

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)
на тему:

«Розробка графіки та меню для логічної гри «What are
the ghosts talking about?»»

Виконав студент групи КН-19-1
МЕНЕС Юлія Володимирівна

Керівник роботи:
МАРЧУК Галина Вікторівна

Рецензент:
ЛЕВКІВСЬКИЙ Віталій Леонідович

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

Марина ГРАФ

«05» січня 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)

Здобувач вищої освіти: МЕНЕС Юлія Володимирівна

Керівник роботи: МАРЧУК Галина Вікторівна

Тема роботи: Розробка графіки та меню для логічної гри «What are the ghosts talking about?»

Затверджена Наказом університету від «05» січня 2023 р. № 10/с

Термін здачі студентом закінченої роботи: «10» червня 2023 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження): використання Adobe Illustrator та Adobe Photoshop для створення графічних матеріалів, а також рушія Unity для створення панелей меню для 2D логічної гри.

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв
1	Галина МАРЧУК	01.09.2022	01.11.2022
2	Галина МАРЧУК	01.11.2022	01.04.2023
3	Галина МАРЧУК	01.04.2023	01.05.2023

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формулювання завдання. Опрацювання літературних джерел	25.09.22 – 10.11.22	Виконав
2	Проектування структури за допомогою діаграм UML	11.11.22 – 20.11.22	Виконав
3	Написання програмного коду та створення графічних матеріалів	21.11.22 – 20.03.23	Виконав
4	Тестування та оформлення пояснювальної записки	21.03.23 – 25.04.23	Виконав
5	Попередній захист	10.06.2023	Виконав

Здобувач вищої освіти Юлія МЕНЕС
Керівник Галина МАРЧУК

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра на тему "Розробка графіки та меню для логічної гри "What are the ghosts talking about?" складається з пояснювальної записки, графічних матеріалів та програмного коду. Пояснювальна записка містить 57 сторінок, 62 ілюстрації та 1 таблицю.

У роботі описано процес розробки графіки та меню гри, включаючи вибір кольорової палітри та створення елементів графіки, таких як персонажі, об'єкти та фони. Також проаналізовано різні методики створення графіки та їх переваги та недоліки.

У роботі представлено опис створення меню гри, включаючи розміщення кнопок та інтерфейсу користувача.

Крім того, у роботі наведено результати тестування створеної графіки та меню для визначення недоліків та їх виправлення.

КЛЮЧОВІ СЛОВА: ГРАФІКА, МЕНЮ, ЛОГІЧНА ГРА, ADOBE PHOTOSHOP, ADOBE ILLUSTRATOR.

КН.КР.Б – 122 – 23 – ПЗ

					КН.КР.Б – 122 – 23 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка графіки та меню для логічної гри «What are the ghosts talking about?» Пояснювальна записка	Літ.	Арк.	Аркуші
Розроб.		Ю.В. МЕНЕС					4	67
Керівник		Г.В. МАРЧУК				Житомирська політехніка, група КН-19-1		
Реценз.		В.Л. ЛЕВКІВСЬКИЙ						
Н. Контр.								
Затверд.		М.С. ГРАФ						

ABSTRACT

Bachelor's degree graduation qualification work on the topic "Development of graphics and menu for the logical game 'What are the ghosts talking about?'" consists of an explanatory note, graphic materials, and program code. The explanatory note contains 57 pages, 62 illustrations, and 1 table.

The work describes the process of developing graphics and the game menu, including the selection of a color palette and the creation of graphic elements such as characters, objects, and backgrounds. Various methodologies for creating graphics are analyzed, along with their advantages and disadvantages.

The work presents a description of the game menu creation, including the placement of buttons and user interface design.

Additionally, the work provides the results of testing the created graphics and menu to identify shortcomings and their rectification.

KEYWORDS: GRAPHICS, MENU, LOGICAL GAME, ADOBE PHOTOSHOP, ADOBE ILLUSTRATOR.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ .	9
1.1. Постановка задачі.....	9
1.2. Аналіз аналогів програмного продукту	10
1.3. Обґрунтування вибору інструментальних засобів для створення графічних елементів.....	13
1.4. Пайплайн.....	15
Висновки до першого розділу	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ ПАНЕЛЕЙ МЕНЮ ТА РОЗРОБКА ГРАФІЧНИХ МАТЕРІАЛІВ.....	20
2.1. Проектування панелей меню за допомогою діаграм UML	20
2.2. Основні етапи реалізації панелей головного меню.....	25
2.3. Створення графічних матеріалів для панелей меню	29
2.4. Створення персонажів за допомогою Adobe Photoshop	32
2.5. Створення катсцен за допомогою Adobe Photoshop	35
2.6. Створення текстового документу з підказками	38
Висновки до другого розділу.....	39
РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ ..	41
3.1. Інтерфейс та порядок роботи з програмним додатком.....	41
3.2. Тестування роботи програмного забезпечення	50
Висновки до третього розділу	54
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
ДОДАТКИ.....	58
Додаток А	59

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

UML (Unified Modeling Language) - уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування

UI – User Interface – інтерфейс, елементи, з якими взаємодіє користувач

NPC - Non-Player Character – персонажі, якими гравець не управляє, та з якими зазвичай можна взаємодіяти.

ПЗ – програмне забезпечення

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Комп'ютерні ігри є важливим аспектом нашого сучасного життя. Вони стали відомим видом розваги, який привертає увагу не лише молоді, але і дорослих. Ігри стали окремим видом мистецтва, що включає в себе елементи техніки, дизайну, музики та кіно.

Логічні ігри є однією з найбільш популярних категорій комп'ютерних ігор. Це ігри, в яких гравець повинен розв'язувати різноманітні головоломки, логічні завдання та проблеми. Ці ігри сприяють розвитку різних когнітивних процесів, таких як логічне мислення, увага, концентрація, пам'ять та інші.

Тому створення логічних ігор є важливим напрямом сучасного програмування, що забезпечує розвиток як технічного, так і художнього боку інформаційних технологій. Для розробки ігор необхідні професійні знання та навички в галузі програмування, дизайну, анімації, звуку та іншого.

Також важливо звернути увагу на дизайн та інтерфейс логічних ігор, які повинні бути цікавими та зрозумілими для гравців. У цьому контексті розробка графіки та меню для логічної гри є важливим завданням, що потребує професійного підходу та знань у галузі дизайну та інтерфейсу.

Крім того, логічні ігри можуть бути використані для навчання та навчальних цілей. Це дозволяє зробити процес навчання більш цікавим та залучити до нього учнів різного віку.

Таким чином, логічні ігри є важливим елементом сучасного життя, що можуть не тільки забавляти нас, але й сприяти розвитку різних когнітивних процесів та використовуватися для навчання та навчальних цілей. Важливо звернути увагу на якість дизайну та інтерфейсу логічних ігор, щоб забезпечити цікавість та зрозумілість для гравців.

Метою кваліфікаційної роботи є розробка графіки для логічної гри, використовуючи такі графічні редактори як: Adobe Photoshop та Adobe Illustrator, а також розробка меню, використовуючи рушій Unity.

Встановлена мета обумовлює наступні завдання:

– проектування та розробка графічних елементів та інтерфейсу користувача гри;

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

- визначення стилістики та дизайну гри, що відповідає основним концепціям та тематиці гри;
- тестування та оптимізація розробленого програмного забезпечення для забезпечення максимальної продуктивності та якості гри.

Об'єктом дослідження є розробка графіки та меню для логічної гри "What are the ghosts talking about ?", зокрема процес проектування та створення графічних елементів гри та їх інтеграція в єдину графічну концепцію, а також розробка та імплементація функціональності інтерфейсу гри.

Предметом дослідження є використання рушія Unity для розробки меню, а також графічних редакторів: Adobe Photoshop та Adobe Illustrator.

В рамках теми кваліфікаційної роботи було опубліковано тези:

Менес Ю. В., Марчук Д.К., 2D ГРАФІКА В ІГРАХ. XIII Міжнародна науково-технічна конференція. Інформаційні комп'ютерні технології - 2023. 30-31 березня 2023 р. м. Житомир.

Менес Ю.В., Марчук Г.В. Наратив в комп'ютерних іграх. Тези Всеукраїнської науково-практичної on-line конференції здобувачів вищої освіти і молодих учених, присвяченої Дню науки 11-15 травня 2021 року. Житомир: «Житомирська політехніка», 2021. С.143.

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1. Постановка задачі

Створення якісної логічної гри потребує розробки ефективного інтерфейсу користувача та графічної складової.

Метою кваліфікаційної роботи є розробка меню та графіки для логічної гри, що буде забезпечувати комфортну гру для користувачів та забезпечувати високий рівень її популярності.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Спроекувати меню за допомогою діаграм UML.
 - 1.1. Створити діаграму варіантів використання.
 - 1.2. Створити діаграму активності.
 - 1.3. Створити діаграму класів.
2. Розробити код меню, орієнтуючись на створені раніше діаграми UML.
3. Розробити дизайн різних панелей меню.
 - 3.1. Головне меню.
 - 3.2. Панель налаштувань.
 - 3.3. Панель «Продовжити гру».
 - 3.4. Панель «Нова гра».
 - 3.5. Панель паузи під час гри.
4. Розробити дизайн усіх локацій (кімнат). Створити відповідний фон для кожної з них, а також додати необхідні предмети, що характеризують кімнату та обов'язкові предмети, з якими можна взаємодіяти (розмістити їх на окремих шарах в графічних редакторах). А також розробити дизайн усіх головоломок.
 - 4.1.«Ванна кімната» (головоломки: «Труби», «П'ятнашки», «Качки»).
 - 4.2.«Хол» (головоломки: «Ялинка», «Новорічний подарунок, «Метелик» та «Скелет»).
 - 4.3.«Кухня» (головоломка «Посуд»).

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

4.4.«Атракціони» (головоломки: «Лабіринт», «Тетріс», «Відгамцелити бобра», «Танці»).

4.5.«Музична кімната» (головоломки: «Музична скринька», «Музичний програвач»).

4.6.«Комора» (головоломки: «Врівноваження вагів», «Кількість води в пляшках», «Пробірки з кольором»).

4.7.«Кімната з телескопом» (головоломки: «Зоряне небо», «Книги», «Пазли»).

4.8.«Кімната зі світлом» (головоломки: «Світлячок», «Світильники»).

5. Створити текстовий документ з підказками, які будуть використовуватись гравцем.

1.2. Аналіз аналогів програмного продукту

Для аналізу було обрано ігри, спираючись, насамперед, на їх графіку.

Ось декілька з них:

<https://store.steampowered.com/app/711660/CHUCHEL/?> - «Chuchel» -

рис.1.1 - це гра в жанрі головоломки від студії Amanita Design. Головним героєм є рудуватий комічний монстр на ім'я Chuchel, який намагається відновити свою цінну волосинку, яка була вкрадена.

Переваги:

- унікальний стиль: гра відрізняється незвичайним мультяшним візуальним стилем, який використовує веселі, яскраві кольори та непередбачувані анімаційні рухи персонажів;
- деталізація та виразність: графіка в "Chuchel" відзначається великою увагою до деталей, створенням унікальних локацій, аксесуарів та різноманітних ситуацій, що додає гумору та живості грі;
- комічність: графічні елементи відповідають гумористичному тону гри, створюючи смішні та незабутні моменти для гравців.

Недоліки:

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

- спрощена стилізація: хоча спрощена графіка додає унікальності, вона може виглядати менш деталізованою порівняно з деякими іншими сучасними іграми;
- обмеженість середовища: оскільки гра пропонує обмежений набір локацій та персонажів, може виникнути відчуття повторюваності під час геймплею;



Рисунок 1.1 - Chuchel

<https://store.steampowered.com/app/956030/Creaks/?> - «Creaks» - рис.1.2 - це гра в жанрі головоломки-платформера від студії Amanita Design. Гравець вирушає у подорож в світ, де таємничі істоти, що живуть у стінах, називаються "Creaks". Головною механікою гри є розв'язування головоломок та уникання ворогів.

Переваги:

- унікальний художній стиль: гра відрізняється незвичайним та особливим художнім стилем, який поєднує мінімалістичність та фантазійність. Вона пропонує візуально захоплюючий світ з детально прорисованими локаціями та креативними дизайнами персонажів;
- атмосфера та настрій: графіка створює особливу атмосферу та глибину, допомагаючи погрузитися в таємничий світ "Creaks". Вона створює напруженість та додає до гри загадковості;

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

- анімація та ефекти: "Creaks" має вражаючу анімацію та спеціальні ефекти, які доповнюють геймплей та роблять його живим.

Недоліки:

- мінімалізм: хоча мінімалістичний стиль гри може бути привабливим для багатьох гравців, деяким користувачам він може здаватися занадто простим або недостатньо деталізованим;
- однотонність: зважаючи на деякі графічні вибори, певні локації в грі можуть виглядати дещо однотонно або повторюватися.



Рисунок 1.2 - Creaks

https://store.steampowered.com/app/714120/Little_Misfortune/ - «Little Misfortune» - рис.1.3 - це пригодницька гра з елементами хоррору та чорного гумору, розроблена студією Killmonday Games. Головною героїнею є дев'ятирічна дівчинка на ім'я Місфорчун, яка вирушає у пригоду, щоб знайти Вічне щастя для своєї матері.

Переваги:

- унікальний художній стиль: графіка в "Little Misfortune" відрізняється особливим кресленням та арт-стилем, який поєднує мінімалістичність з деталізованими та емоційними персонажами, локаціями;
- контрастність: графічний стиль створює контраст між яскравими, привабливими кольорами та темними, хоррор-елементами, що додає загадковості до гри;

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

- емоційність: графіка допомагає передати емоційну глибину історії та персонажів, відтворюючи їхні почуття та страждання;
- Недоліки:
- мінімалістичність: для деяких гравців мінімалістичний стиль графіки може здаватися занадто простим або недостатньо деталізованим;



Рисунок 1.3 - Little Misfortune

Отже, проаналізувавши вище зазначені ігри було вирішено створити привабливий та унікальний дизайн для гри «What are the ghosts talking about?», який буде запам'ятовуватись користувачам та відображати настрій гри через правильно підібрані кольорові палітри.

Меню гри має бути простим та зручним у використанні.

1.3. Обґрунтування вибору інструментальних засобів для створення графічних елементів

При розробці графічних елементів для гри, вибір відповідних інструментальних засобів є важливим кроком для досягнення бажаного результату. В даному розділі будуть обґрунтовані вибрані для роботи з 2D графікою інструментальні засоби, такі як Adobe Photoshop та Adobe Illustrator.

1. Adobe Photoshop.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Adobe Photoshop(рис.1.4) - це визнаний лідер у галузі растрового графічного дизайну. Використання даної програми має кілька обґрунтувань. По-перше, Adobe Photoshop надає широкі можливості для редагування та маніпулювання растровими зображеннями, що дозволяє створювати складні та деталізовані графічні елементи для гри. Він має розширений набір інструментів, фільтрів та ефектів, що дозволяють досягати бажаних візуальних ефектів у грі. Крім того, Adobe Photoshop підтримує роботу з шарами, що робить процес створення та редагування графіки більш зручним та організованим.

