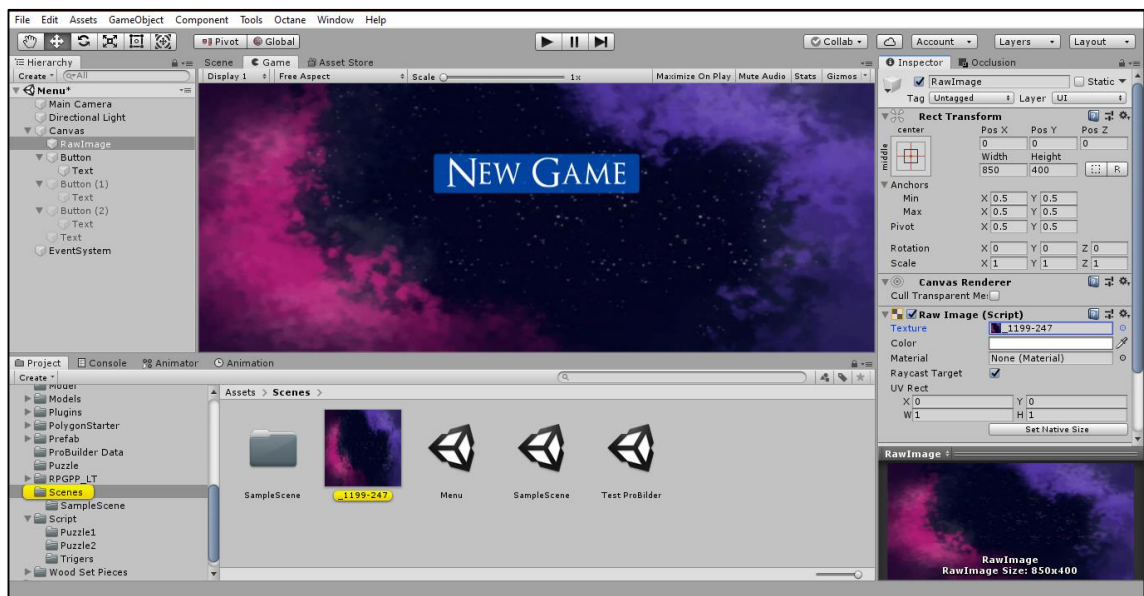


Рисунок 3.3 – Зміна фону головного меню гри

Змн.	Арк.	№ докум.	Підпис	Дата

ДП.КН.20.416.20.000 ПЗ

Арк.

27

Для надання інтерактивності створеним кнопкам меню необхідно додати скрипт для їх використання. Результат подано на рисунку 3.4.

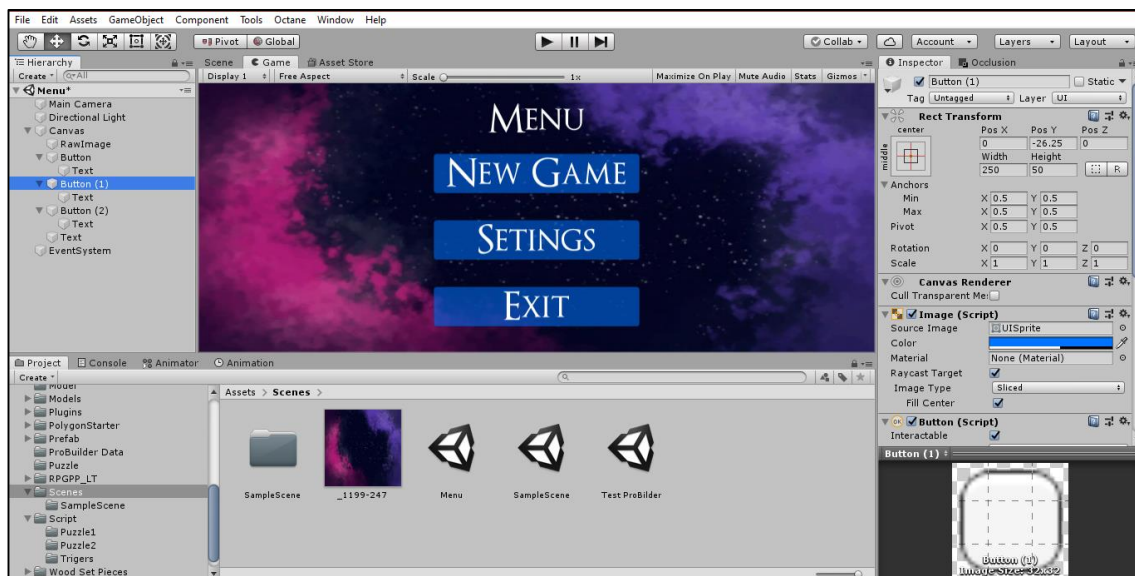


Рисунок 3.4 – Демонстрація головного меню з фоном

Додавши в основній папці з грою C# script та прописавши в ньому програмний код, за допомогою якого користувач натиснувши на кнопку “New Game” перейде в сцену з першим рівнем гри. Невеликий код для меню подано у лістингу 3.1, сам програмний код подано в додатку А.

Лістинг 3.1 – Код меню

```
Public class Mennu : MonoBehaviour{
    Public void StartGame(){
        Application.LoadLevel(1);
    }
    Public void Settings{
        Application.LoadSettings(1);
    }
    Public void ExitGame(){
        Application.Quit();
    }
}
```

Таким чином була спроектована та розроблена система керування головним меню завдяки якому можна налаштувати якість зображення та клавіші керування персонажем.

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

3.2 Реалізація рівнів гри

Для створення ігрової локації було використано такий компонент як «Terrain» та готові «prefabs». Елементи Terrain використовуються для створення рельєфу дерев і трави, а для побудови рівнів використовувалися заготовки 3D моделей тобто (prefabs).

Завдяки ним було створено основний рельєф рівнів та задано напрямок по якому буде рухатися гравець.

Нижче наведено приклад створення рельєфу в рушії Unity. Рисунок 3.5.

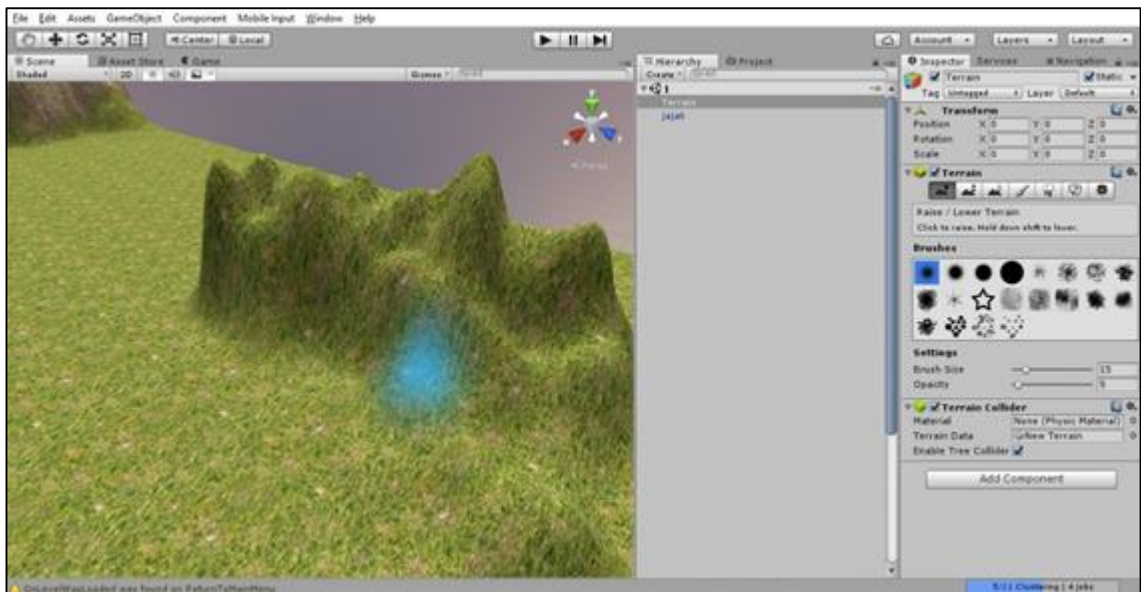


Рисунок 3.5 – Створення рельєфу

Для наданням "горам" та іншим об'єктам цього типу певного кольору або малюнку використовуються текстури у вкладці кисть. Щоб досягти більшої правдоподібності горам потрібно надати їм відповідної текстури у налаштуваннях до місцевості.

Для того щоб нанести текстуру на місцевість необхідно додати варіанти текстур в вікно «Textures» і вже після цього вибираючи з представленого списку обирати кисть і розподіляти обрану текстуру по території.

На рисунку 3.6 наведено приклад накладання текстури на гірський масив.

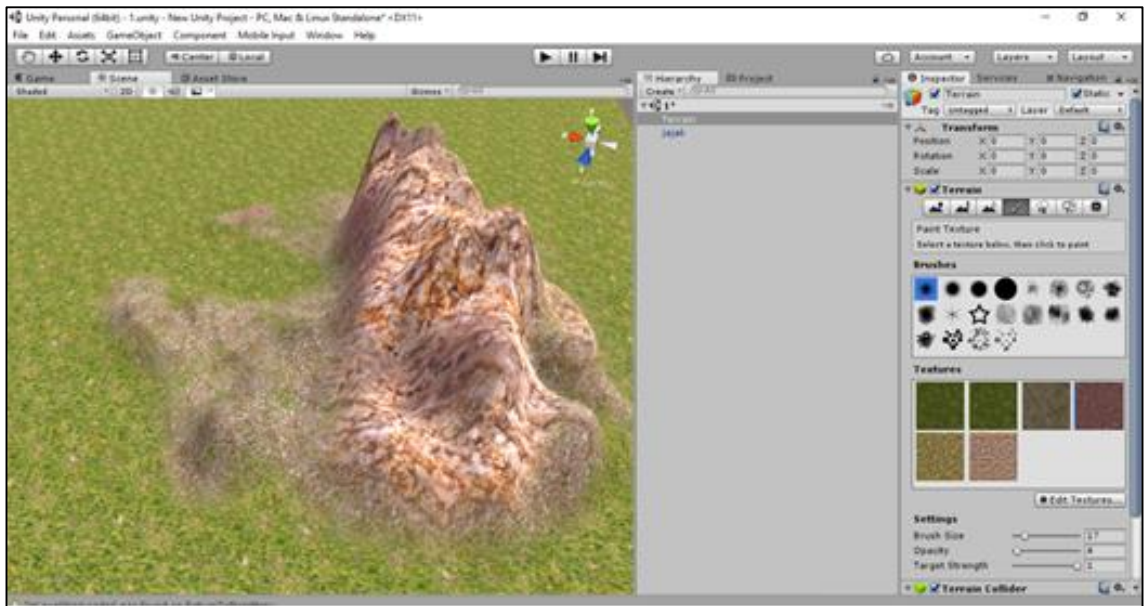


Рисунок 3.6 – Накладання кам'яної текстури на гори

Для накладання всіх інших текстур на ландшафт використовується така сама послідовність дій. Головною умовою є вибір якісних текстур з високим розширенням які будуть виглядати найбільш подібно до реальних. Нижче наведено приклад вибору текстур в меню налаштуваннях Unity, рисунок 3.7.

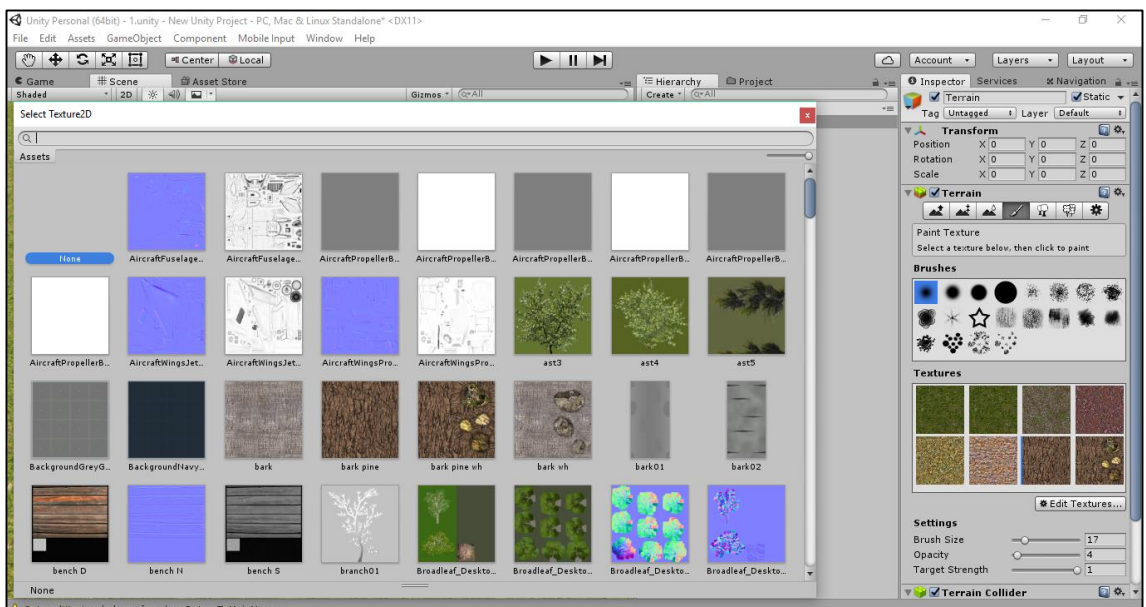


Рисунок 3.7 – Приклад текстур

Для реалізації рослинності на ігровій карті були створені 3d моделі дерев, кущів та трави. Також для більшої різноманітності були взяті деякі моделі рослинності з додатку «Assets Store» який є обов'язково присутній в ігровому

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

рушії Unity. Щоб додати окрему рослину на територію ландшафту створюємо в кореневій папці гри окрему папку в яку добавимо 3d моделі рослин. Нижче показано папку з перенесеними в неї моделями дерев, рисунок 3.8.

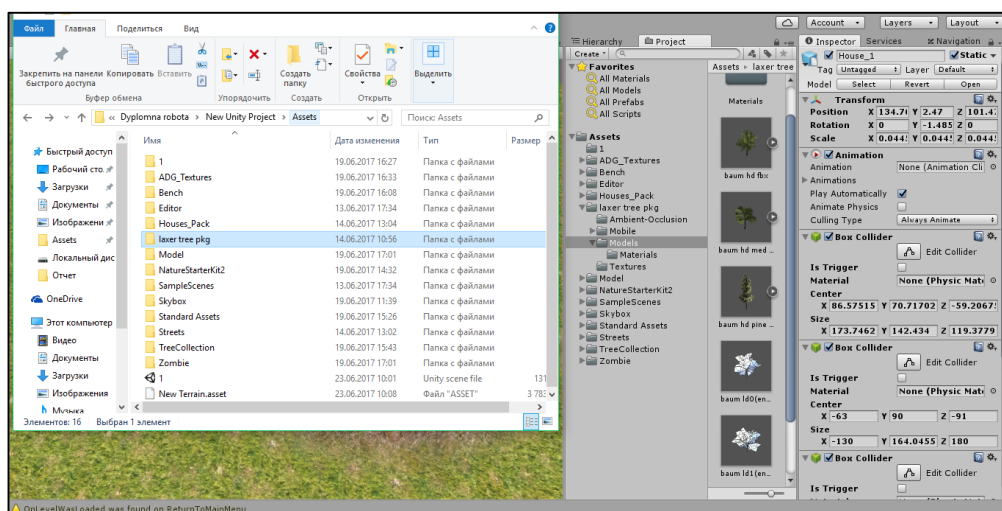


Рисунок 3.8 – Перенесення моделей дерев в кореневу папку

Додавати рослинність на місцевість можна по одинці чи наносити масивом з різною щільністю та висотою об'єктів, дана функція показана на рисунку 3.10. Щоб змінити показники щільності, висоти, випадковості розміщення а також різноманітності кольорів використовується панель «Inspector» – «Settings». Нижче наведений приклад накладання рослинного масиву на місцевість, рисунок 3.9. Синім кольором показані межі накладання масиву на ландшафт.

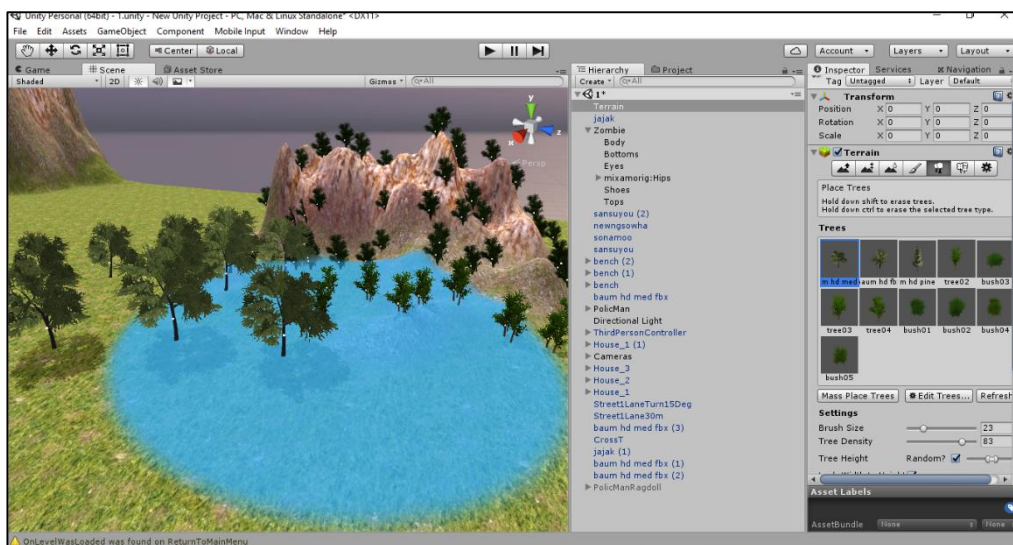


Рисунок 3.9 – Накладання рослинного масиву на місцевість

Змн.	Арк.	№ докум.	Підпис	Дата

ДП.КН.20.416.20.000 ПЗ

Арк.

31

Щоб гравець не губився на рівні потрібно задати напрямок руху такими методами як:

- освітлення;
- дизайн карти.

Враховуючи, що у грі буде імітуватися ніч, на рівні буде доволі темно і потрібно дати хороше освітлення і головне якісне його розташування, щоб можна було направити гравця в потрібний напрямок. Для освітлення карти використовувалися такі об'єкти як «point light». Їхнє освітлення можна доволі гнучко використовувати як для створення нічних ліхтарів так і для м'якого освітлення кристалів які будуть показувати шлях гравцеві. Приклад освітлення у грі зображено на рисунку 3.10.

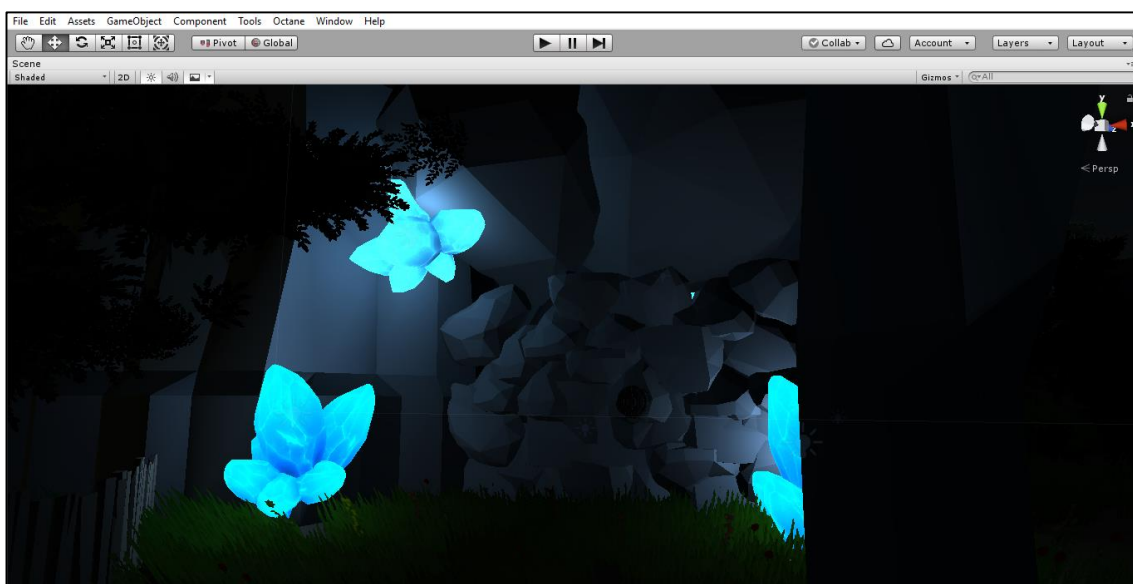


Рисунок 3.10 – Практичне використання освітлення

Для того щоб створити "point light" потрібно перейти на сцену та натиснути на випадаюче вікно "Create" – "Light" – "point light". Для збільшення інтенсивності, кольору та радіусу світла потрібно перейти в налаштування «Inspector» – «Settings». Налаштування освітлення зображено на рисунку 3.11.