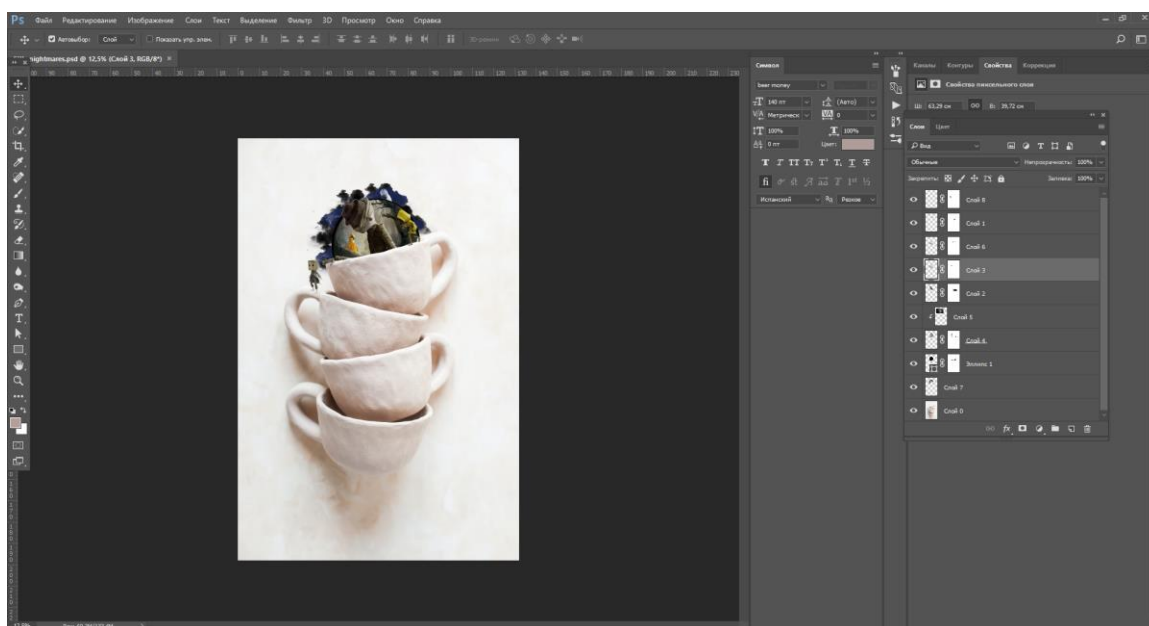


Рисунок 1.4 - Adobe Photoshop

2. Adobe Illustrator.

Adobe Illustrator(рис.1.5) є потужним векторним графічним редактором, який також є відмінним інструментом для створення графічних елементів для 2D гри. Векторна графіка, створена в Adobe Illustrator, має кілька переваг для гейм-дизайнерів. По-перше, векторна графіка може бути легко масштабована без втрати якості, що дуже важливо для графічних елементів, які можуть змінюватися у розмірі на різних пристроях. Відтак, використання Adobe Illustrator дозволяє створювати високоякісні та гнучкі графічні елементи для різних роздільних здатностей. Крім того, Adobe Illustrator пропонує широкий набір інструментів для створення та редагування векторної графіки, таких як

розумне керування кривими, маніпулювання формами, робота з текстом та багато іншого.

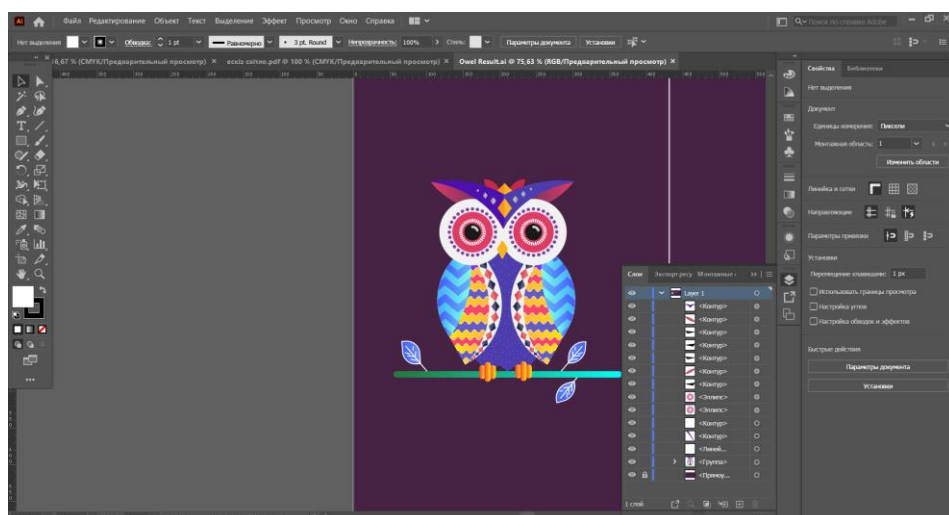


Рисунок 1.5 - Adobe Illustrator

Отже, для створення графічних елементів було вирішено використовувати Adobe Photoshop та Adobe Illustrator так як ці програми є стандартом у галузі графічного дизайну та широко використовуються професіоналами. Вони надають розширений набір інструментів, функцій та можливостей, що дозволяють реалізувати ідеї та створити високоякісну графіку для гри.

1.4. Пайплайн

Пайплайн(рис.1.6) - це документ, що візуалізує процес розробки продукту. Він є послідовністю етапів, розташованих так, що кінець попереднього є початком наступного. Завдяки цьому створюється ефект виробничого конвеєра чи трубопроводу, яким проект рухається від початкової ідеї до конкретного продукту.

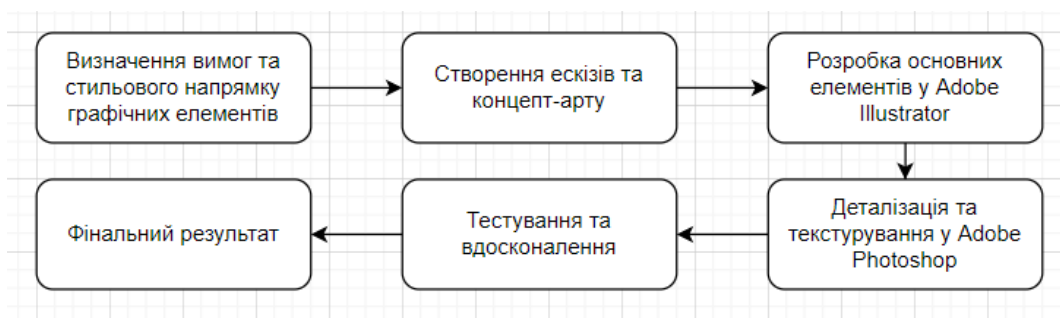


Рисунок 1.6 - Пайплайн створення графічних елементів для гри

Розглянемо реалізацію пайплайну на прикладі локації «Кімната зі світлом».

1. Визначення вимог та стильового напрямку графічних елементів.

Дана кімната має містити в собі предмети побуту, які будуть виконані в одному стилі та ті, з якими зможе взаємодіяти користувач. Локація має бути темною, оскільки особливість даної кімнати це взаємодія користувача зі світлом.

Предмети, з якими взаємодіє користувач:

- коробка;
- світильники на стелі;
- велика лампа;
- свічка;
- банка;

Також необхідно розробити дизайн інвентарю, яким також буде користуватись гравець.

Для підбору кольорової палітри було використано Adobe Color(рис.1.7), який дозволяє користувачам створювати, відтворювати та керувати кольоровими схемами.

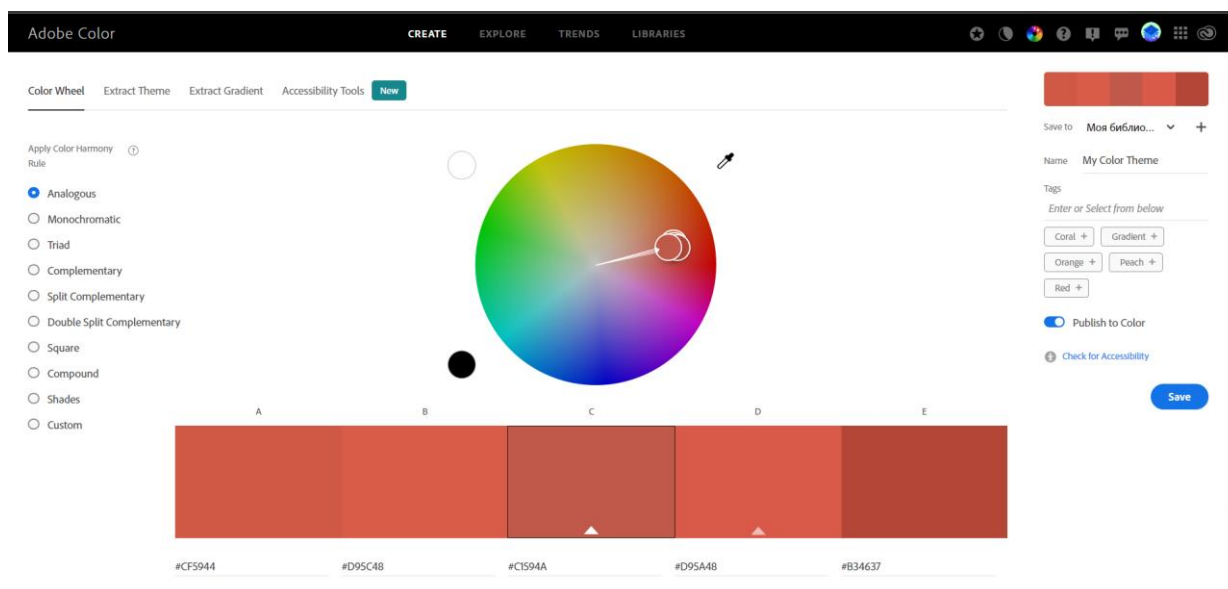


Рисунок 1.7 - Підбір кольорової палітри, використовуючи Adobe Color

2. Створення ескізів та концепт-арту.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Початковий ескіз було виконано вручну на папері (рис.1.8) та після чого було створено концепт-арт в Adobe Illustrator (рис.1.9).

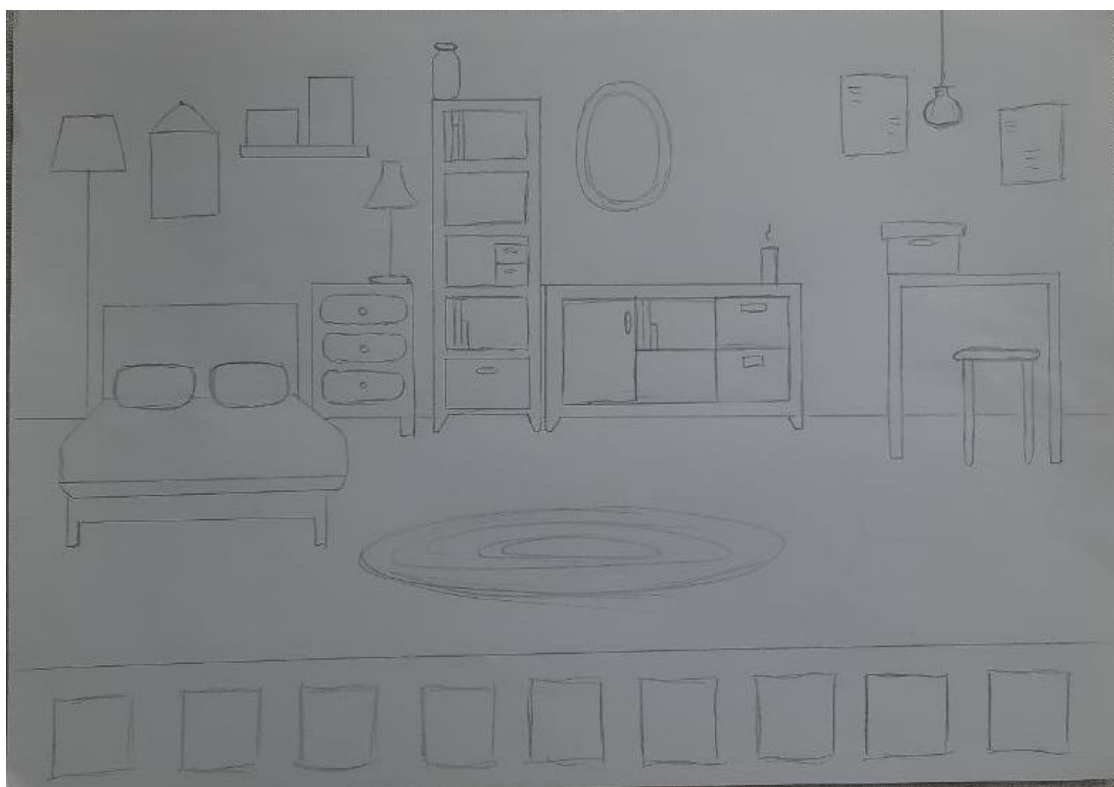


Рисунок 1.8 - Ескіз локації

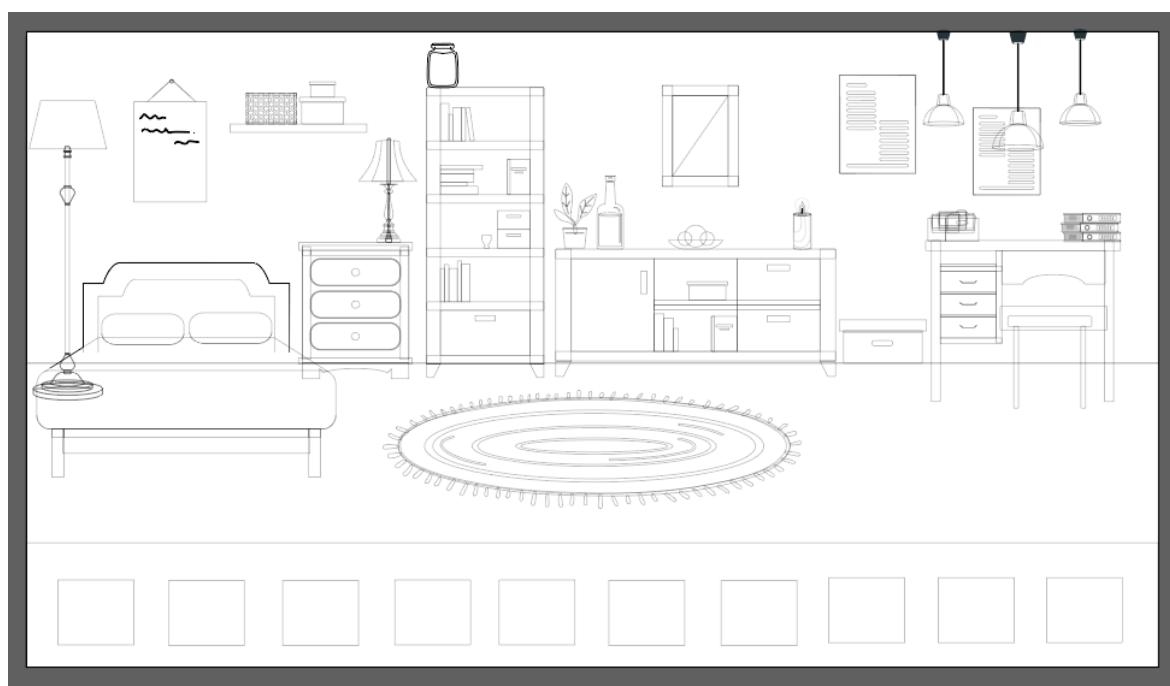


Рисунок 1.9 - Концепт-арт, створений за допомогою Adobe Illustrator

3. Розробка основних елементів у Adobe Illustrator.

Під час даного етапу було повністю розроблено локацію(рис.1.10), починаючи від форм предметів, закінчуючи їх кольором.

Варто зазначити, що інвентар, а також певні світлові ефекти було вирішено реалізувати за допомогою Adobe Photoshop.

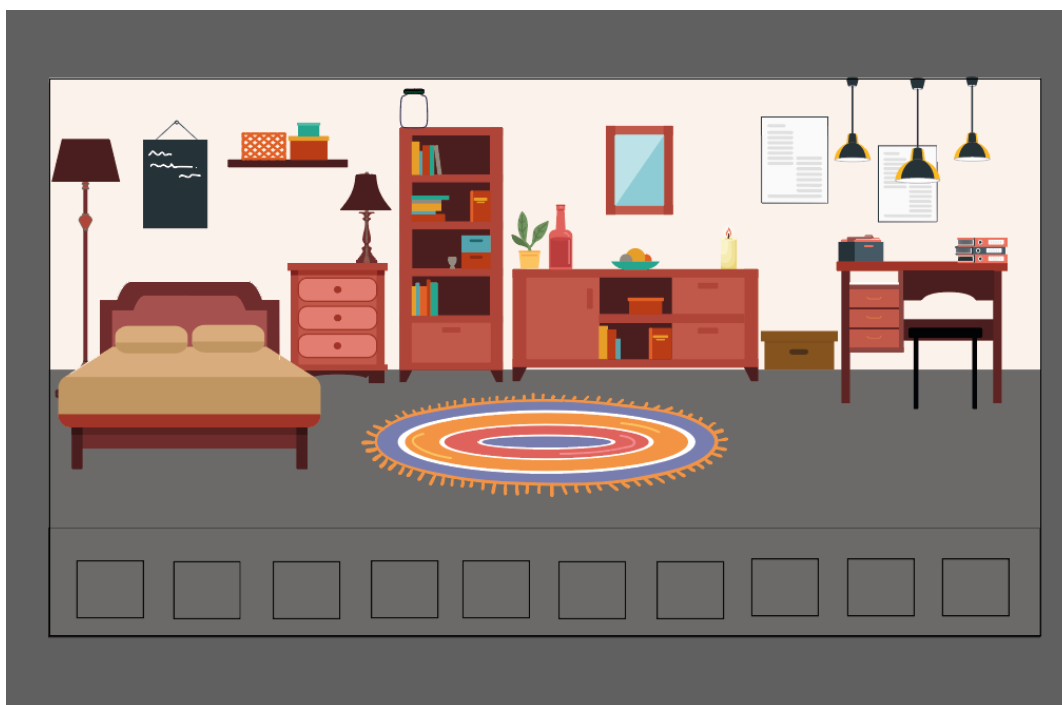


Рисунок 1.10 - Розробка основних елементів у Adobe Illustrator

4. Деталізація та текстурювання у Adobe Photoshop.

За допомогою Adobe Photoshop були створені такі елементи як світло та інвентар(рис.1.11). Також було використано інструмент «Затемнення», щоб підкреслити світлові ефекти, які було створено за допомогою інструмента «Гرادієнт». Щоб підкреслити, що в даній кімнаті давно ніхто не проживає було додано павутиння та розводи на стінах.

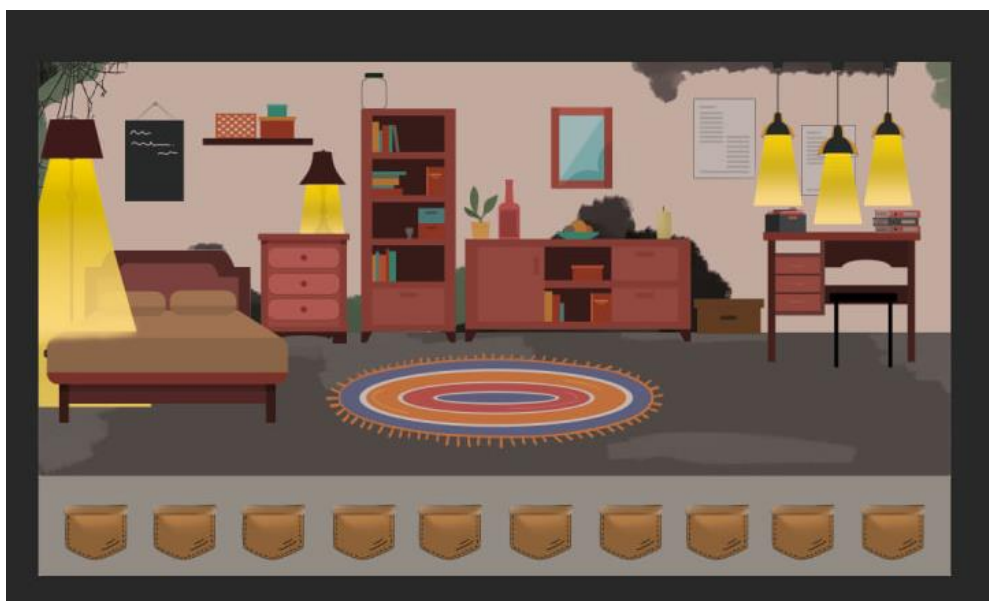


Рисунок 1.11 - Деталізація кімнати за допомогою Adobe Photoshop

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

5. Тестування та вдосконалення.

Було використано фільтр «Розмиття за Гаусом», щоб зробити світло більш природнім(рис.1.12).

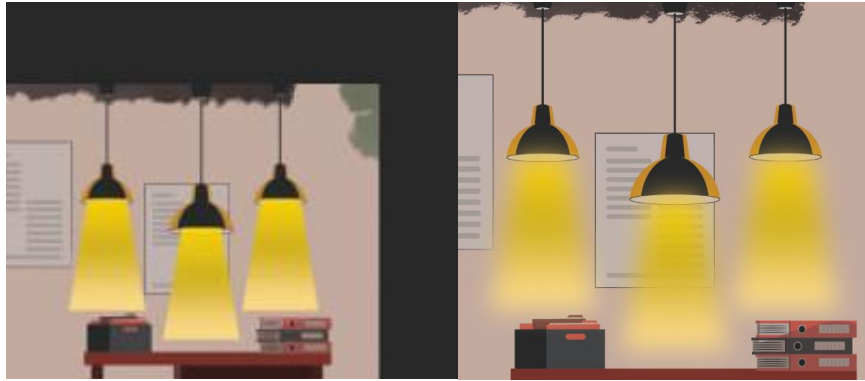


Рисунок 1.12 - До та після використання фільтру «Розмиття за Гаусом»

6. Фінальний результат.

Отже, було створено повноцінну локацію «Кімната зі світлом», яка відповідає усім поставленим вище вимогам(рис.1.13).

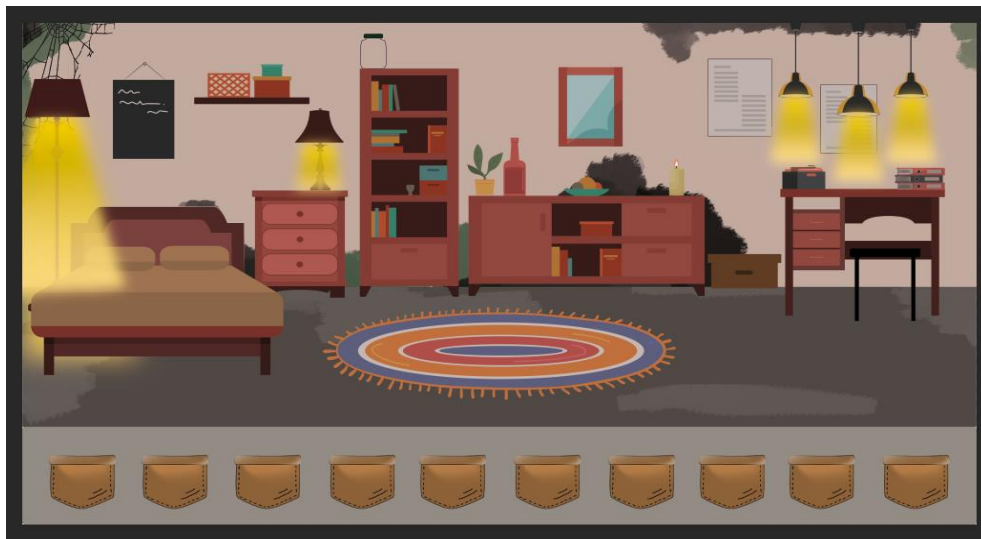


Рисунок 1.13 - Фінальний результат

Висновки до першого розділу

Проведений аналіз аналогів програмного продукту дозволив визначити, який тип графіки варто використати під час створення гри «What are the ghosts talking about?» та які панелі меню необхідно створити. Було обрано графічні редактори для створення графіки, а також створено пайплайн, який допомагає візуалізувати процес розробки графічних матеріалів.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПАНЕЛЕЙ МЕНЮ ТА РОЗРОБКА ГРАФІЧНИХ МАТЕРІАЛІВ

2.1. Проектування панелей меню за допомогою діаграм UML

Розробка будь-якої гри розпочинається з проектування різних можливостей, які будуть доступні користувачу. Для цього і використовується UML. Основною причиною використання мови UML є спілкування розробників між собою.

Було створено діаграму варіантів використання, яка дозволяє проаналізувати послідовність дій, учасників та обмін даними у системі або процесі(рис.2.1).

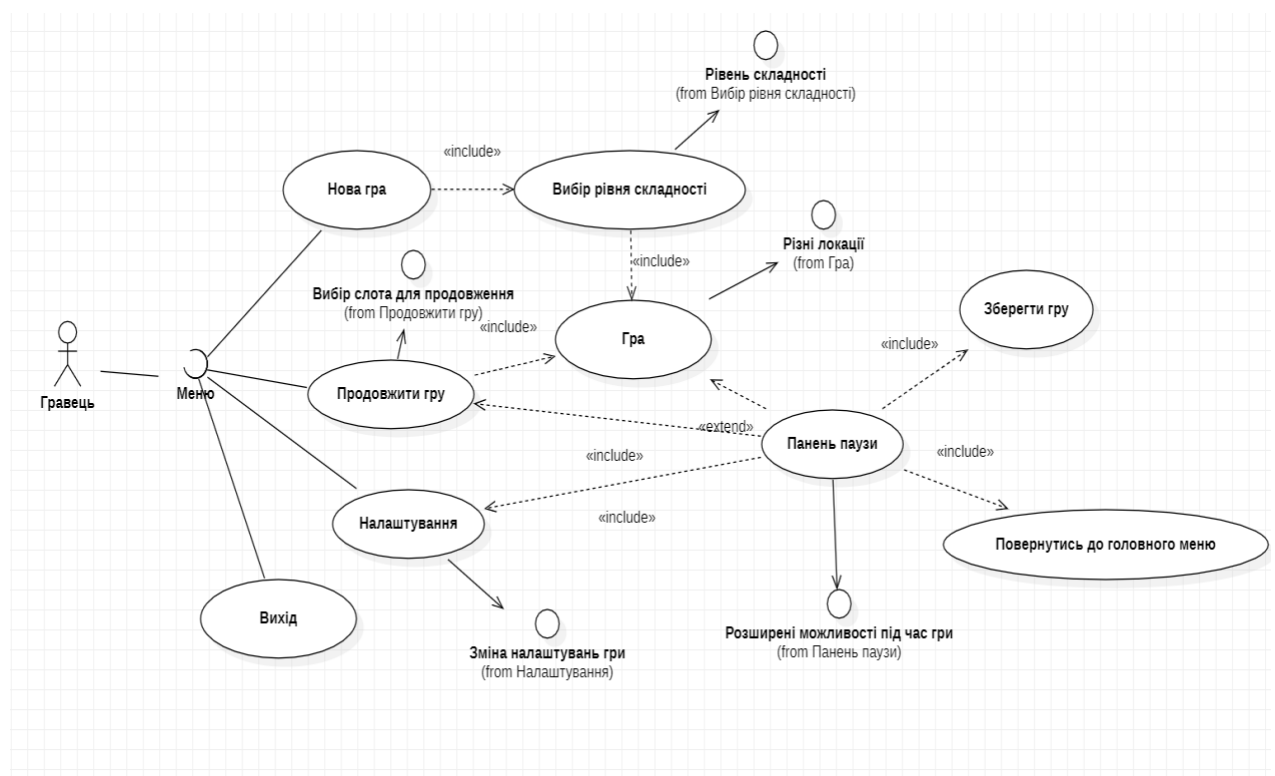


Рисунок 2.1 - Діаграма варіантів використання

Перед гравцем постає інтерфейс головного меню, яке складається з таких пунктів: «Нова гра», «Продовжити гру», «Налаштування» та «Вихід». Якщо гравець обирає пункт «Нова гра», то перед ним постає інтерфейс, який представляє собою вибір складності гри. Після вибору складності на гравця очікує сама гра, яка містить різні локації, а отже, різні інтерфейси. При виборі пункту «Продовжити гру» - гравець обирає слот гри, в якому він хоче продовжити грати. Також під час процесу гри можна звернутись до «Панель

паузи», яка включає в себе різноманітні налаштування. «Панель паузи» також є окремим інтерфейсом.

Діаграма атиквності – ще одна важлива діаграма UML, що описує динамічні аспекти системи. Це, по суті, блок-схема, що представляє потік від однієї дії до іншої. Діяльність може бути описана як робота системи.

На рисунку 2.2 зображена діаграма активності при виборі користувачем пункту «Нова гра». Після того як даний пункт було обрано, перед користувачем постає вибір щодо складності подальшої гри. В залежності від рішення гравця будуть підвантажуватись все ті ж локації, але з головоломками різної складності.