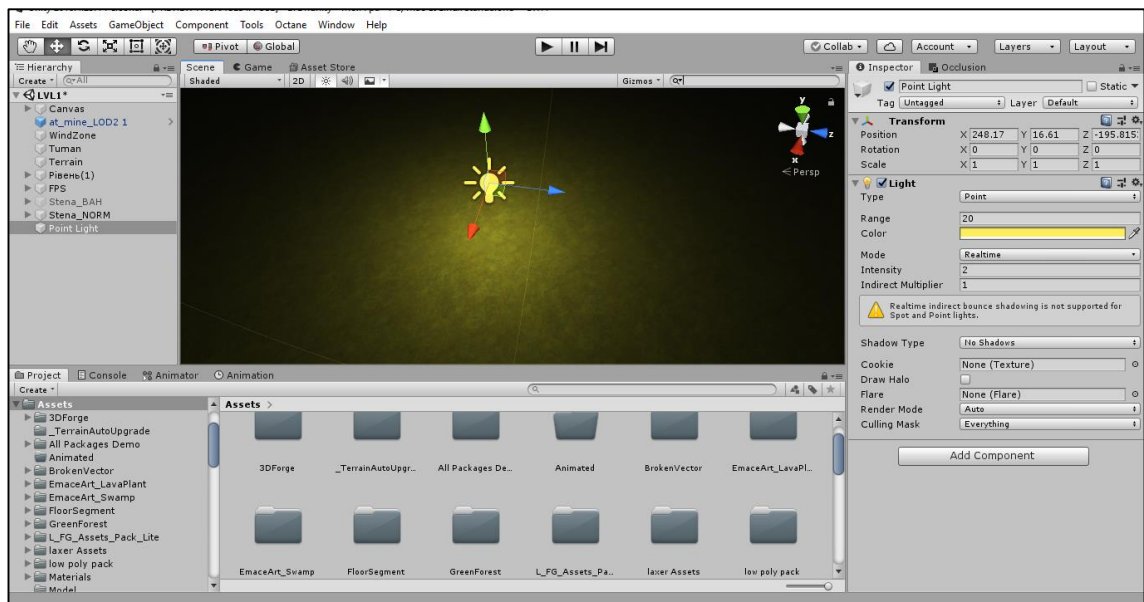


Рисунок 3.11 – Налаштування освітлення

Дизайн карти це чи не найважливіше у грі, тому що те, як гравець буде переміщуватися по рівні, залежить від того, як добре зроблена карта та розміщені ті чи інші об'єкти на рівні. Для того, щоб зробити дизайн карти, потрібно мати об'єкти для його створення, в даному випадку використовувалися асети з "Asset store" на стіни та будівлі. Приклад дизайну рівня зображено на рисунку 3.12. [2]

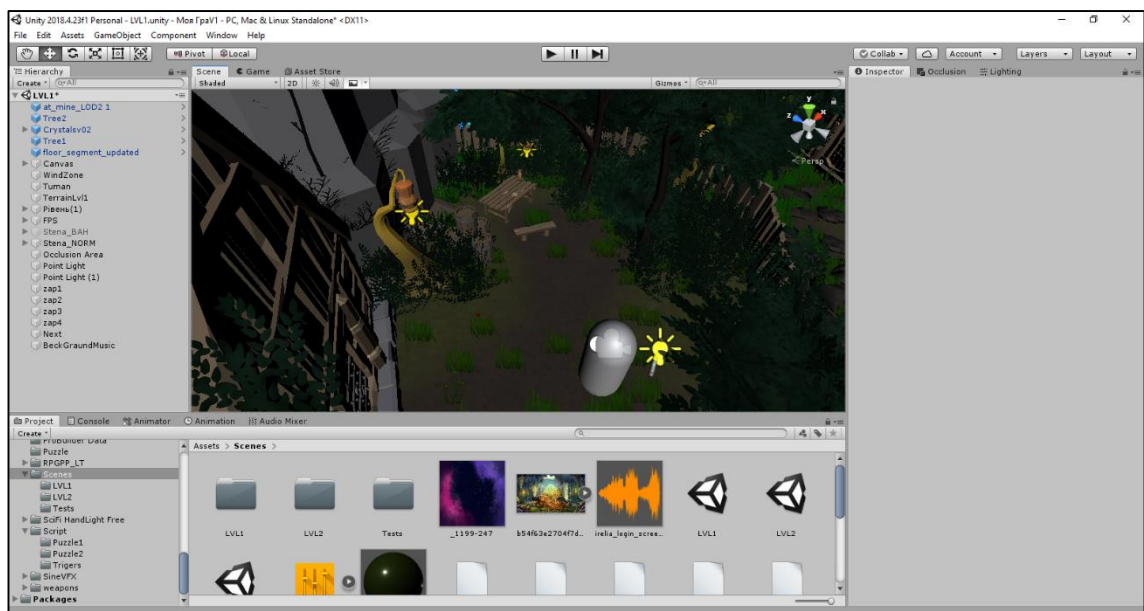


Рисунок 3.12 – Дизайн рівня

Для ускладнення життя гравцеві було розроблено такі перешкоди:

– закодовані двері;

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

- складення окремих деталей в зображення;
- переплигування з одної платформи на іншу;
- знаходження певних предметів для відкриття проходу;
- знаходження правильного шляху в лабіринті з туманом за допомогою кристалів.

Перешкода «Закодовані двері». Гравець повинен відшукати правильну комбінацію символів для відкриття проходу. Код розбитий на частини по всьому рівні, які знаходяться на уривках папірусу.

Коли гравець відшукав всі частинки та склав правильну комбінацію з них, тоді він введе цю комбінацію у панель для коду. Після чого двері або прохід відчиниться і гравець просунеться далі до фіналу гри. Приклад перешкоди «Закодовані двері» зображено на рисунку 3.13.

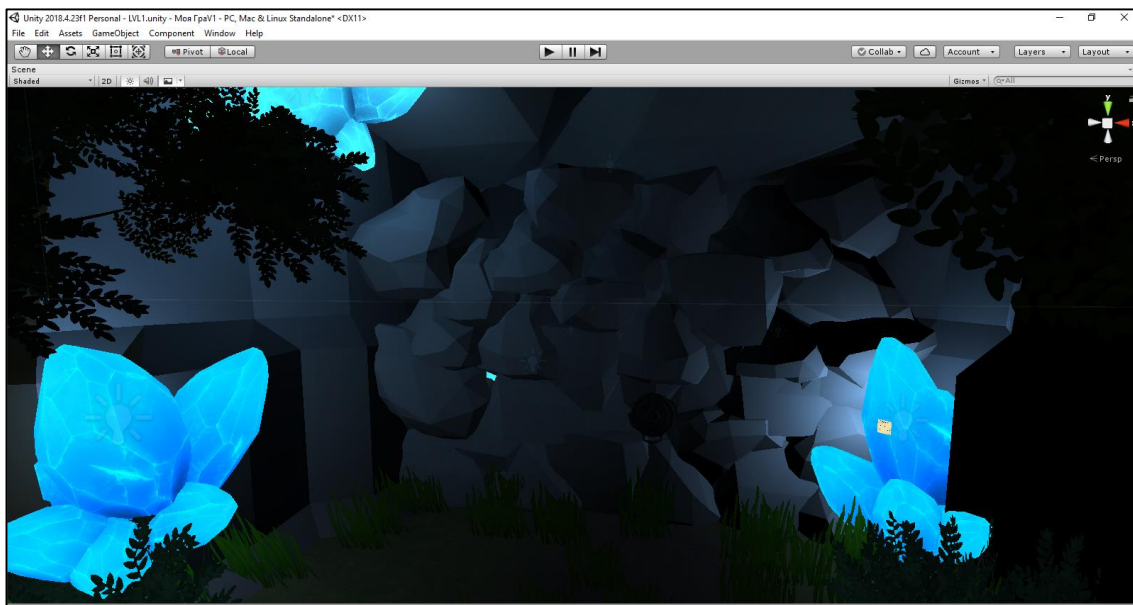


Рисунок 3.13 – Перешкода «Закодовані двері»

Перешкода «Складення окремих деталей в зображення». Гравцеві потрібно переставити частинки зображення у правильній послідовності щоб утворилося коректний рисунок. Після того як гравець розгадає цю головоломку, тоді відкриється прохід до наступного рівня. Приклад перешкоди відтворено на рисунку 3.14.

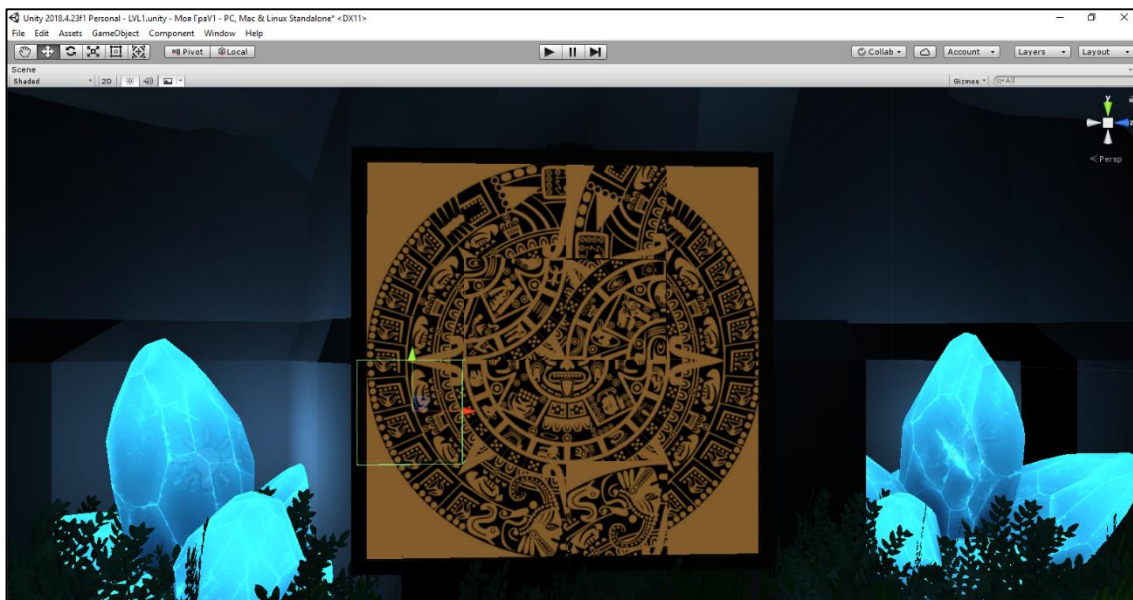


Рисунок 3.14 – Складення коректного зображення

Перешкода «Переплигування з одної платформи на іншу». Гравець повинен перейти з точки А в точку В, під час того він буде переплигувати з однієї платформи на іншу.

Коли гравець плигає через платформи, то вони з різним інтервалом часу падають у прірву, час спрацьовування платформ регулюється певним параметром, який задається у коді гри або у редакторі сцени. Використання цієї перешкоди зображено на рисунку 3.15.

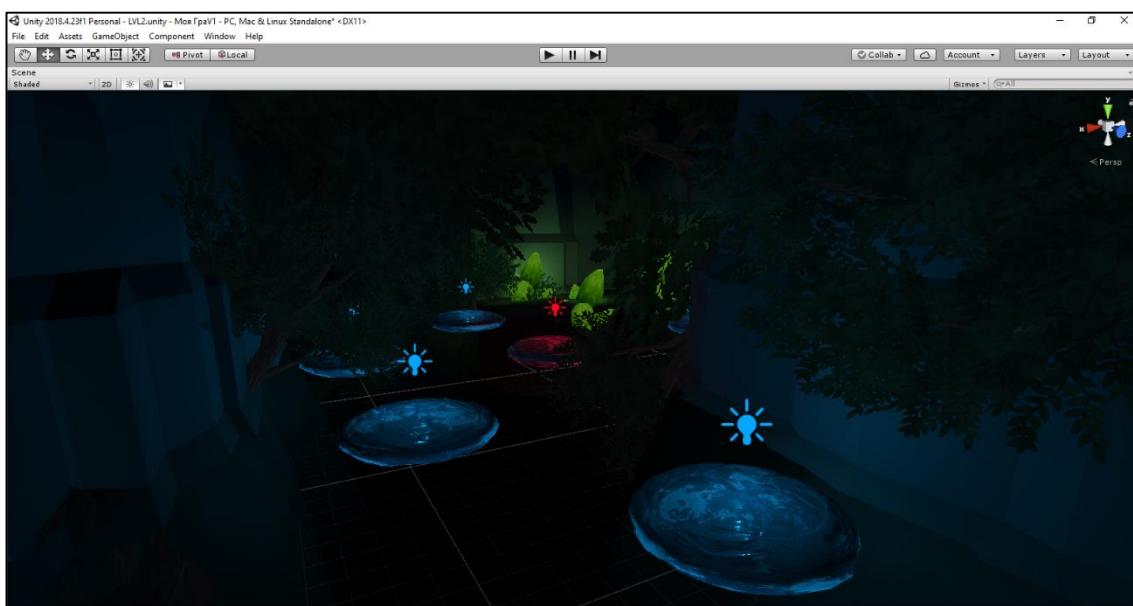


Рисунок 3.15 – Перешкода «паркур»

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

Перешкода «Знаходження певних предметів» для відкриття проходу. На спеціальному рівні знаходяться деякі предмети (артефакти), які потрібно розмістити на певні місця на рівні. Артефакти виглядають як малі, зелені куби, які світяться у темряві для їхнього зручного пошуку.