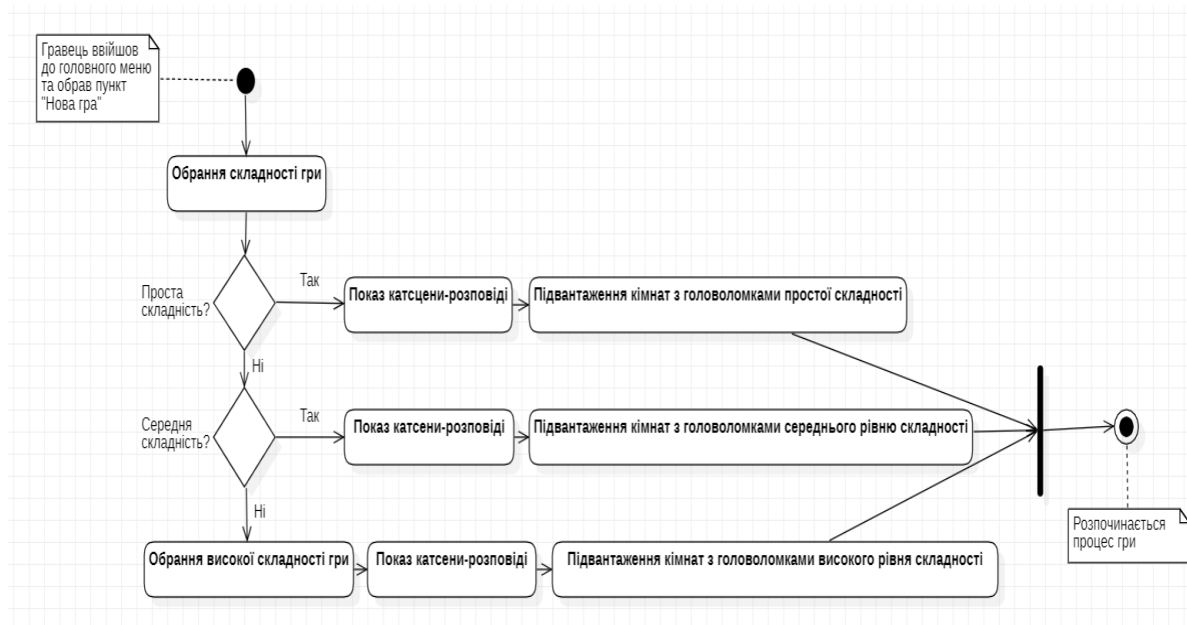


Рисунок 2.2 - Діаграма активності при виборі пункту «Нова гра»

На рисунку 2.3 зображена діаграма активності при виборі користувачем пункту «Продовжити гру». Після того як даний пункт було обрано, в системі програми здійснюється пошук файлів збереження та, якщо такі файли були знайдені, з'являється інтерфейс зі слотами, в які була збережена гра, відповідно до її складності і надалі відбувається підвантаження локації, на якій зупинився користувач. Якщо ж файлів збереження не було знайдено, то користувачу буде показано відповідне повідомлення.

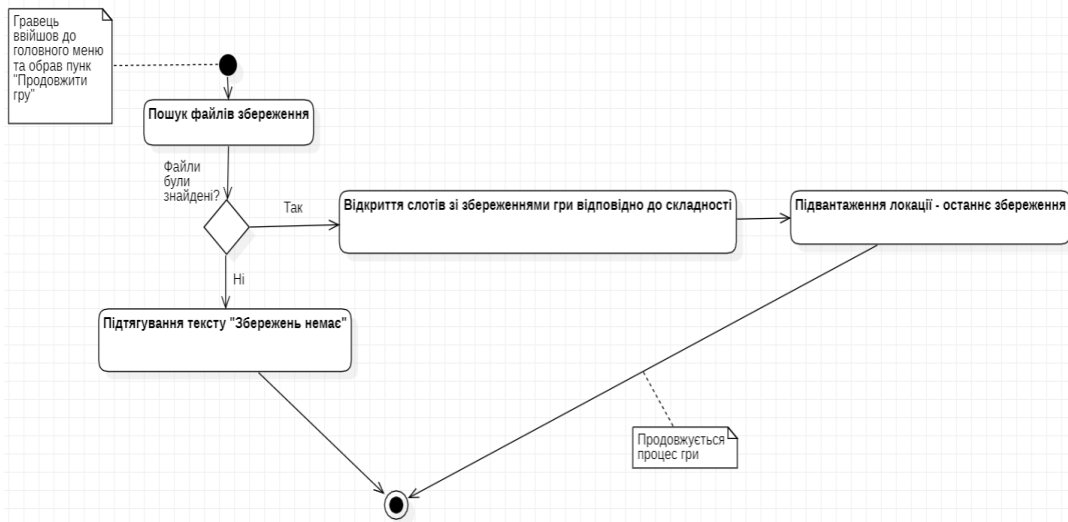


Рисунок 2.3 - Діаграма активності при виборі пункту «Продовжити гру»

На рисунку 2.4 зображена діаграма активності при виборі користувачем пункту «Панель паузи».

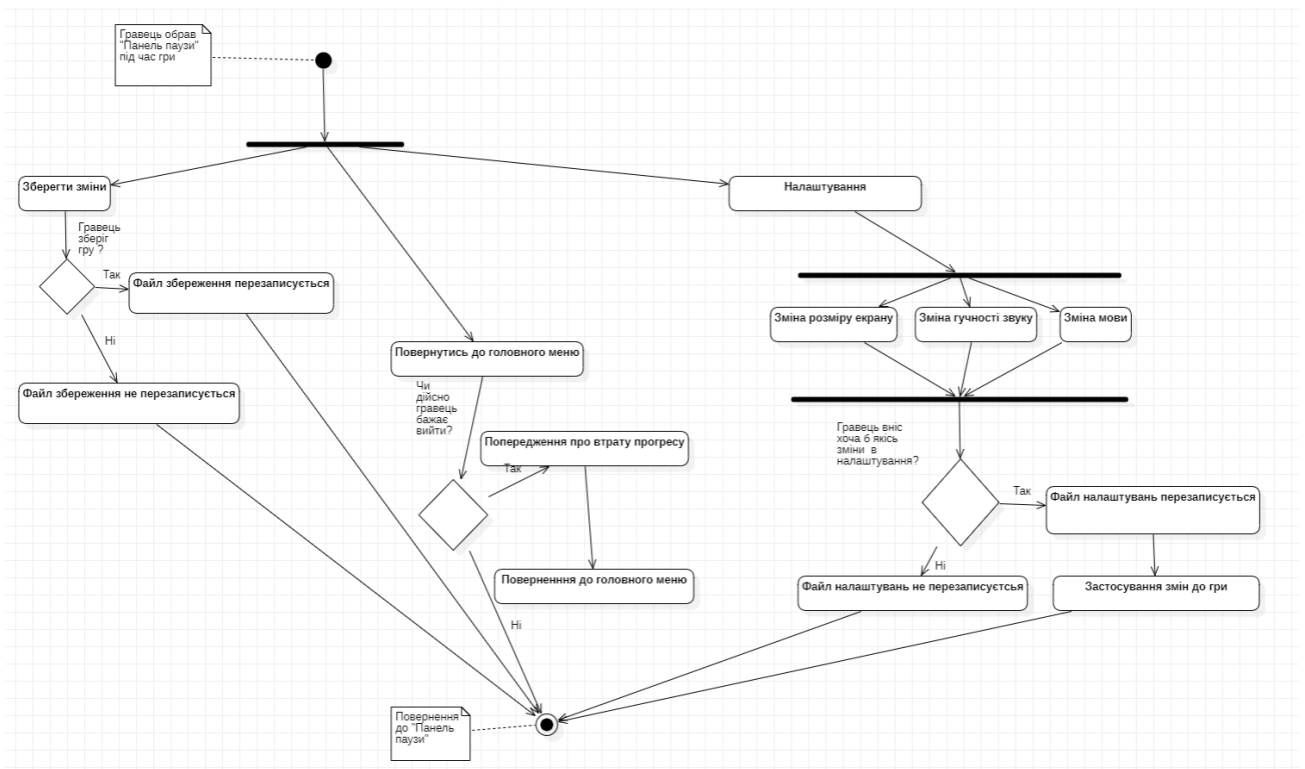


Рисунок 2.4 - Діаграма активності при виборі пункту «Панель паузи»

Дана діаграма є найбільшою серед усіх та містить багато важливої інформації. При відкритті «Панель паузи» перед користувачем постають такі можливості: «Зберегти зміни», «Повернутись до головного меню» та «Налаштування». Якщо гравець обирає «Зберегти зміни», то файл збереження перезаписується, інакше – ні. Якщо ж гравець обирає пункт

«Повернутись до головного меню», то перед ним постає попередження про можливу втрату прогресу. В пункті «Налаштування» гравець може внести зміни в такі пункти як «Розмір екрану», «Гучність» та «Зміна мови». Якщо ж гравець вніс хоча б якісь зміни, то файл налаштувань перезаписується та дані зміни застосовуються до гри.

Діаграма класів – це UML-діаграма, яка описує систему, візуалізуючи різні типи об'єктів усередині системи та види статичних зв'язків, що існують між ними. Вона також ілюструє операції та атрибути класів.

Зазвичай вони використовуються для вивчення концепцій галузі, розуміння вимог до програмного забезпечення та опису детальних проектів. Спираючись на дану інформацію було побудовано діаграму класів, що зображена на рисунку 2.5.

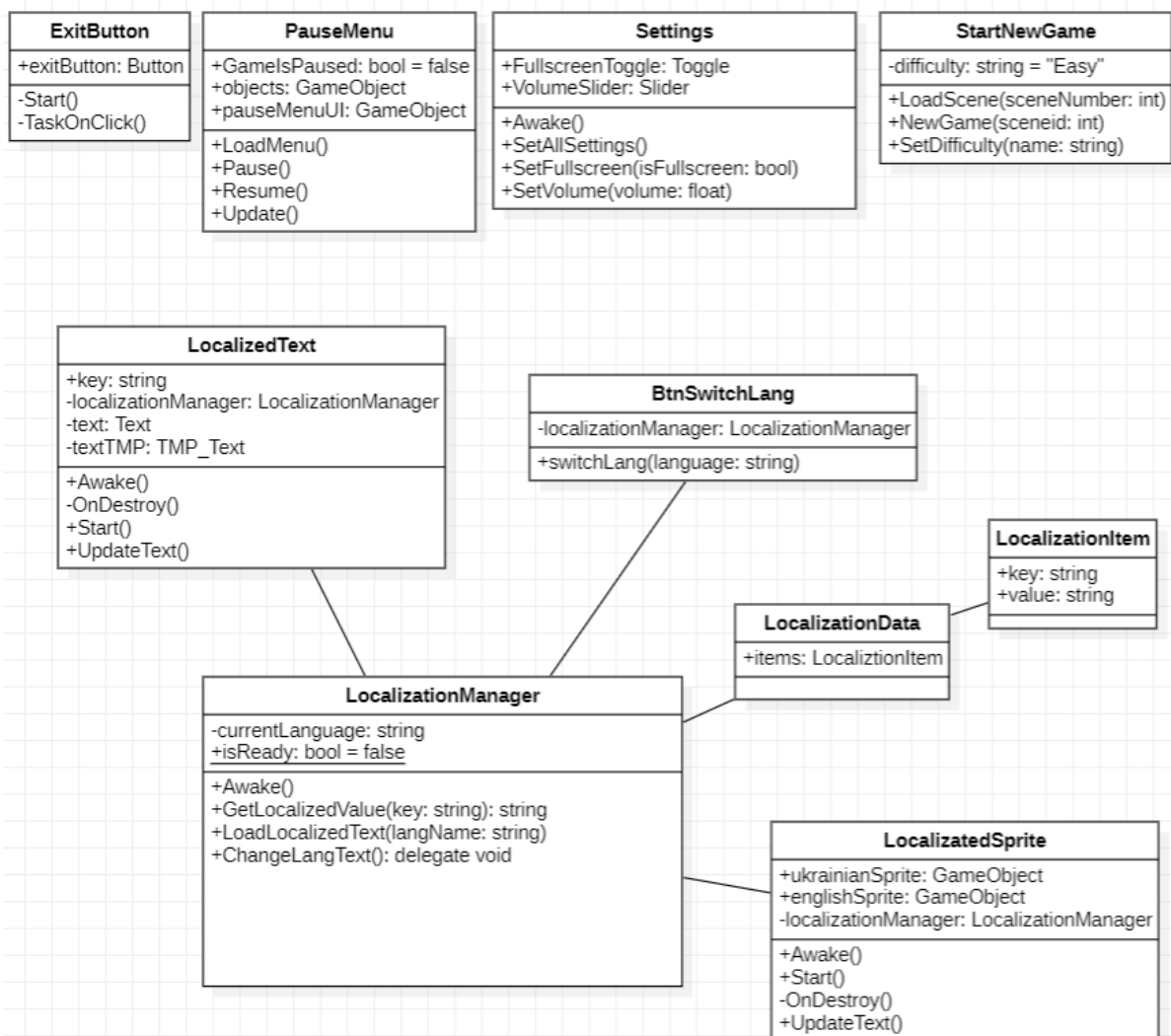


Рисунок 2.5 - Діаграма класів

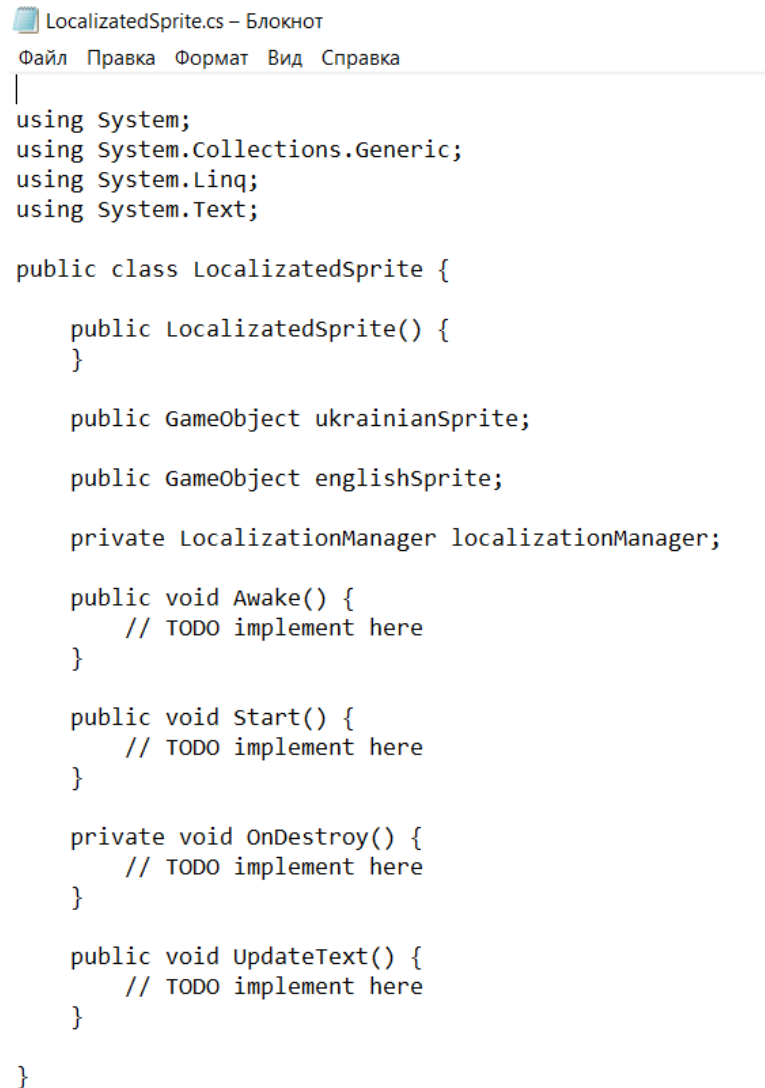
Опис класів:

1. ExitButton – публічний клас, який викликається лише тоді, коли гравець натискає на кнопку «Вихід».
2. PauseMenu – публічний клас, який викликається при натиску гравцем клавіши Escape.
3. Settings – публічний клас, який відповідає за налаштування гри, а саме: зміна мови, гучність звуку, масштабування екрану.
4. StartNewGame – публічний клас, який викликається, коли гравець натискає на кнопку «Нова гра».
5. LocalizationData – публічний клас, який буде використовуватись для зберігання текстів різної мови в файлі json по принципу ключ-значення.
6. LocalizationManager – публічний основний клас, в якому буде визначатись мова системи за допомогою статичного класу PlayerPrefs. В властивості CurrentLanguage отримаємо доступ до поточної мови. В методі LoadLocalizedText ми отримаємо шлях до json-файлів, в яких буде міститись переклад усіх текстів.
7. BtnSwitchLang – публічний клас, який буде застосовано, коли користувач змінить мову в налаштуваннях.
8. LocalizedText – публічний клас, що буде додаватись до кожного текстового елемента на сцені. Також в ньому буде вказуватись ключ, і при зміні мови кожен LocalizedText буде оновлювати значення того тексту, до якого доданий цей клас, беручи значення з файлу по ключу.
9. LocalizedSprite – публічний клас, який використовується для зміни написів, що подані у вигляді картинок.

Після проектування діаграми класів її було конвертовано в C# код. Для цього усі класи було поміщено в Package та за допомогою Tools→C#→Generate Code було експортовано до теки під назвою «Scripts».

В результаті було отримано 10 C#-скриптів готових до використання(рис.2.6).

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24



```

LocalizedSprite.cs – Блокнот
Файл  Правка  Формат  Вид  Справка

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class LocalizedSprite {

    public LocalizedSprite() {
    }

    public GameObject ukrainianSprite;

    public GameObject englishSprite;

    private LocalizationManager localizationManager;

    public void Awake() {
        // TODO implement here
    }

    public void Start() {
        // TODO implement here
    }

    private void OnDestroy() {
        // TODO implement here
    }

    public void UpdateText() {
        // TODO implement here
    }

}

```

Рисунок 2.6 - Приклад згенерованого скрипта під назвою LocalizedSprite

2.2. Основні етапи реалізації панелей головного меню

Спочатку було розроблено клас StartNewGame, який застосований до компоненту ChooseDifficulty в Unity для керування початком нової гри.

Основні методи і функції класу мають наступні завдання:

1. SetDifficulty(string name): цей метод встановлює рівень складності гри. Він отримує рядок name в якості параметра, який відповідає обраному рівню складності. Значення цього параметра зберігається у змінній difficulty, а потім виводиться у консоль.

```

public void SetDifficulty(string name)
{
    difficulty = name;
    Debug.Log(difficulty);
}

```

2. `NewGame(int sceneId)`: цей метод створює нову гру з вказаним ідентифікатором сцени. Він використовує `PlayerPrefs` для зберігання обраного рівня складності. Спочатку він встановлює рівень складності, зберігаючи значення змінної `difficulty` у `PlayerPrefs`. Потім він завантажує нову сцену, використовуючи метод `LoadScene(int sceneNumber)` і передає ідентифікатор сцени як параметр. Нарешті, він також зберігає ідентифікатор сцени у `PlayerPrefs` під ключем `"SaveGame"`.

```
public void NewGame(int sceneId)
{
    PlayerPrefs.SetString("Difficulty", difficulty);
    LoadScene(sceneId);
    PlayerPrefs.SetInt("SaveGame", sceneId);
}
```

3. `LoadScene(int sceneNumber)`: цей метод завантажує сцену з вказаним ідентифікатором `sceneNumber` за допомогою функціоналу `Unity SceneManager.LoadScene(sceneNumber)`.

```
public void LoadScene(int sceneNumber)
{
    SceneManager.LoadScene(sceneNumber);
}
```

Також було реалізовано клас `PauseMenu`, який використовується лише тоді, коли користувач хоче поставити гру на паузу. Основні методи і функції класу мають наступні завдання:

1. `SetCanBePaused(bool newBool)`: цей метод встановлює можливість паузи в грі. Він отримує булеве значення `newBool` в якості параметра, яке визначає, чи може гра бути призупинена. Значення цього параметра зберігається у змінній `GameCanBePaused`.

```
public void SetCanBePaused(bool newBool)
{
    GameCanBePaused = newBool;
}
```

2. `Update()`: цей метод перевіряє, чи натиснута клавіша `Escape`. Якщо гра вже призупинена, то виконується метод `Resume()`, щоб продовжити гру. У

випадку, коли гру можна призупинити (GameCanBePaused дорівнює true), викликається метод Pause(), щоб призупинити гру.

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.Escape))
    {
        if (GameIsPaused)
        {
            Debug.Log("Resume");
            Resume();
        }
        else if (GameCanBePaused)
        {
            Pause();
        }
    }
}
```

3. Resume(): цей метод відновлює гру після призупинення. Він встановлює значення GameIsPaused на false, що позначає, що гра не призупинена. Також він ховає візуальний елемент паузи pauseMenuUI, відновлює видимість всіх об'єктів з масиву objects і встановлює швидкість часу (Time.timeScale) на 1, щоб гра продовжувалась звичайним темпом.

```
public void Resume()
{
    GameIsPaused = false;
    pauseMenuUI.SetActive(false);
    foreach (GameObject obj in objects)
        obj.SetActive(true);
    Time.timeScale = 1f;
}
```

4. Pause(): цей метод призупиняє гру. Він встановлює значення GameIsPaused на true, що позначає, що гра призупинена. Також він показує візуальний елемент паузи pauseMenuUI, ховає всі об'єкти з масиву objects, встановлює швидкість часу (Time.timeScale) на 0, щоб зупинити рух у грі.

```
void Pause()
{

```

```

    pauseMenuUI.SetActive(true);
    foreach (GameObject obj in objects)
        obj.SetActive(false);
    Time.timeScale = 0f;
    GameIsPaused = true;
}

```

5. **LoadMenu()**: цей метод викликається при переході до головного меню гри. Він викликає метод **Resume()** для відновлення гри перед завантаженням головного меню. Потім він завантажує сцену з ідентифікатором 0 (перша сцена) за допомогою **SceneManager.LoadScene(0)**.

```

public void LoadMenu()
{
    Resume();
    SceneManager.LoadScene(0);
}

```

Наступним було реалізовано клас **Settings** для керування налаштуваннями гри. Основні методи і функції класу мають наступні завдання:

1. **SetAllSettings()**: цей метод встановлює всі налаштування, які були збережені за допомогою **PlayerPrefs**. Він перевіряє, чи існує значення для гучності (**Volume**) і режиму вікна на весь екран (**isFullscreen**) в **PlayerPrefs**. Якщо такі значення існують, вони використовуються для встановлення відповідних елементів керування (**VolumeSlider** і **FullscreenToggle**).

```

public void SetAllSettings()
{
    if (PlayerPrefs.HasKey("Volume"))
    {
        Debug.Log("Set volume");
        VolumeSlider.value = PlayerPrefs.GetFloat("Volume");
    }
    if (PlayerPrefs.HasKey("isFullscreen"))
    {
        FullscreenToggle.isOn = (PlayerPrefs.GetInt("isFullscreen") == 1);
    }
}

```

2. **SetFullscreen(bool isFullscreen):** цей метод встановлює режим вікна на весь екран. Він приймає булеве значення `isFullscreen`, яке вказує, чи потрібно встановити режим вікна на повний екран. Залежно від значення `isFullscreen`, використовуючи `Screen.fullScreen`, встановлюється відповідний режим вікна. Крім того, метод зберігає стан повноекранного режиму в `PlayerPrefs` під ключем `"isFullscreen"`.

```
public void SetFullscreen(bool isFullscreen)
{
    Screen.fullScreen = isFullscreen;
    if(isFullscreen)
        PlayerPrefs.SetInt("isFullscreen", 1);
    else
    {
        PlayerPrefs.SetInt("isFullscreen", 0);
    }
}
```

3. **SetVolume(float volume):** цей метод встановлює гучність звуку гри. Використовуючи `AudioListener.volume`, встановлюється відповідний рівень гучності звуку. Крім того, метод зберігає значення гучності в `PlayerPrefs` під ключем `"Volume"`.