Щоб розмістити їх у правильних місцях гравцеві потрібно натиснути, та утримувати ЛКМ (ліву клавішу миші) для їхнього перенесення між рівнем, якщо гравець відпустить ЛКМ, тоді артефакт, який утримував гравець, упаде на землю. Практичне використання цієї перешкоди зображено на рисунку 3.16.



Рисунок 3.16 – Пересування артефактів

Перешкода «Туманний лабіринт». Гравець опиняється на рівні, який заповнений густим, сірим туманом, кольоровими кристалами та різною рослинністю такі як: дерева, кущі, трава звичайна, суха трава, квіти жовті, червоні та фіолетові.

Щоб орієнтуватися на рівні «Туманний лабіринт», потрібно найти зелені кристали та слідувати за ними, щоб знайти правильний шлях та вийти з лабіринту, але якщо не йти за кристалами, тоді гравець може заблукати, та витратити більше часу на проходження рівня. Перешкода «Туманний лабіринт» зображено на рисунку 3.17.

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

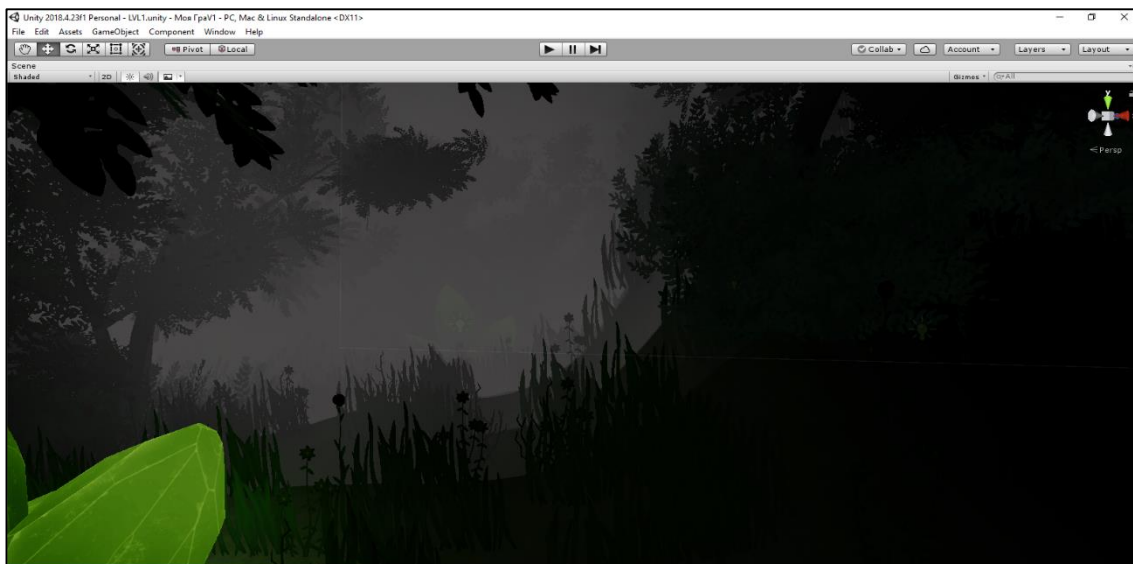


Рисунок 3.17 – Знімок екрана з рівня «Туманний лабіринт»

Таким чином, було реалізовано цілий ряд рівнів та локацій з різного типу головоломками для зацікавлення користувача під час проходження гри.

3.3 Реалізація рухів персонажів

Щоб персонаж при переміщенні по місцевості не провалювався в текстури є необхідність додавання колайдери на модель. У вкладці Inspector для моделі персонажу додається компонент «Capsule Collider». Колайдер по стандарту розміщується по центру моделі, тому необхідно розтягнути капсулу на всю висоту моделі. На рисунку 3.18 показано приклад накладення колайдери на модель, та налаштування розміру колайдери.

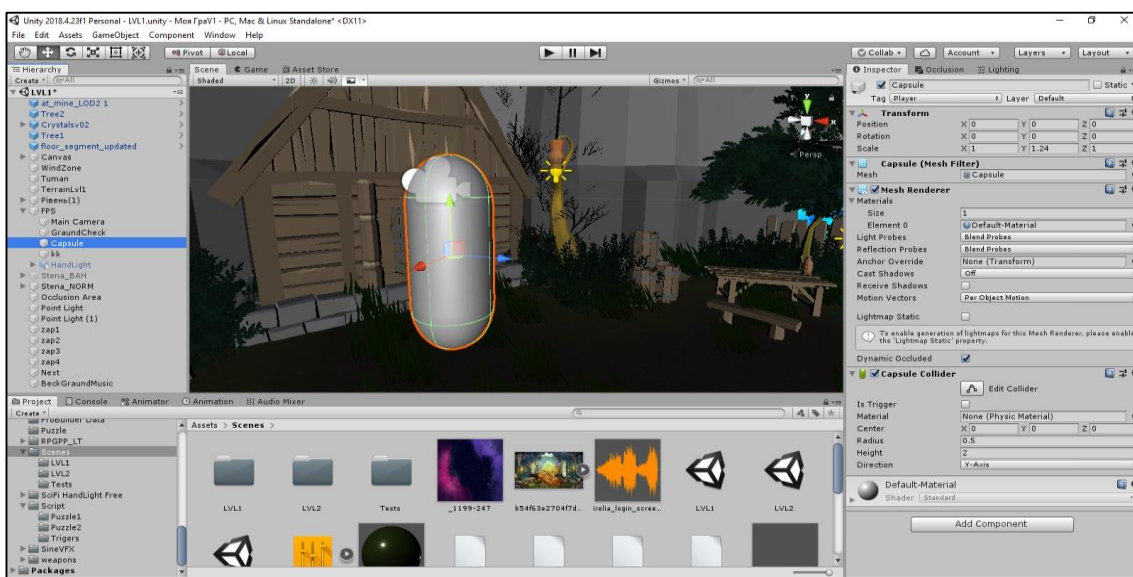


Рисунок 3.18 – Накладення колайдери на модель

									Арк.
									37
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.20.416.20.000 ПЗ				

Для реалізації рухів персонажа в ігровому рушії Unity було обрано мову програмування C#. Для того щоб модель виконувала повністю всі дії, які вказано в програмному коді необхідно створити «Ragdoll» для нашої моделі і прив'язати кістки до цього об'єкту. На рисунку 3.19 показано приклад накладення колайдери на модель, та налаштування розміру колайдери.

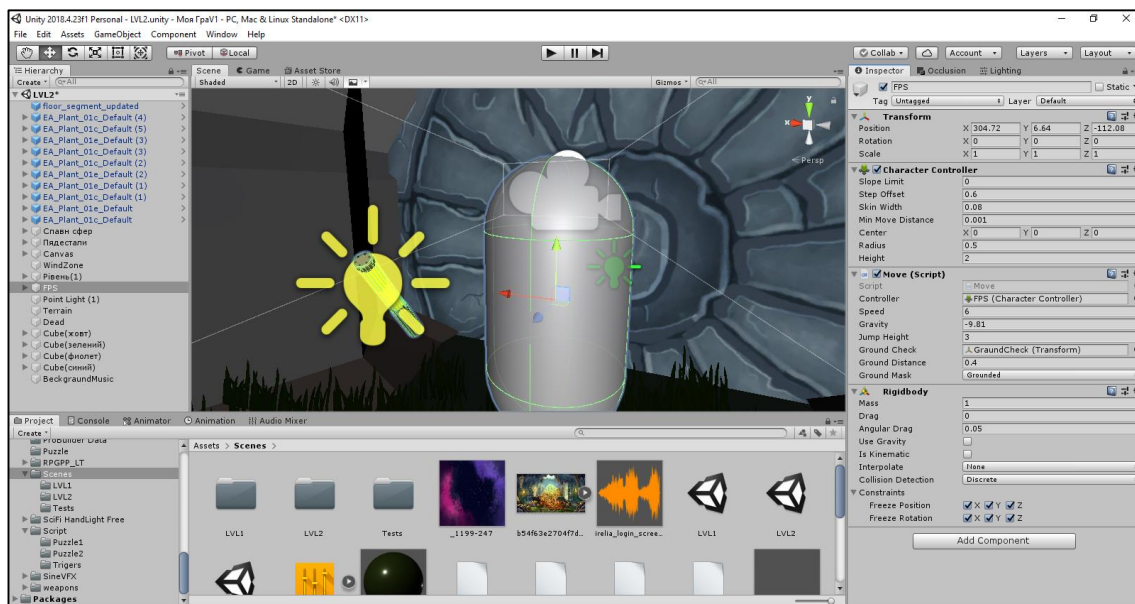


Рисунок 3.19 – Прив'язка регдолу до об'єкту

Наступним етапом створення було перенесення регдолу з ієрархії в папку моделей в основному файлі гри, і написання тригера який буде виконувати функцію смерті для персонажа.

Тепер, коли персонаж легко переміщується по локації і керується від першої особи було створено скрипт, за допомогою якого персонаж яким керує користувач помирав. Тобто зайшовши в заборонену область, або ж при отриманні значних ушкоджень від ворогів модель головного персонажа припиняє своє існування і на сцені з'являється повідомлення про те що ви померли. Під час потрапляння в зону дії скрипта, який в грі ми задаємо тригером персонаж загине. Таким чином, було реалізовано основні параметри персонажа, його реакції та дії. Програмний код подано в додатку А.

									Арк.
									38
Змн.	Арк.	№ докум.	Підпис	Дата					

4 ТЕСТУВАННЯ

Тестування систем та програм – метод постійної перевірки створеного продукту, у якому тестер має в опрацьованому програмою послідовності різноманітних наборів тестів із завідома відомими результатами. Тести підбираються так, щоб вони охопило як найбільше можливих ситуацій.

Основні види робіт з тестування:

- верифікація результатів розроблення програмного продукту на кожному етапі життєвого циклу;
- упорядкування плану тестування і підготовки тестів для перевірки окремих елементів розробленої програми та програми в цілому;
- керування виконанням тестів та аналіз результатів тестування;
- повторне тестування.

Мета тестування – виявити наявність помилок та недоліків. Іншими словами, це виявлення в першу чергу критичних помилок які приведуть ігровий проєкт то збою, далі знаходження помилок які впливають на ігрові об'єкти та сам процес і останнє це виявлення незначних орфографічних помилок або неточностей у цифрах.