```
public void SetVolume(float volume)
{
    AudioListener.volume = volume;
    PlayerPrefs.SetFloat("Volume", volume);
}
```

2.3. Створення графічних матеріалів для панелей меню

Спочатку було створено дизайн головного меню(рис.2.7), в якому наявні такі розділи: «Нова гра», «Продовжити гру», «Налаштування» та «Вихід», а також назва гри. Оскільки дані розділи є клікабельними, вони були розташовані на окремих шарах. Було вирішено, що саме на цій панелі має бути відображено головного персонажа та предмети, що, безпосередньо, відносяться до гри. Це сумка художника, яка є інвентарем та тиква, що символізує Хелловін, під час якого відбуваються дії гри.

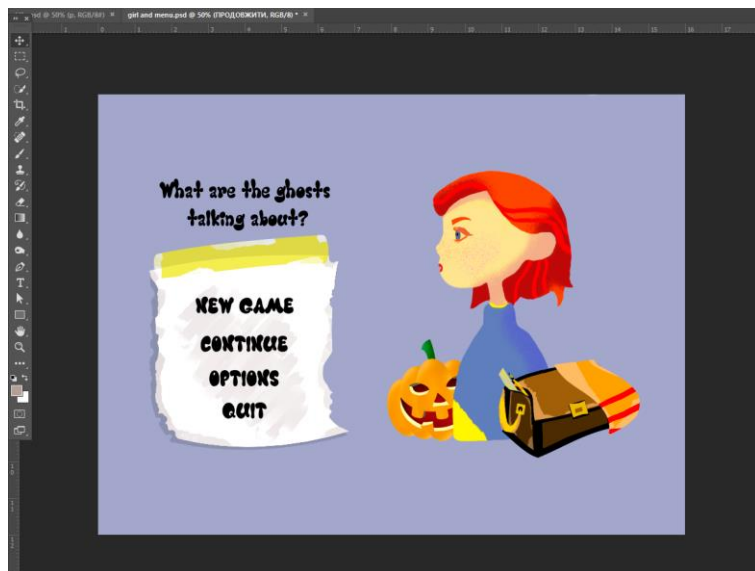


Рисунок 2.7 - Дизайн головного меню

Усі панелі меню було створено за допомогою за допомогою Adobe Photoshop та Freeerik-платформи для безкоштовного та преміум-векторного графічного контенту.

Наступним було реалізовано дизайн для панелі налаштувань(рис.2.8), яка має містити місце для зміни масштабу екрана, зміни гучності та мови, а також кнопку для збереження внесених змін.

Крім цього було реалізовано дизайн панелі меню для вибору складності гри після того, як користувач натиснув кнопку «Продовжити гру»(рис.2.9). Для даної панелі необхідно розробити дизайн двох кнопок – «Завантажити» та «Назад», а також місце, де буде відображатись складність гри, в яку вже встиг пограти користувач.

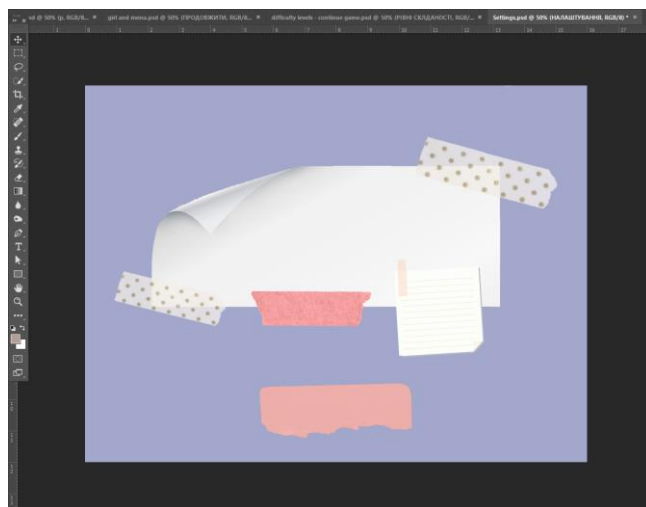


Рисунок 2.8 - Дизайн панелі налаштувань

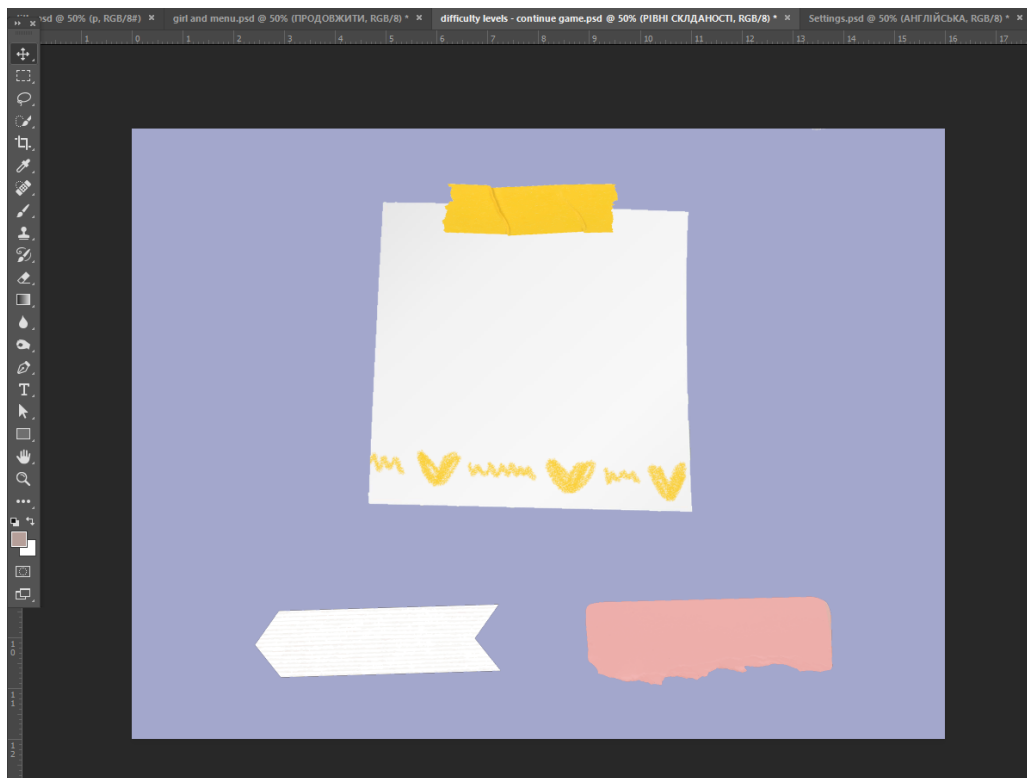


Рисунок 2.9 - Дизайн панелі вибору складності гри

Також було реалізовано дизайн панелі вибору складності після того, як користувач натиснув «Нова гра»(рис.2.10). Для даної панелі необхідно було реалізувати дві кнопки – «Назад» та «Почати нову гру», місце до вибору рівня складності та місце для опису рівня складності.

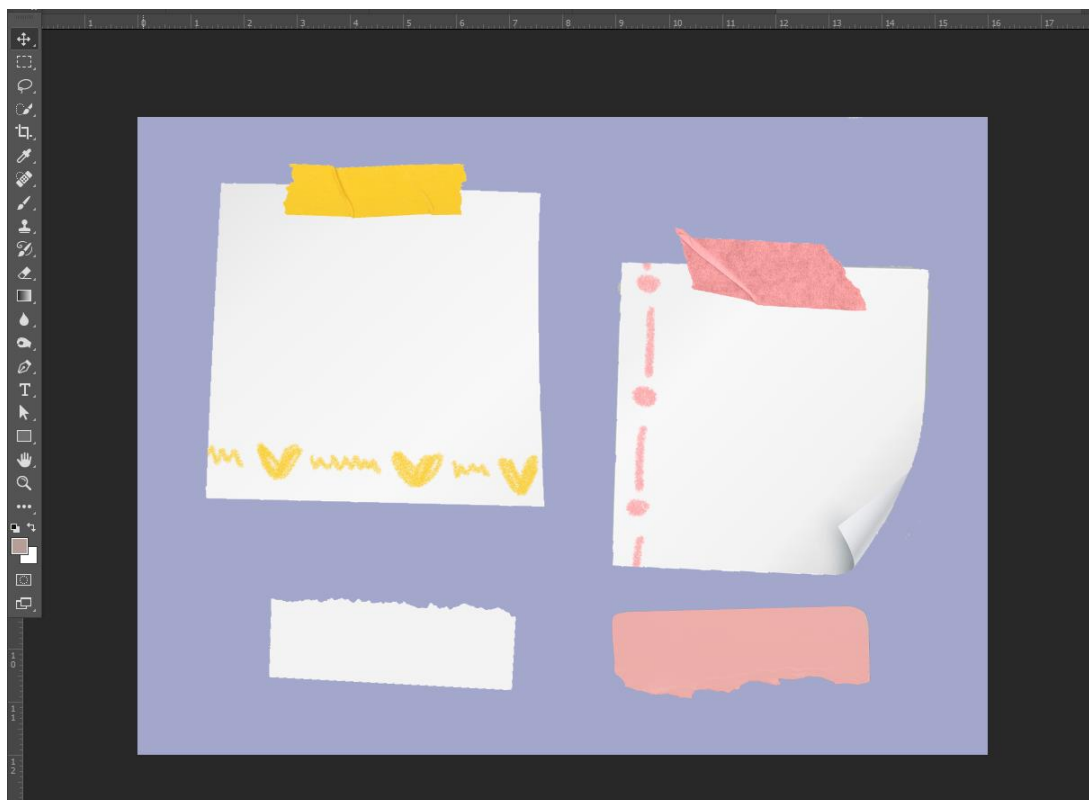


Рисунок 2.10 - Дизайн панелі вибору складності гри

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

І останнім було створено дизайн дл панелі паузи під час гри(рис.2.11). Дана панель має містити всього 3 кнопки – «Продовжити», «Налаштування», «Вихід».

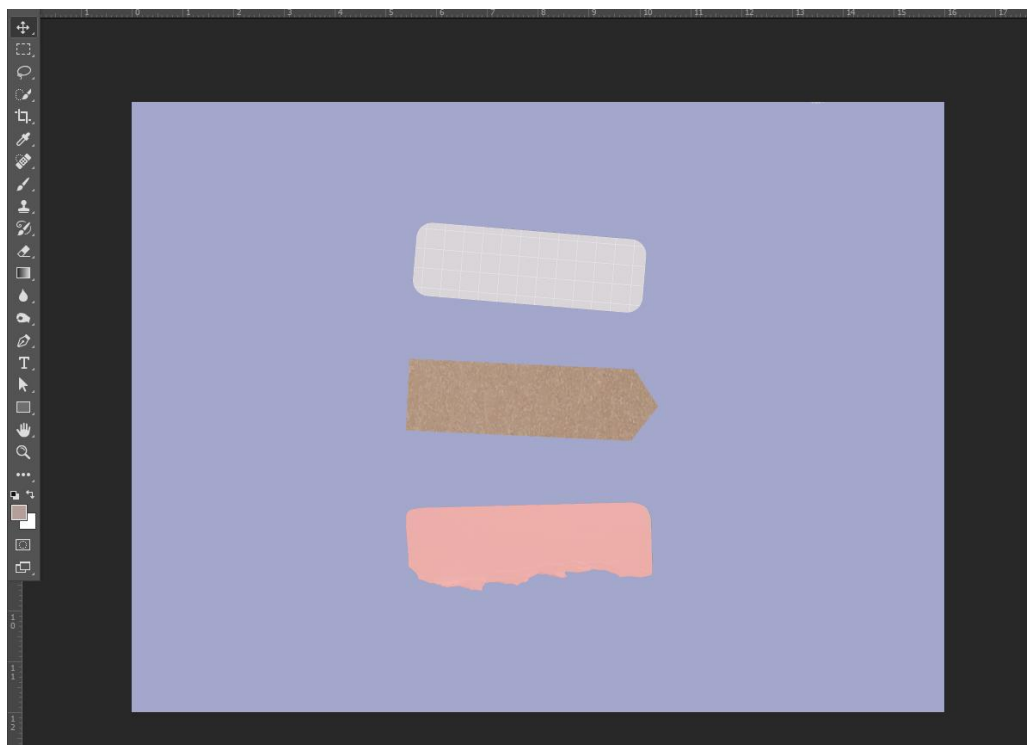


Рисунок 2.11 - Дизайн панелі паузи

2.4. Створення персонажів за допомогою Adobe Photoshop

Спочатку було створено головну героїню-художницю(рис.2.12), яка має такий зовнішній вигляд: виразні риси обличчя та високий зріст. Волосся героїні має довжину до плечей і помаранчевий колір, можливо, з легкими променями світлішого відтінку. Вона має великі, виразні очі, які відображають її допитливість та сміливість. Її шкіра свіжа і рум'яна, з легким природним макіяжем, що підкреслює її природну красу.

Героїня одягнена у стильний, але комфортний одяг, що відображає її художній стиль і незалежність. Вона має на собі м'який в'язаний светр та джинси. Взуття героїні - це зручні, але стильні черевики, які додають їй енергії та рухливості.

Дівчина має з собою свою надійну сумку, в якій вона носить свої малюнки, пензлі, кольорові олівці та інші художні інструменти та яка буде слугувати інвентарем під час гри.



Рисунок 2.12 – Головна героїня

Також було реалізовано привида – другого головного героя гри(рис.2.13). Він має форму класичного довгого рушника, що повільно пливе у повітрі. Проте, для створення цікавішого вигляду, його можна дещо модифікувати. Привид має лише два великих ока. Загалом, привид повинен виглядати дружелюбним, але водночас трохи таємничим і загадковим. Його форма та характеристики повинні викликати бажання розкрити його історію і допомогти йому вирішити проблему, що тримає його у цьому будинку.



Рисунок 2.13 – Привид

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

Також присутні і другорядні персонажі. Графець може взаємодіяти з домовичком(рис.2.14) та скелетом(рис.2.15) і просто спостерігати за чортиками(рис.2.16) і круп'є(рис.2.17).

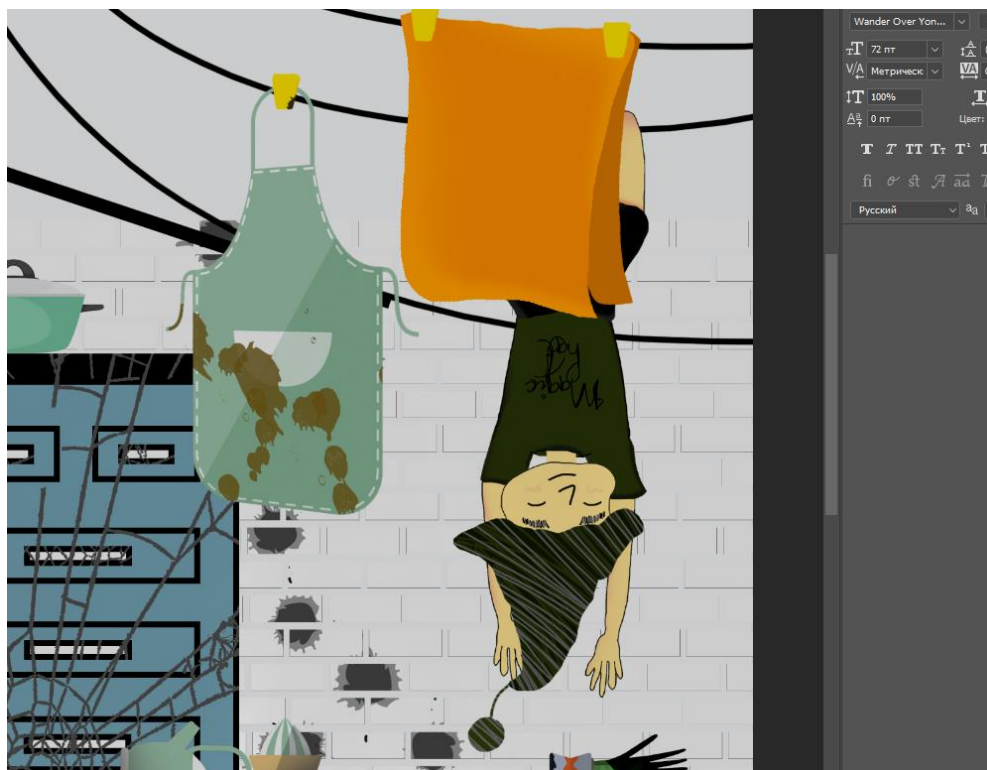


Рисунок 2.14 – Домовичок

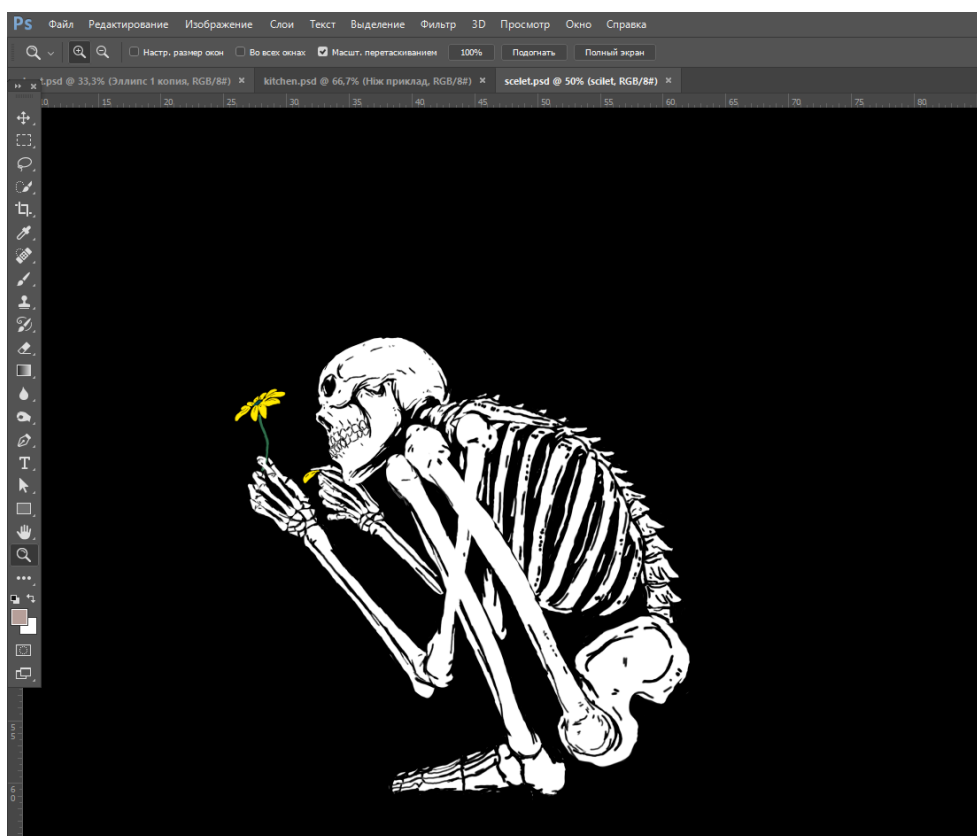


Рисунок 2.15 – Скелет

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34



Рисунок 2.16 – Чортики



Рисунок 2.17 – Круп'є

2.5. Створення катсцен за допомогою Adobe Photoshop

Початкова і кінцева сцени в грі є важливими елементами для створення належної структури і надання гравцеві контексту та цілі гри.

У початковій катсцені(рис.2.18) ми представляємо героїню, дівчину-художницю, яка малює у своєму домі. Вона отримує лист від незнайомця, який виявляється запрошенням на вечірку. Початкова сцена встановлює контекст і запускає події, що ведуть до основної історії.

Катсцени було вирішено створити у вигляді карток, які і будуть розповідати усю історію. Загалом було створено 17 карток для початкової катсцени.



Рисунок 2.18 – Перші дві картки початкової катсцени

У 3 та 4 сценах(рис.2.19) головна героїня читає вміст самого листа, а в 5-6 сценах(рис.2.20) вона вже прибуває до назначеного місця (будинку) та заходить до середини. З 6 картки у дівчинки з'являються фрази гумористичного змісту, які точно не залишать байдужими користувача.