Проведемо тестування створеного ігрового продукту на ігровому рушії Unity. Для цього спершу запусимо файл запуску гри "MyGame.exe".

При пуску гри, перше, що бачить користувач це стартове меню гри.

На стартовому екрані гри реалізовано 3 інтерактивні кнопки, при натисканні на котрі відбуваються певні дії, кнопка «New Game» відповідає за те щоб користувач переключився на стартову локацію гри яка представлена невеличким селом біля лісу, меню гри показано на рисунку 4.1.

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

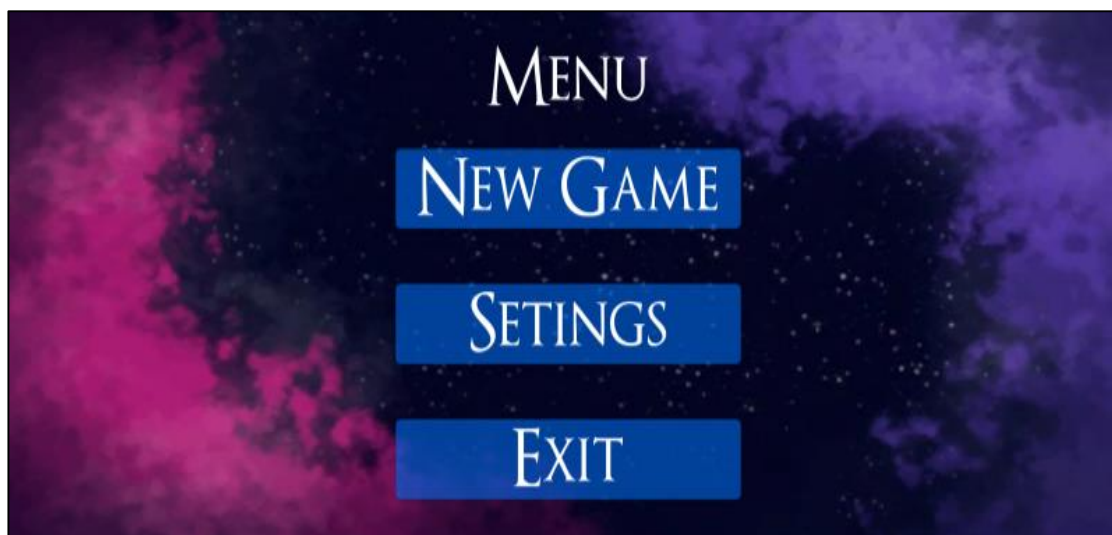


Рисунок 4.1 – Головне меню гри

При натисненні кнопки «New game» починається загрузка першого рівня гри. Наступним кадром, який бачить користувач, є сцена, в якій головний персонаж знаходиться в невеликому поселенні біля лісу в ночі, представлено на рисунку 4.2.

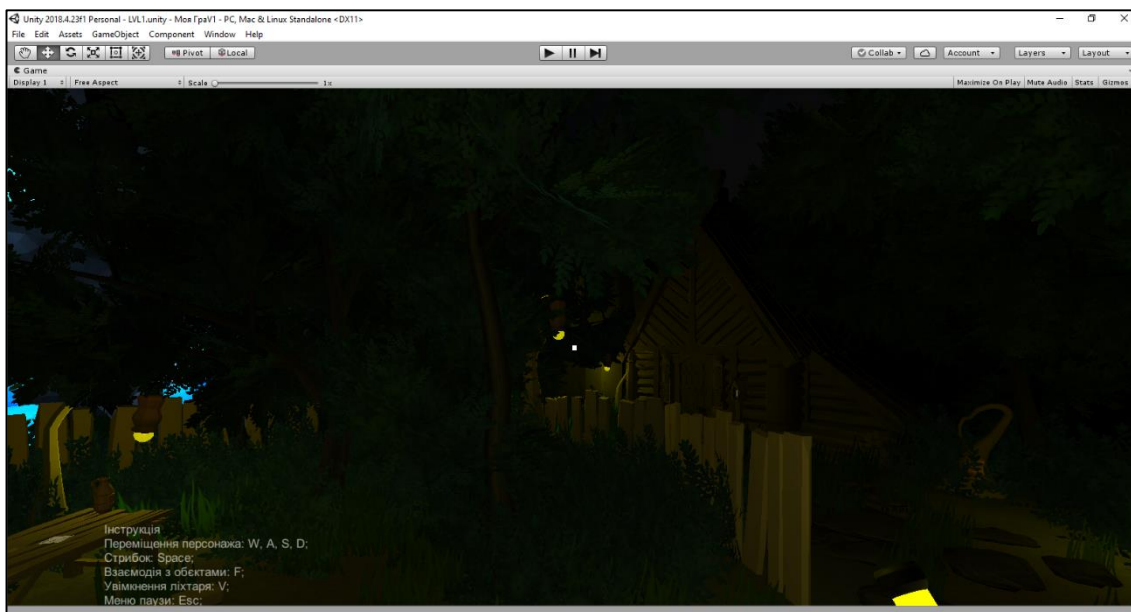


Рисунок 4.2 – Гравець на першому рівні

Якщо гравець натисне на кнопку «Settings», йому відкриється вікно, де він зможе налаштувати рівень якості відображення місцевості, чутливість мишки, клавіші керування, гучність музики та звуків у грі.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Одним із завдань було створення контролера у грі, який би дозволив пересуватися та переплигувати перешкоди. На рисунку 4.3 показано процес переплигування перешкоди.

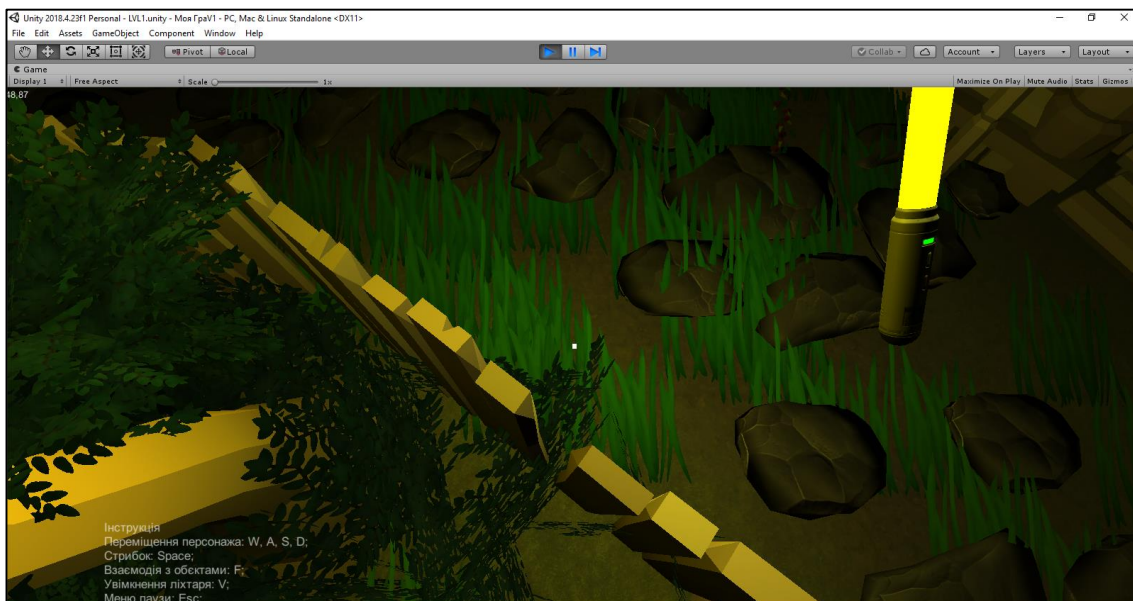


Рисунок 4.3 – Переплигування перешкоди на рівні

У грі було реалізовано якісне та комфортне освітлення для гравця. На малюнку 4.4 показано відрізок ігрового процесу.

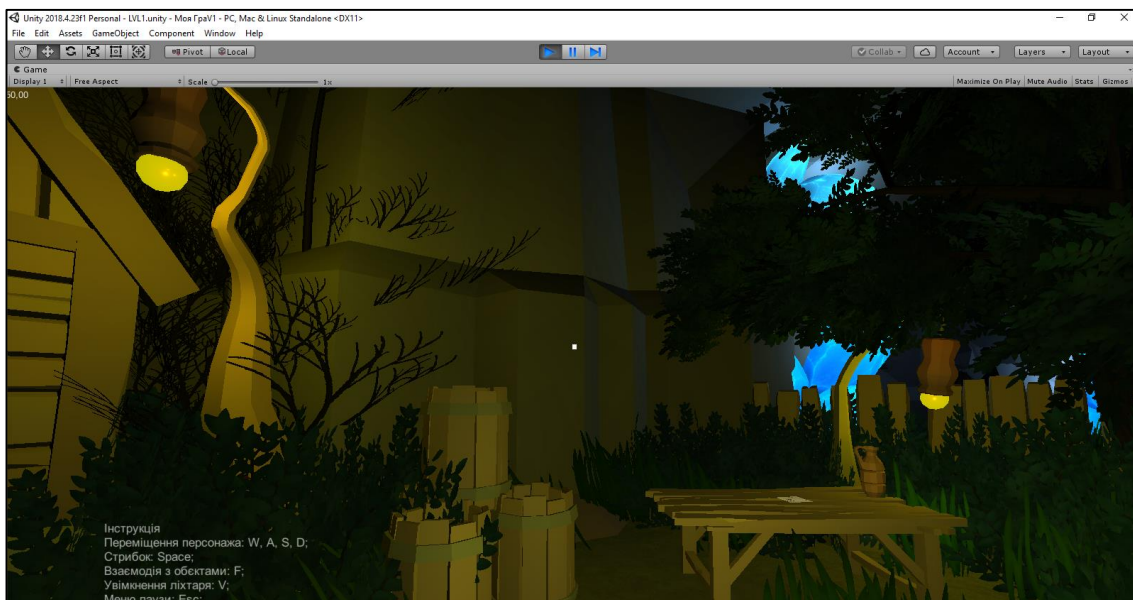


Рисунок 4.4 Відрізок ігрового процесу

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

Реалізована перешкода «Закодовані двері», яка не пропускає гравця на наступний рівень, без знайдення усіх частин паролю до них. На рисунку 4.5 показано яким реалізована перешкода в кінцевому варіанті ігрового продукту.



Рисунок 4.5 – Перешкода закодовані двері

Однією із перешкод гри був рівень «Лабіринт туману», у якому гравець потрапляє в невеличку імітацію лісу, який повністю заповнений густим туманом. Для того, щоб гравець пройшов його, потрібно орієнтуватися за зеленими яскравими кристалами, які світяться у темряві. Процес проходження рівня показано на рисунку 4.6.