Під час 7-8 сцени(рис.2.21) головна героїня наштовхується на привида та це стає вихідною точкою для гри і початком пригоди головної героїні.



Рисунок 2.19 – Третя та четверта картки



Рисунок 2.20 – П’ята та шоста картки



Рисунок 2.21 – Сьома та восьма картки

У кінцевій катсцені(рис.2.22) героїня доходить до розв'язання таємниці будинку та привида. Вона збирає картину разом, що є ключем до визволення привида і завершення його прокляття. Кінцева сцена показує як героїня примиряється з привидом і допомагає йому знайти спокій та як вона покидає будинок, відчуваючи задоволення від своєї подорожі та неймовірну втому(рис.2.23). Це підсилює відчуття завершеності і задоволення від успішного завершення гри.

Загалом було створено 5 карток для фінальної катсцени.

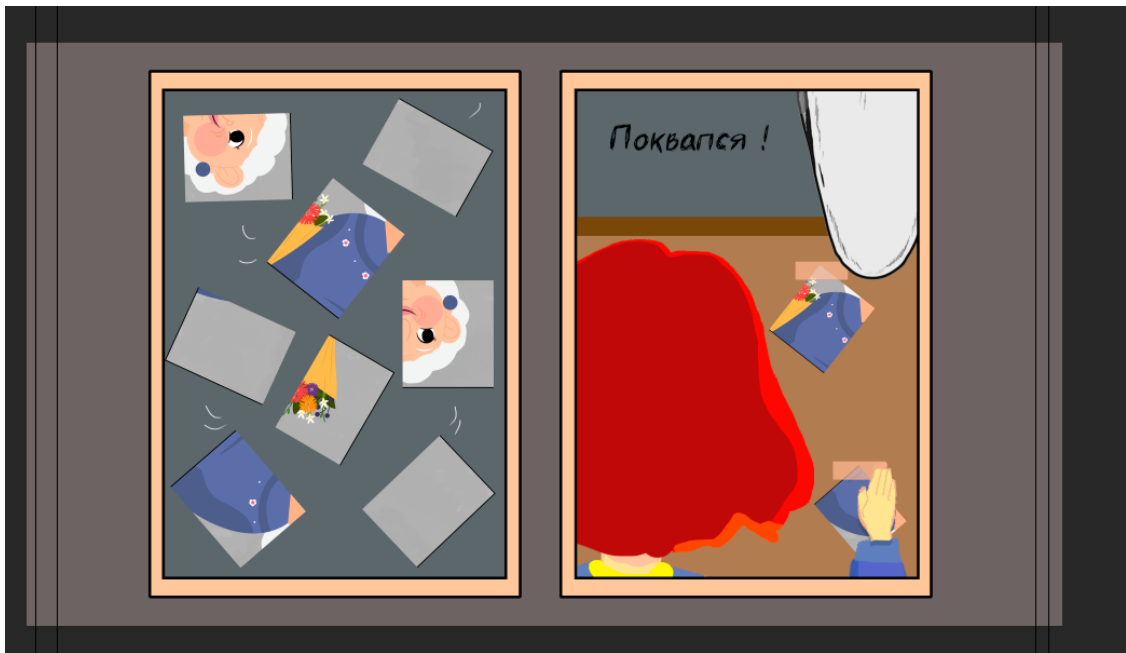


Рисунок 2.22 – Перші дві картки фінальної катсцени



Рисунок 2.23 – Фінальні картки катсцени

2.6. Створення текстового документу з підказками

Створення текстового документа з підказками(рис.2.24) до гри може бути дуже корисним інструментом для покращення геймплею та заглиблення гравця в історію. Підказки є важливими не тільки тому, що вони допомагають гравцеві, якщо він потрапив до складної ситуації, а й надають додаткову інформацію. В текстовому документі можна включити додаткові деталі, факти або історії, які не були згадані безпосередньо в грі. Це дозволяє гравцям

глибше зануритись в сюжет і отримати більше контексту щодо персонажів, подій або місць, що зустрічаються на їх шляху.

tipsAttractions.txt – Блокнот

Файл Правка Формат Вид Справка

Так, в мене є окрема кімната з ігровими автоматами, міг собі дозволити.

Ці чорти і сюди дібрались. Розпивають мою колекцію пива.

Покер це файна гра як на мене.

Моя улюблена гра - це тетріс.

Заради свого друга-інтроверта я купив автомат з танцями на одну персону. Він танцював годинами.

Дітей як і онуків я не маю, тому грати приходили сусідські діти, а їх батьки полюбляли сидіти за зеленим столиком, хіхі.

В мене була файна колекція пива, елітного. Друзі привозили з різних куточків світу.

Гра "Відлупчий бобра" - це просто знахідка для тих, кому не має куди подіти свою агресію.

Нічого не можу сказати про нашого файного круп'є. Вперше бачу. До того ж, він постійно мовчить.

А гру з лабіринтом обожнювала моя дружина.

Ці чортики були б файними, якби не були б крадіями.

Емблему на стіні, що напряму вказує на мою колекцію, я сам робив. Моя гордість.

Взагалі тут мало бути приміщення для картоплі, але ідея з автоматами - найкраща. Занадто багато місця для якоїсь там картоплі.

Фіолетова підлога? Чому б і ні?

Тут мав би ще бути дискошар, але ці чорти кудись його поділи.

Якщо маєш бажання, можеш взяти кілька баксів. Надіюсь, що тобі не прийдеться за них боротись.

Чотири стільці... Взагалі вони іменні. Для чотирьох друзів. Ми були майже як мушкетери.

Кольори для автоматів обирала дружина, а ми разом фарбували. Ми придбали їх на розпродажі.

Автомати працюють і без монеток, дивно. Хоча мені вже і байдуже. Яке діло мені до тих грошей ?

Ці чорти навіть б'ються за мої гроші.

Рисунок 2.24 – Приклад файлу з підказками для локації «Атракціони»

Також було створено текстовий файл з фразами головної героїні(рис.2.25), які будуть виокристовуватись, коли гравець захоче повзаємодіяти з будь-якими предметами побуту.

*girlAttractionsPhrases.txt – Блокнот

Файл Правка Формат Вид Справка

Файні створіння. Цікаво, а можна взяти одного додому ? (чорт)

Юху, пиво! Як я розумію чорта на полиці (діапазон з пивом, пх)

Мені не потрібні чужі гроші. (гроші)

Не найкращий час для гри в карти. (стіл)

Рисунок 2.25 – Приклад файлу з фразами дівчини для локації «Атракціони»

Висновки до другого розділу

Було розглянуто основні етапи проектування панелей меню та розробки графічних матеріалів для гри "What are the ghosts talking about?". Застосування діаграм UML допомогло спланувати структуру панелей меню, а використання Unity та C# надали можливість реалізувати їх функціонал.

Для створення графічних матеріалів, включаючи панелі меню, персонажів та катсцени, було використано Adobe Photoshop, а також Freepik-платформа для безкоштовного та преміум-векторного графічного контенту. Це дозволило створити зображення, які відповідають загальному стилю та настрою гри.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Крім того, було створено текстові документи з підказками, що допомагатимуть гравцям отримувати додаткову інформацію та підтримувати їх у випадках, коли вони зіштовхуються з труднощами.

Застосування вищезазначених методів та інструментів у процесі розробки панелей меню та графічних матеріалів допоможе покращити візуальний вигляд та взаємодію гравців з грою "What are the ghosts talking about?".

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

3.1. Інтерфейс та порядок роботи з програмним додатком

Представлено дії гравця на 2 локаціях («Атракціони», «Кухня»), а також робота з панелями меню.