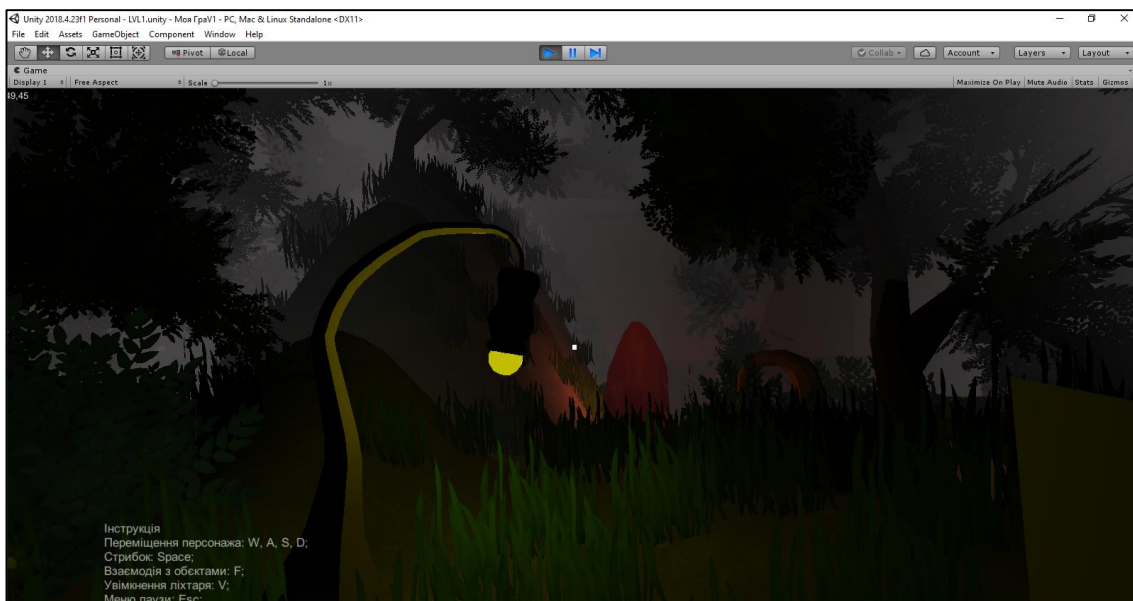


Рисунок 4.6 – Вставка з рівня «Лабіринт туману»

					ДП.КН.20.416.20.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Між рівнем «Лабіринт туману» та рівнем із знаходженням певних предметів, для відкриття проходу у фінал гри, була реалізована перешкода «Каньйон дерев», яка в свою чергу доволі унікальна, як тільки гравець підійде до цієї перешкоди, перед ним розвернеться доволі дивна картина, глибокий каньйон, з якого виростають дерева та інші рослини і прямо по центрі літаючі, невеликі платформи які світяться приємним синім світлом.

Після того, як гравець стане на одну із платформ, вона повинна впасти через деякий час, але якщо гравець не встигне переплигнути на іншу платформу, то він упаде разом із нею у каньйон, тоді гра буде завершена, через те що персонаж програв. Процес проходження рівня «Каньйон дерев» зображено на рисунку 4.7.

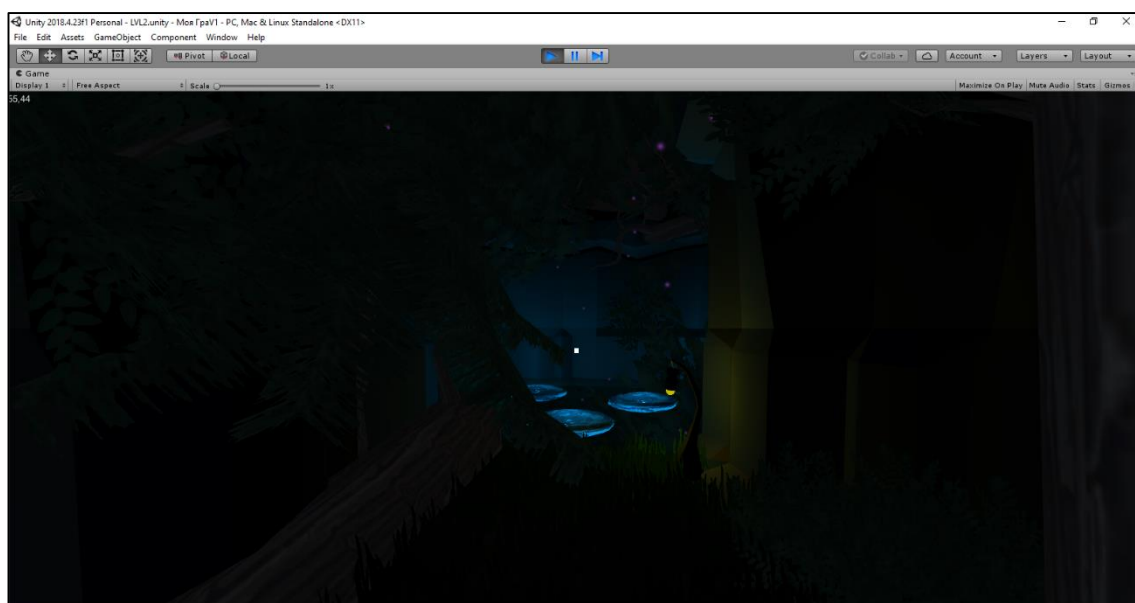


Рисунок 4.7 – Зовнішній вигляд рівня «Каньйон дерев»

Як тільки гравець пройде рівень «Каньйон дерев», він потрапить на рівень, у якому повинен знайти п'ять «артефактів» та поставити їх на платформу, щоб відкрити прохід далі.

Зовнішній вигляд рівня подібний на давні руїни, а артефакти виглядають, як невеликі зелені куби, які м'яко світяться зеленим, щоб їх можна було знайти на темному рівні. Процес збирання артефактів зображено на рисунку 4.8.

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

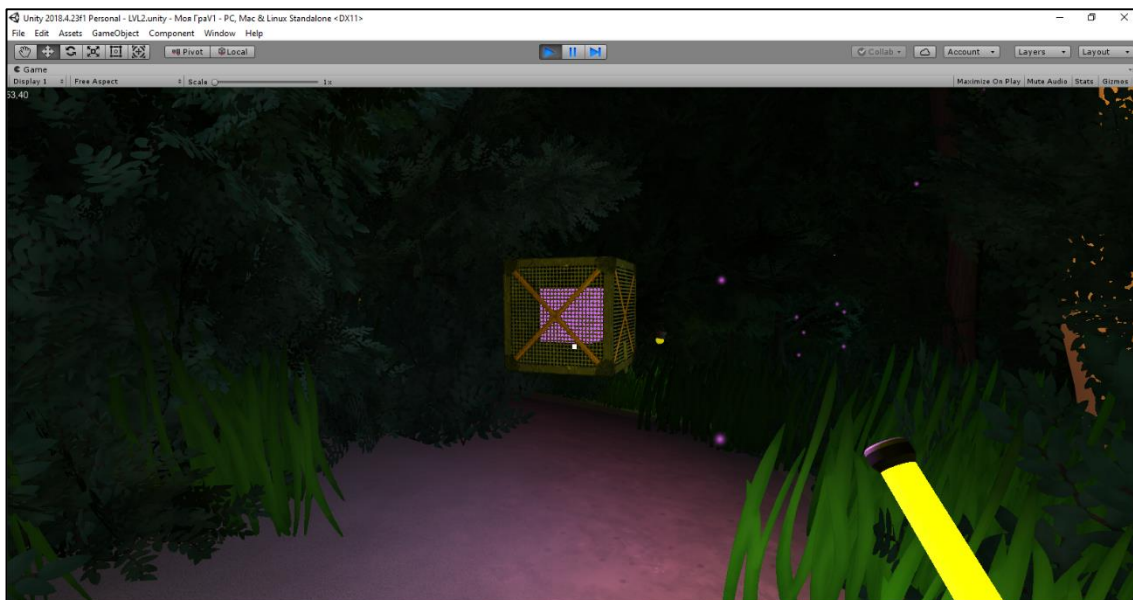


Рисунок 4.8 – Уривок з процесу збирання артефактів

Розроблена також перешкода «Puzzle», у якій потрібно зібрати із поданих гравцеві фрагментів, які розміщені на платформі, у цілісну картину, для того щоб відчинити прохід на наступний рівень. Процес проходження перешкоди «Puzzle» зображено на рисунку 4.9.

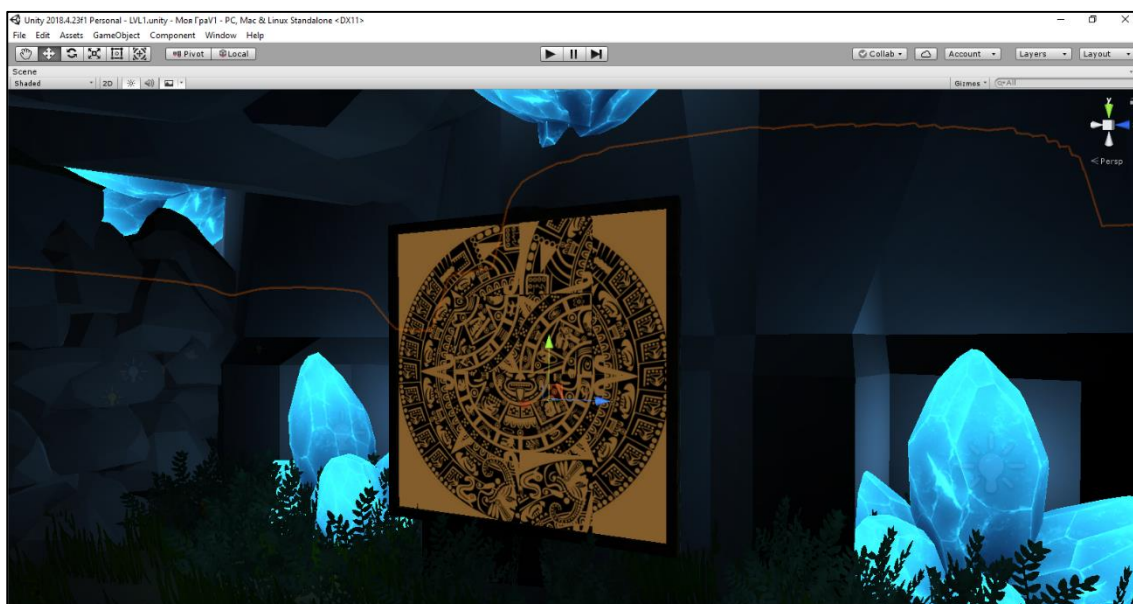


Рисунок 4.9 – Проходження перешкоди «Puzzle»

					ДП.КН.20.416.20.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

5 ТЕХНІКО – ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

Створення якісної продукції розраховує здійснення техніко – фінансових розрахунків, з завданням опису ефективних маршрутів рішення технологічних питань під час проєктної підготовки, конструювання, формування, автоматизації, обслуговування продукції. Обов'язковою частиною є впровадження знань і набуття практичних навичок в вирішення таких питань: техніко – фінансове пояснення конкретного конструкторського рішення, підрахунок витрат на проєктування.

5.1 Аналіз ринку

Створення гри та подання її покупцям, справа відповідальна і нелегка. Але до всього потрібно ставитися творчо, особливо, коли мова йде про логічну – розвиваючу гру. Чим складніше і різноманітніше те, що потрібно створити, тим більше необхідність у спрощенні.

Увага людини дуже нестійка і не завжди можна утримати інтерес до своєї гри, тому існує доволі багато методів які допомагають зацікавити покупця та буквально затягнути у світ гри.

Перший метод – це подання насиченої історії, коли гравець занурюється у історію самої гри, та її сюжет. Після того користувач буде відноситись до гри начебто він може вплинути на хід гри та відчуває себе на місті головного персонажа але це ілюзія.

Другий метод – це гемплей та всякі «фічі» які може дати гра, наприклад: керування багатотисячним військом (бойових кораблів, рицарів, або навіть потужною технікою) та спорудження різних стратегічних будівель, цей жанр часто називають «стратегія» .

Третій метод – це надзвичайно приємна графіка, вона може бути як і реалістична так і більш фентезійна. Зазвичай перше враження від гри йде саме від графіки тому непотрібно економити ресурси на її розвиток.

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

5.2 Розрахунок витрат на проектування

Оплата праці – це грошове вираження вартості та ціни робочої сили, яка виступає в ролі заробітку, виплаченого власником підприємства працівнику за виконану роботу.

Розмір заробітної плати залежить від складності та умов виконуваної роботи, кількості виконавців, професійно-ділових якостей працівника, результатів його праці.

Витрати на розробку апаратних та програмних засобів включають:

- заробітну плату розробників;
- відрахування у спеціальні державні фонди;
- накладні витрати;
- інші витрати;
- витрати на використання комп'ютерної техніки.

На стадії розробки проєкту перш за все потрібно визначити працівники якої кваліфікації потрібні для реалізації даного проєктного рішення.

Отже, до розробки даного проєктного рішення були залучені такі спеціалісти, як: проєктувальник, графічний дизайнер, програміст (таблиця 5.1).

Таблиця 5.1 – Розрахунок заробітної плати персоналу

№	Посада	Оклад	Відрахування	Кількість	Сума
1	Програміст	7500грн./р.	1462 грн./р.	1 чол.	6037,5грн.
2	Графічний дизайнер	4384грн./р.	836 грн./р.	1 чол.	7164,5грн.
3	Проєктувальник	4106грн./р.	800 грн./р.	1 чол.	14812грн.
		Усього зарплати:			28 014 грн.

Витрати підприємства на соціальні заходи виникають внаслідок нарахувань сум в залежності від розміру фактичних витрат на оплату праці

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

працівників, які включають витрати на основну і додаткову заробітну плату та здійснення інших видів заохочень та виплат.

Обов'язкові відрахування здійснюються на:

- податок на доходи фізичних осіб;
- військовий збір;
- єдиний внесок.

Програміст:

- Рахуємо податок на доходи фізичних осіб: $7500 \times 18\%$ (ставка податку на доходи фізичних осіб) = 1350 грн;
- Рахуємо військовий збір: $7500 \times 1,5\%$ (ставка військового збору) = 112,5 грн;
- Рахуємо єдиний внесок: $7500 \times 22\%$ (ставка ЄСВ) = 1650 грн;
- Утримання становлять – 1462,5 грн (1350 грн. + 112,5 грн.);
- До виплати працівникові – 6037,5 грн ($1 \times (7500 \text{ грн.} - 1462,5 \text{ грн.})$).

Графічний дизайнер:

- Рахуємо податок на доходи фізичних осіб: $8900 \times 18\%$ (ставка податку на доходи фізичних осіб) = 1602 грн;
- Рахуємо військовий збір: $8900 \times 1,5\%$ (ставка військового збору) = 133,5 грн;
- Рахуємо єдиний внесок: $8900 \times 22\%$ (ставка ЄСВ) = 1958 грн;
- Утримання – 1735,5 грн (1602 грн. + 133,5 грн.);
- До виплати працівникові – 7164,5 грн ($8900 \text{ грн.} - 1735,5 \text{ грн.}$).

Проектувальник:

- Рахуємо податок на доходи фізичних осіб: $9200 \times 18\%$ (ставка податку на доходи фізичних осіб) = 1656 грн;
- Рахуємо військовий збір: $9200 \times 1,5\%$ (ставка військового збору) = 138 грн;
- Рахуємо єдиний внесок: $9200 \times 22\%$ (ставка ЄСВ) = 2024 грн;
- Утримання – 1794 грн (1656 грн. + 138 грн.);
- До виплати працівникові – 14812 грн ($2 \times (9200 \text{ грн.} - 1794 \text{ грн.})$).

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

Відрахування на соціальні потреби становлять – 7656 грн.
 $((1650+1958+(2024*2))$

Таблиця 5.2 – Кошторис витрат на проектування

Найменування статей витрат	Сума, грн
1 Зарплата працівників.	2875
2. Обов'язкові відрахування	28014
3. Відрахування на соціальні потреби.	7656
4. Розрахунки матеріальних витрат	199
5. Витрати на використання комп'ютерної техніки.	804
6. Витрати на відлагодження і дослідну експлуатацію програмного продукту	673
7. Накладні витрати експлуатацію	2123
8.Усього, кошторисна вартість проєкту.	21650

5.3 Обґрунтування необхідності розробки

Ігри та подібні продукти і заходи такі як кіберспортивні змагання, виставка нових відеоігор, фестивалі на різну тематику та інші, приносять величезні прибутки і мають колосальний вплив на інші види для проведення часу. Глобальний ринок відеоігор розростається з кожним роком, в 2018 році його оцінили від \$ 57 млрд до \$ 78700000000. Число гравців по всьому світу склала 1800000000 людей, найбільшу кількість гравців була зареєстрована в країнах Азії.

Найбільшими регіональними ринками на 2019 були:

- північноамериканський (\$ 23600000000);
- азіатський (\$ 23100000000);
- європейський (\$ 22 млрд);
- американський (\$ 4500 млн).

Найбільші прибутки отримали:

- мобільні ігри (\$ 22300000000);

					ДП.КН.20.416.20.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

- роздрібні продажі (\$ 19700000000);
- free-to-play MMO гри (\$ 8700 млн);
- ігри в соціальних мережах (\$ 7900 млн);
- завантажуються DLS до вже випущених відеоігор (\$ 7500 млн);
- консолі (\$ 3100 млн);
- pay-to-play MMO (\$ 2700 млн);
- платні відео (\$ 15000000000);
- кіберспортивні змагання (\$ 612 млн);
- VR технології (\$ 225 млн).

Наразі щоб почати розробку відеоігор потрібно дуже багато вкладати грошей в сам проєкт, але не факт, що він окупить себе, тому потрібно створювати щось оригінальне та розкручене рекламою, щоб заохотити інвесторів які допоможуть фінансувати проєкт.

В ринок відеоігор для ПК доволі важко втиснутися, тому що на ньому є дуже багато конкурентів які не хочуть поступатися місцем для інших малих та середніх компаній, але на ринку мобільних відеоігор ситуація зовсім інша, багато місць для малих та середніх компаній, які вже закріпилися там і отримують хоча і малий але стабільний дохід, це і є причиною того, що гіганти індустрії не звертають увагу на нібито "неприбуткову" мобільну платформу.

Переважно в малі компанії вступають інді розробники, які працювали над своїми успішними проєктами, та мають хороші навички в роботі та ентузіазм щоб розвиватися далі та збільшувати свої знання. Такі компанії розробляють креативні та сміливі проєкти, які використовуються не для заробітку, а для задоволення потреб користувачів та покращення свого рейтингу у топах компаній.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Метою дипломного проєкту є створення логічно–розвиваючої гри. Для досягнення поставленої мети було досліджено предметну область, проаналізовано існуючі популярні ігри даного жанру, розглянуто інструментальні засоби для реалізації комп'ютерних ігор даного типу, розроблено сценарій власної гри та спроектовано і розроблено необхідні компоненти гри.

Під час роботи над дипломним проєктом було вдосконалено та застосовано такі практичні навички: створення полігональних 3D моделей за допомогою програми Blender, створення Voxel 3D моделей з використанням засобів програми MagicVoxel, продумане використання освітлення на місцевості, грамотне розташування об'єктів на локації, створення програмного коду продукту, тонкості використання стандартних та додаткових інструментів у рушії Unity, продуманий дизайн рівнів та оптимізація проєкту. Загалом, в цій роботі було використано велика кількість 3D моделей для створення навколишнього середовища, цікавих локацій, для того щоб користувач під час гри відчув себе на місці головного героя.

Щоб гра правильно функціонувала на комп'ютерах різної потужності, було використано тільки примітивні, низькополігональні моделі стін, будівель та декоративних об'єктів. Для побудування красивих та продуманих рівнів використовувалися малюнки різних художників та незвичайні фотографії. При побудові рівня враховувалися розміри персонажа, щоб оточуючі об'єкти не були занадто великими або навпаки, дуже маленькими.

Підводячи підсумки по розробці проєкту можна сказати, що це була хороша практика у застосуванні знань та навичок, які були здобуті під час навчання і дуже пригодилися на всіх стадіях розробки проєкту, від чіткого планування концепції гри, і до написання якісного коду для функціонування усіх підсистем, які є у грі.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Під час виконання дипломного проєкту було спроектовано і розроблено ігровий програмний продукт у жанрі головоломка (логічна–розвиваюча гра, який забезпечує виконання всіх необхідних функцій та відповідає поставленим вимогам. Програмний продукт готовий до використання, в подальшому планується розширення функціоналу, збільшення кількості рівнів гри та локацій і розповсюдження безкоштовно на тематичних сайтах.

					ДП.КН.20.416.20.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Введение в Visual Studio. Что такое Visual Studio. Начало работы. *Microsoft.com*: веб-сайт. URL: <https://visualstudio.microsoft.com/ru/vs/> (дата звернення: 10.02.2020).
2. Введение в геймдизайн: Основные понятия и принципы проектирования игр. *Vc.ru*: веб-сайт. URL: <https://vc.ru/flood/10495-gamedev-challenges> (дата звернення: 04.02.2020).
3. Инструкция начинающего разработчика игр. *Habr.com*: веб-сайт. URL: <https://habr.com/ru/post/153933/> (дата звернення: 04.02.2020).
4. Начните воплощать ваши идеи сегодня с Unity — платформой разработки 3D-приложений в реальном времени. *Unity*: веб-сайт. URL: <https://unity.com/ru/products/core-platform> (дата звернення: 15.02.2020).
5. Начало работы с Unity. *Unity3D*: веб-сайт. URL: <https://docs.unity3d.com/ru/current/Manual/UnityBasics.html> (дата звернення: 11.02.2020).
6. Приемы при проектировании архитектуры игр. *Habr.com*: веб-сайт. URL: <https://habr.com/ru/post/255561/> (дата звернення: 04.02.2020).
7. Создание 3D с помощью Blender. *Blender*: веб-сайт. URL: <https://www.blender.org/> (дата звернення: 02.02.2020).
8. Blender 2.82 Руководство Пользователя. *Blender*: веб-сайт. URL: <https://docs.blender.org/manual/ru/dev/> (дата звернення: 02.02.2020).
9. MagicaVoxel free lightweight 8-bit voxel art editor. *MagicaVoxel*: веб-сайт. URL: https://ephtracy.github.io/index.html?page=mv_main (дата звернення: 02.02.2020).

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

ДОДАТКИ

Додаток А

Файл Cameracontrols.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Cameracontrols : MonoBehaviour
{ public float updateInterval = 0.5F;
  private double lastInterval;
  private int frames = 0;
  private float fps;

  public float mouseSensitivity = 100f;
  public Transform playerBody;
  float xRotation = 0f;

  void Start()
  {

    Cursor.lockState = CursorLockMode.Locked;
    lastInterval = Time.realtimeSinceStartup;
    frames = 0;

  }

  void OnGUI()
  {
    GUILayout.Label("" + fps.ToString("f2"));
  }

  void FixedUpdate()
  {
    /*FPS*/
    ++frames;
    float timeNow = Time.realtimeSinceStartup;
    if (timeNow > lastInterval + updateInterval)
    {
      fps = (float)(frames / (timeNow - lastInterval));
      frames = 0;
      lastInterval = timeNow;
    }
    /*FPS*/

    /*moveCam*/
    float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
    float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;

    xRotation -= mouseY;
    xRotation = Mathf.Clamp(xRotation, -90f, 90f);

    transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
    playerBody.Rotate(Vector3.up * mouseX);
    /*moveCam*/

  }
```

					ДП.КН.20.416.20.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

Файл Died.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class Died : MonoBehaviour
{
    [SerializeField]
    private GameObject YouDead;
    [SerializeField]
    private GameObject FPS;
    public string LVL;
    void Start()
    {
        Time.timeScale = 1;
        YouDead.SetActive(false);
        FPS.SetActive(true);
    }
    void OnTriggerEnter(Collider obj)
    {
        if (obj.tag == "Player")
        {
            StartCoroutine(ExampleCoroutine());
        }
    }
    IEnumerator ExampleCoroutine()
    {
        yield return new WaitForSeconds(1);
        FPS.SetActive(false);
        YouDead.SetActive(true);
        yield return new WaitForSeconds(4);
        SceneManager.LoadScene(LVL);
    }
}
```

Файл Grab.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Grab : MonoBehaviour
{
    public Transform theDest;
    private void OnMouseDown()
    {
        GetComponent<BoxCollider>().enabled = false;
        GetComponent<Rigidbody>().useGravity = false;
        this.transform.position = theDest.position;
        this.transform.parent = GameObject.Find("Capsule").transform;
    }
    private void OnMouseUp()
    {
        this.transform.parent = null;
        GetComponent<Rigidbody>().useGravity = true;
        GetComponent<BoxCollider>().enabled = true; }
}
```

					ДП.КН.20.416.20.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

Файл Move.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Move : MonoBehaviour
{
    public CharacterController controller;
    public float speed = 12f;
    public float gravity = -9.81f;
    public float jumpHeight = 3f;
    public Transform groundCheck;
    public float groundDistance = 0.4f;
    public LayerMask groundMask;
    Vector3 velocity;
    bool isGrounded;
    // Update is called once per frame
    void FixedUpdate()
    {
        isGrounded = Physics.CheckSphere(groundCheck.position, groundDistance, groundMask);
        if(isGrounded && velocity.y < 0)
        {velocity.y = -2f;}
        float x = Input.GetAxis("Horizontal");
        float z = Input.GetAxis("Vertical");
        Vector3 move = transform.right * x + transform.forward * z;
        controller.Move(move * speed * Time.deltaTime);
        if(Input.GetButtonDown("Jump") && isGrounded)
        {velocity.y = Mathf.Sqrt(jumpHeight * -2f * gravity);}
        velocity.y += gravity * Time.deltaTime;
        controller.Move(velocity * Time.deltaTime); }
    }
```

Файл MovePlatform.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MovePlatforme : MonoBehaviour
{
    private Vector3 a;

    public void Start()
    {
        a = this.transform.position;
        GetComponent<Rigidbody>().useGravity = false;
    }
    private void OnTriggerStay(Collider obj)
    {if (obj.tag == "Player")
        { StartCoroutine(ExampleCoroutine());}}
    IEnumerator ExampleCoroutine()
    { yield return new WaitForSeconds(Random.Range(1f,2f));
        GetComponent<Rigidbody>().useGravity = true;
        yield return new WaitForSeconds(4);
        GetComponent<Rigidbody>().useGravity = false;
        GetComponent<Rigidbody>().Sleep();
        yield return new WaitForSeconds(12);
        this.gameObject.SetActive(false);
        this.transform.position = a;
        this.gameObject.SetActive(true);
    }
}
```

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

Файл Password_Start.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class Password_Start : MonoBehaviour
{
    public string pass;
    public GameObject item_1;
    public GameObject item_2;
    public GameObject boom;
    public GameObject panel;
    public InputField UserInput;
    public GameObject ok;
    public GameObject no;
    private string User;
    void Start()
    {
        item_1.SetActive(true);
        boom.SetActive(true);
        item_2.SetActive(false);
        panel.SetActive(true);
        ok.SetActive(false);
        no.SetActive(false);
    }
    public void Enter()
    {
        User = UserInput.text;
        if (User != null)
        {
            if (pass == User )
            {
                StartCoroutine(ExampleCoroutine());
            }
            else
            {
                {ok.SetActive(false);
                no.SetActive(true); } }
            UserInput.text = null; }
    public void Exit()
    {panel.SetActive(false);
    Cursor.lockState = CursorLockMode.Locked;
    Time.timeScale = 1f; }
    IEnumerator ExampleCoroutine()
    { no.SetActive(false);
    ok.SetActive(true);
    Time.timeScale = 1f;
    Cursor.lockState = CursorLockMode.Locked;
    panel.transform.position = panel.transform.position - new Vector3(0, -1000, 0);
    yield return new WaitForSeconds(1);
    item_1.SetActive(false);
    boom.SetActive(false);
    item_2.SetActive(true);
    yield return new WaitForSeconds(6);

    item_2.SetActive(false);
    panel.SetActive(false);

    }
}

```

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Файл PauseMenu.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class PauseMenu : MonoBehaviour
{
    public GameObject Flash;
    private bool on;
    public GameObject pauseMenu;
    public GameObject panel;
    public bool isPaused;
    // Start is called before the first frame update
    void Start()
    {
        isPaused = false;
        pauseMenu.SetActive(false);
        on = false;
        Flash.SetActive(false);
    }
    // Update is called once per frame
    void Update()
    {
        if(Input.GetKeyDown(KeyCode.Escape))
        {
            if(isPaused)
            {
                ResumeGame();
            } else
            {
                panel.SetActive(false);
                isPaused = true;
                pauseMenu.SetActive(true);
                Cursor.lockState = CursorLockMode.None;
                Time.timeScale = 0f;
            }
        }
        if (Input.GetKeyDown(KeyCode.V))
        {
            on = !on;
            if (on == true)
            {
                Flash.SetActive(false);
            }
            if (on == false)
            {
                Flash.SetActive(true);
            }
        }
    }
    public void ResumeGame()
    {
        isPaused = false;
        pauseMenu.SetActive(false);
        Cursor.lockState = CursorLockMode.Locked;
        Time.timeScale = 1f;
    }
    public void Restart()
    {
        SceneManager.LoadScene("LVL1");
    }
    public void Exit()
    {
        SceneManager.LoadScene("Menu");
    }
}
```

					ДП.КН.20.416.20.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

Файл PressF.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class PressF : MonoBehaviour
{ [SerializeField]
  private GameObject Text;
  [SerializeField]
  private GameObject Password;
  // Start is called before the first frame update
  void Start()
  { Text.SetActive(false);
    Password.SetActive(false); }
  private void OnTriggerStay(Collider obj)
  { if (obj.tag == "Player")
    { Text.SetActive(true);
      if (Input.GetKeyDown(KeyCode.F))
      { Password.SetActive(true);
        Text.SetActive(false);
        Cursor.lockState = CursorLockMode.None;
        Time.timeScale = 0f; } } }
  private void OnTriggerExit(Collider obj)
  { if (obj.tag == "Player")
    { Text.SetActive(false); } } }
```

Файл GameControl.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class GameControl : MonoBehaviour
{ [SerializeField]
  private Transform[] pictures;
  [SerializeField]
  private GameObject zaslonka;
  [SerializeField]
  private GameObject objec1;
  public static bool youWin;
  void Start()
  { zaslonka.SetActive(true);
    objec1.SetActive(false);
    youWin = false; }
  void FixedUpdate()
  { if (pictures[0].rotation.z == 0 &&
    pictures[1].rotation.z == 0 &&
    pictures[2].rotation.z == 0 &&
    pictures[3].rotation.z == 0 &&
    pictures[4].rotation.z == 0 &&
    pictures[5].rotation.z == 0 &&
    pictures[6].rotation.z == 0 &&
    pictures[7].rotation.z == 0 &&
    pictures[8].rotation.z == 0 &&
    pictures[9].rotation.z == 0 &&
    pictures[10].rotation.z == 0 &&
    pictures[11].rotation.z == 0 &&
    pictures[12].rotation.z == 0 &&
    pictures[13].rotation.z == 0 &&
    pictures[14].rotation.z == 0 &&
    pictures[15].rotation.z == 0)
    { youWin = true;
      zaslonka.SetActive(false);
      objec1.SetActive(true);
    } } }
```

					ДП.КН.20.416.20.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Б

Таблиця Б.1 – Клавiшi управлiння у грі

Опис	Клавiшi
Рух гравця вперед	W
Рух гравця назад	S
Рух гравця на лiво	A
Рух гравця на право	D
Стрибок	Space
Пауза	Esc
Взаємодiя з об'єктами	F
Вихiд iз записок	E
Увiмкнення додаткового освiтлення (лiхтар)	V
Пiдняття об'єктiв	ЛКМ

РЕЦЕНЗІЯ
на дипломний проект
студента відділення комп'ютерних та видавничих технологій
Галицького коледжу імені В'ячеслава Чорновола

студента IV курсу групи К-47

Лісного Олега Володимировича
(прізвище та ініціали)

Спеціальність 122 „Комп'ютерні науки та інформаційні технології”

Обсяг дипломного проекту: 59 стор.

Кількість сторінок записки: 52 стор.

Тема: логічно-розвиваюча гра

1. Актуальність теми: розробка гри-головоломки з
можливістю вибору рівня складності та
локації

2. Практична або теоретична цінність опрацьованих питань
досліджено предметну область проаналізованих
інструментальних засобів реалізації, розроблено
цікавий сценарій гри та необхідні
компоненти гри.

3. Недоліки роботи суттєвих недоліків не виявлено
використання лише інтернет-джерел в якості
інформаційних джерел. Відсутність посилань
на наукові джерела.

4. Загальний висновок студент продемонстрував
високі програмістські навички, а
робота заслуговує на високу оцінку

Рецензент Глинка Марина Любомирівна
(прізвище та ініціали рецензента)

«25» червня 2020 р.

elv
(підпис)

ВІДГУК
на дипломний проект
студента відділення комп'ютерних та видавничих технологій
Галицького коледжу імені В'ячеслава Чорновола

студента IV курсу групи К-47

Лісний Олега

(прізвище та ініціали)

Спеціальність 122 „Комп'ютерні науки та інформаційні технології”

Керівник ДП Курчиченко Н.З.

Тема: логічно-розвиваюча гра

1. Загальна характеристика студента Власно виконує поставлені завдання, вміє аналізувати інформаційні джерела, проектувати ПЗ, розробляти ПЗ з використанням сучасних засобів та нові програмування

2. Практична або теоретична цінність опрацьованих питань Детально проаналізовано сферу комп'ютерних ігор, зокрема, головоломок, проект може застосовуватись на практиці, містить логічно-розвиваючі завд. для дітей шкільного віку.

3. Недоліки роботи

незначні недоліки в оформленні пояснювальної записки

4. Загальний висновок

Студент реалізував проект відповідно до поставлених вимог, додаток готовий до практичного використання.

Керівник дипломного проекту

Курчиченко Н.З.

(прізвище та ініціали)

25.06.2020р.

6