Коли гравець запускає гру, то перше, що він бачить – головне меню(рис.3.1). Якщо ж гравець обирає розділ «Нова гра», то для нього відкриється меню для вибору складності гри, в якому міститься інформація про складність, а також наявна можливість повернутись до головного меню(рис.3.2).



Рисунок 3.1 – Головне меню

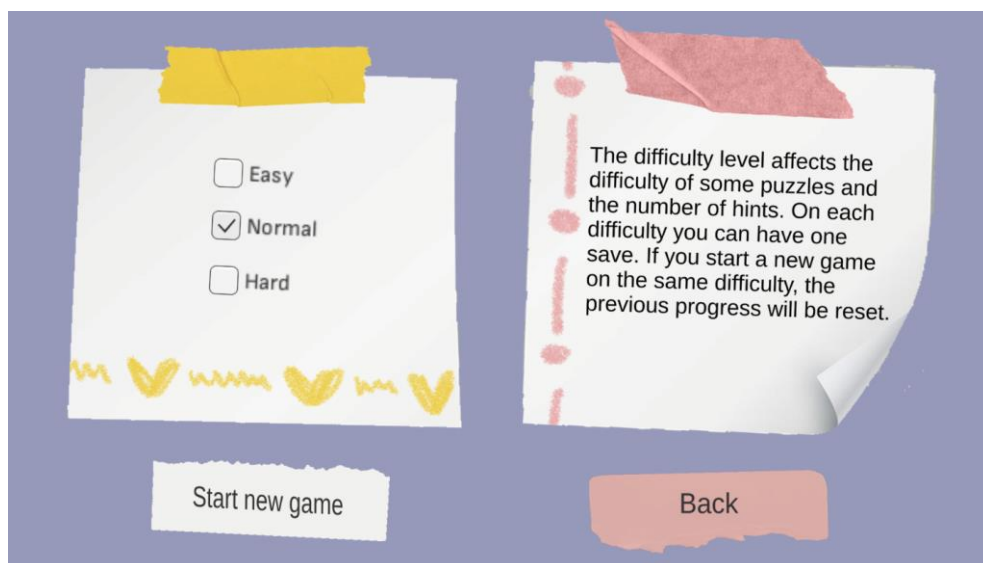


Рисунок 3.2 – Вибір складності гри

Якщо ж гравець обирає розділ «Продовжити гру», то перед ним постає меню для вибору складності гри, в яку він вже грав(рис.3.3). Тут наявна можливість продовжити гру з того місця, де гравець її закінчив раніше.

При виборі розділу «Налаштування», користувач потрапляє до панелі меню, де наявні усі можливі налаштування для гри, а саме: зміна гучності, мови, розміру екрана(рис.3.4).

Якщо ж гравець вже знаходиться на певній локації, але з якихось причин має бажання поставити гру на паузу, то натиснувши Esc, він може звернутись до панелі паузи(рис.3.5), в якій наявні такі можливості – «Продовжити гру», «Налаштування» та «Вихід».

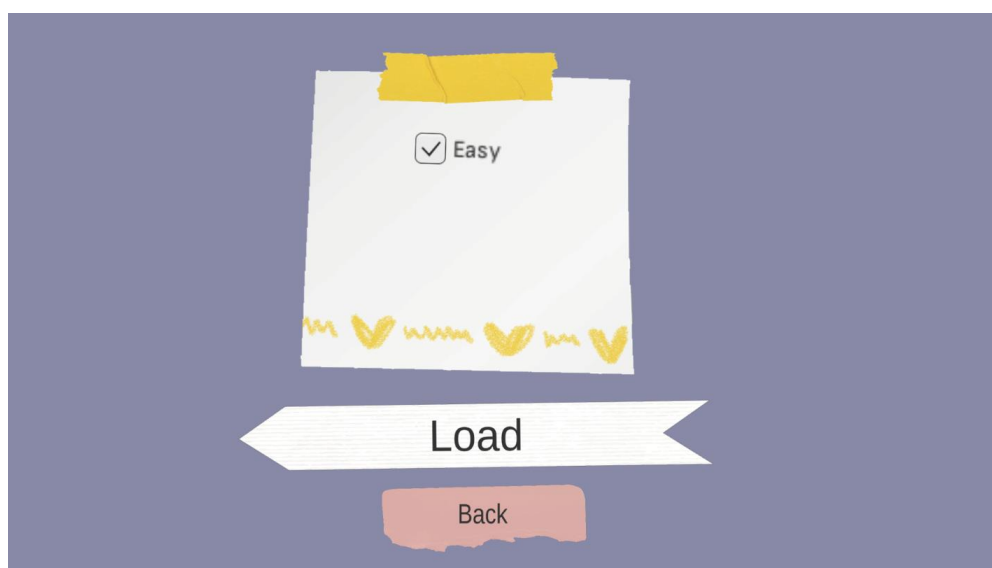


Рисунок 3.3 – Вибір складності гри

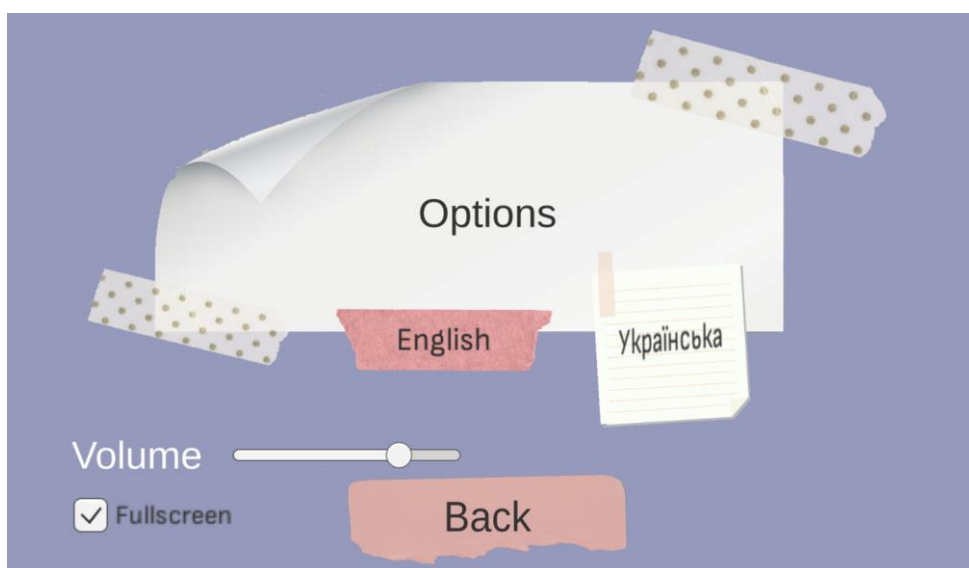


Рисунок 3.4 – Налаштування гри

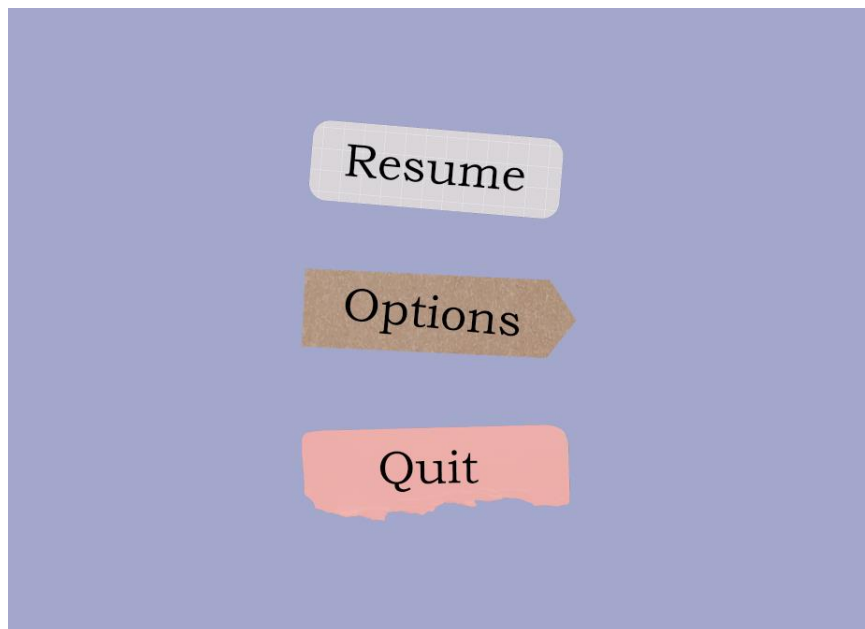


Рисунок 3.5 – Панель паузи

Якщо ж користувач має бажання покинути гру з панелі паузи, то перед ним з'явиться нове вікно з наступним повідомленням: «Ви дійсно хочете вийти? Весь незбережений прогрес буде втрачено»(рис.3.6). І гравець може обрати: або ж повернутись до гри або її покинути.

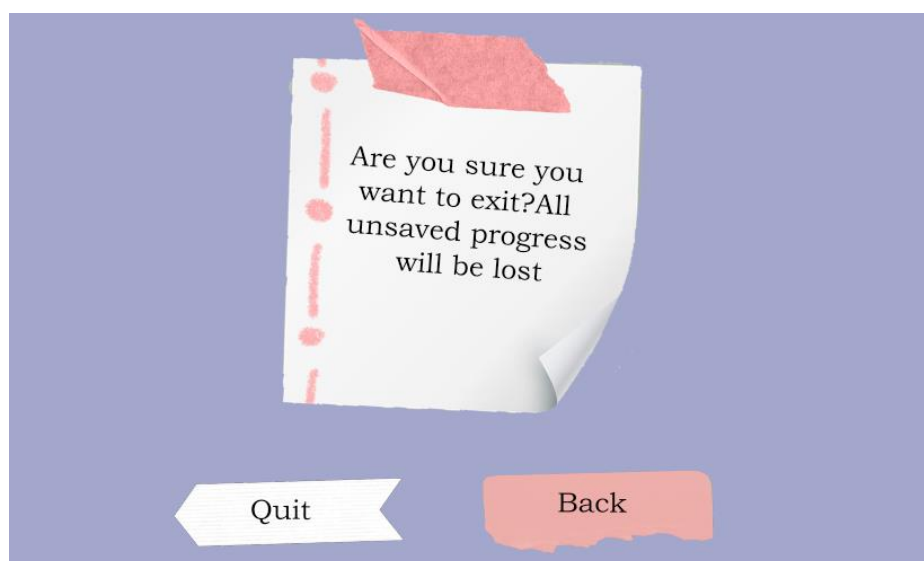


Рисунок 3.6 – Вікно з попередженням про втрату прогресу

Коли гравець опиняється на локації «Атракціони»(рис.3.7), то він може відкрити головоломки натискаючи на різні ігрові автомати. Наприклад, якщо гравець натисне на зелений автомат, то перед ним постане головоломка під назвою «Відгамцелити бобра»(рис. 3.8). за успішне проходження головоломки гравець отримує монетку(рис.3.9), яку зможе використати для отримання шматка картини.



Рисунок 3.7 – Локація «Атракціони»

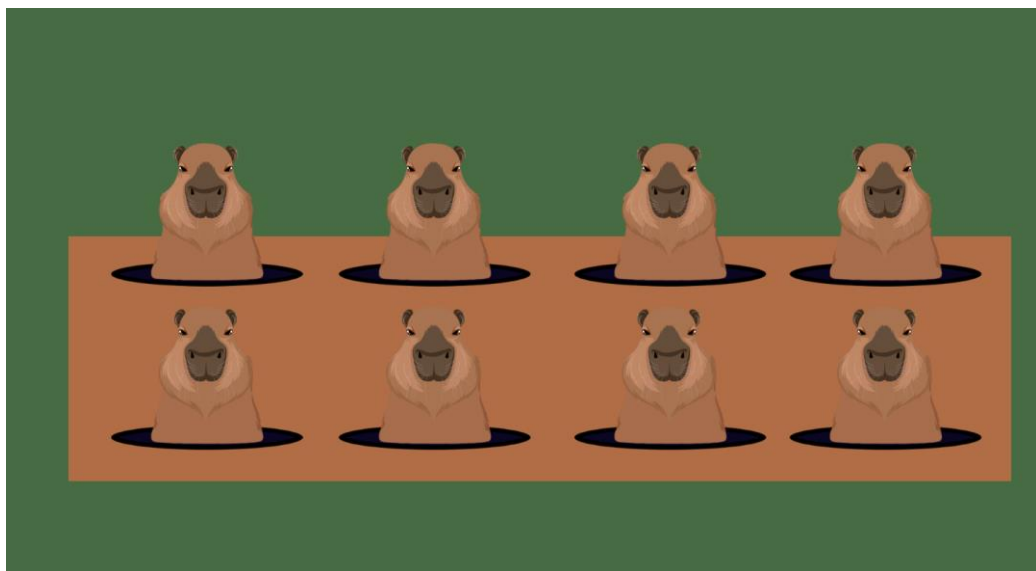


Рисунок 3.8 – Головоломка «Відгамцелити бобра»

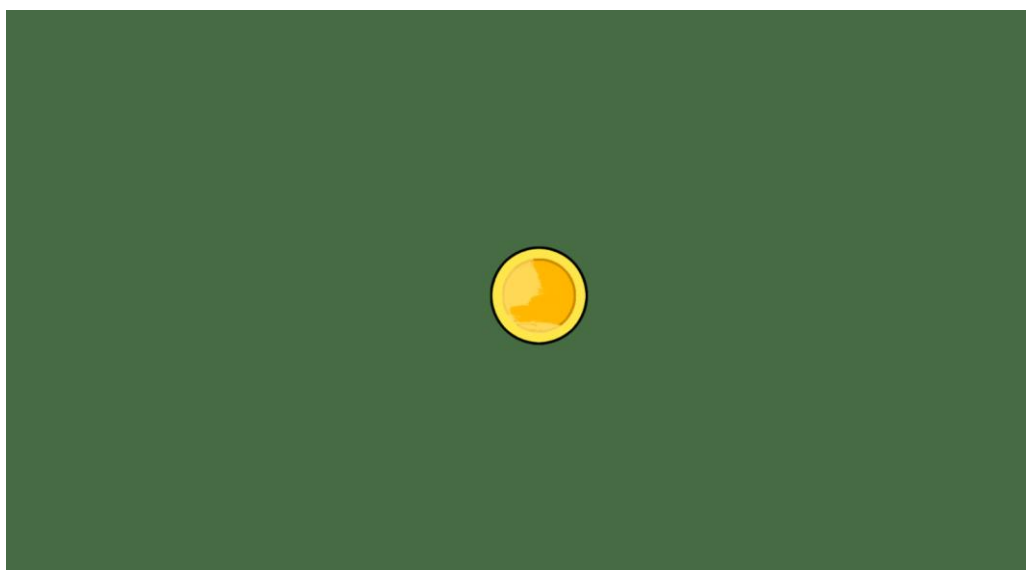


Рисунок 3.9 – Отримання монетки за успішне проходження головоломки

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

Якщо ж гравець натискає на синій автомат, то перед ним постає головоломка «Тетріс»(рис.3.10), якщо на рожевий – «Лабіринт»(рис.3.11), оранжевий – «Танці»(рис.3.12).

Після того, як користувач пройшов усі головоломки та отримав усі монетки він може натиснути на копілку та отримати шматок картини.

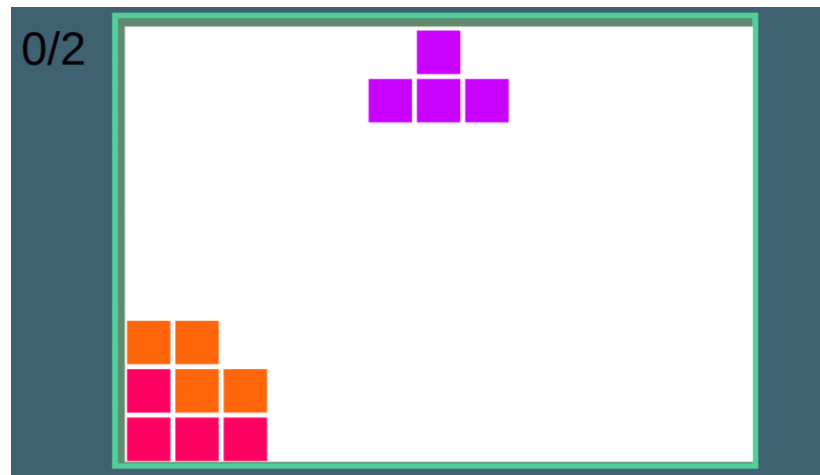


Рисунок 3.10 – Головоломка «Тетріс»

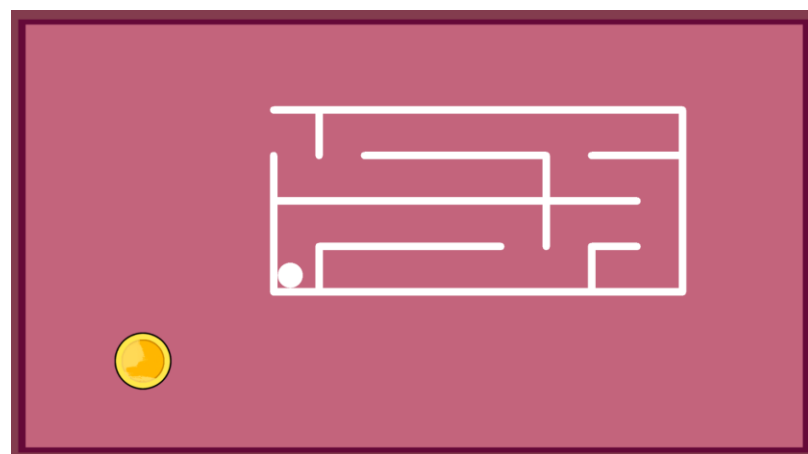


Рисунок 3.11 – Головоломка «Лабіринт»

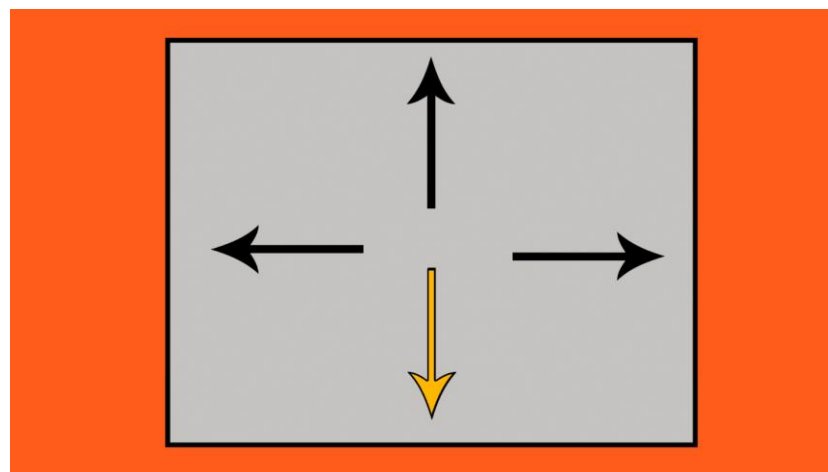


Рисунок 3.12 – Головоломка «Танці»

Якщо ж гравець відчуває, що йому необхідна допомога, то він може скористатись підказками(рис.3.13) і привид залюбки підкаже, що робити в тій чи іншій ситуації.

Також є фрази головної героїні при кліку на певні предмети, такі як: чортеня(рис.3.13), стійка з колекцією алкоголю(3.14) та стіл(3.15).



Рисунок 3.12 – Використання підказок



Рисунок 3.13 – Фраза героїні при кліку на чортеня



Рисунок 3.14 – Фраза героїні при кліку на стійку з колекцією алкоголю



Рисунок 3.15 – Фраза героїні при кліку на стіл

Також гравець може опинитись на локації під назвою «Кухня»(рис.3.16). Головна мета гравця на даній локації – зробити їжу для домовичка(рис.3.17), що тей віддав шматок картини. Для цього користувач може взаємодіяти з холодильником(рис.3.18), коміркою від шафи(рис.3.19), ножем, дошкою для нарізання(для того, щоб порізати продукти) та сковорідкою для приготування їжі.



Рисунок 3.16 – Локація «Кухня»



Рисунок 3.17 – Взаємодія з домовичком



Рисунок 3.18 – Взаємодія з холодильником

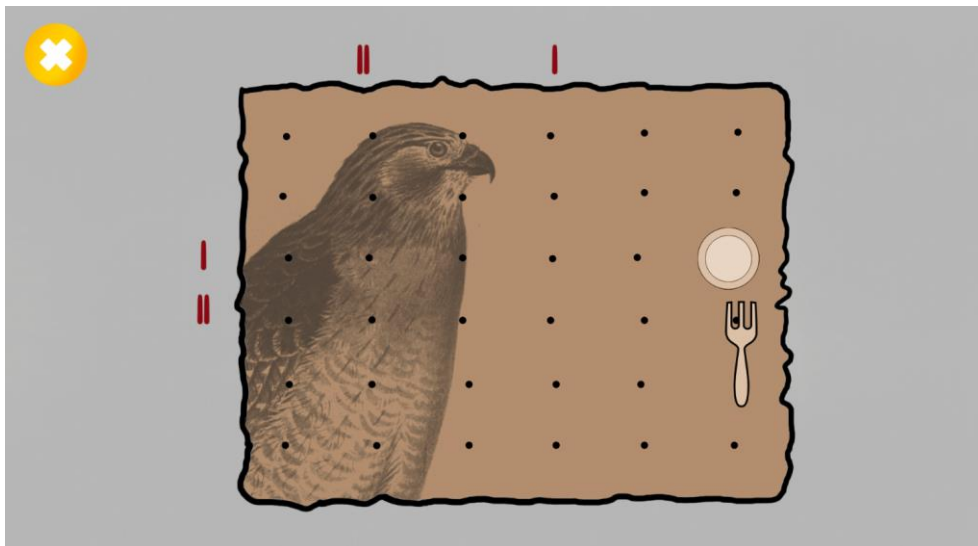


Рисунок 3.19 – Головоломка «Посуд»

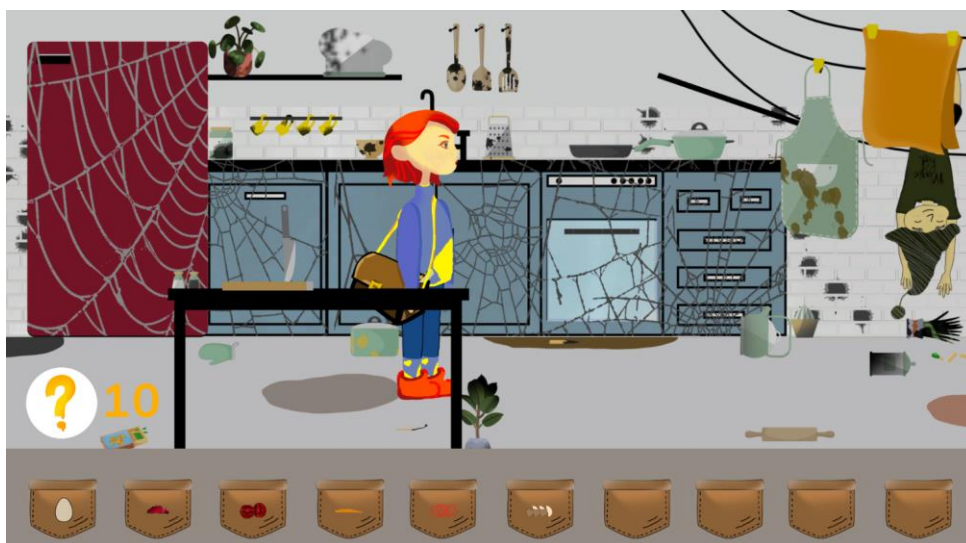


Рисунок 3.20 – Порізані продукти в інвентарі

Також при взаємодії з різними предметами, такими як: підлога(рис.3.21), посуд(рис.3.22), фартух(рис.3.23), головна героїня вимовляє різні фрази.



Рисунок 3.21 – Фраза героїні при кліку на підлогу

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49



Рисунок 3.22 – Фраза героїні при кліку на посуд



Рисунок 3.23 – Фраза героїні при кліку на фартух

3.2. Тестування роботи програмного забезпечення

Проведено тестування панелей меню а також двох локацій – «Атракціони» та «Кухня» для виявлення та усунення недоліків. Представлено пункти, які було протестовано – таблиця 3.1.

Таблиця 3.1

Меню та локації	Вхідні дані	Очікувані дані	Спостережені дані
Головне меню	Натиск кнопки «Нова гра»	Відкривається меню вибору складності гри	Як і очікувалося

	Натиск на кнопку «Продовжити гру»	Відкривається меню вибору складності збереженої гри	Як і очікувалося
	Натиск на кнопку «Налаштування»	Відкривається меню налаштувань	Як і очікувалося
	Натиск на кнопку «Вихід»	Вихід з гри	Як і очікувалося
Меню «Нова гра»	Вибір складності гри і натискання на кнопку «Почати нову гру»	Підвантаження наступних локацій з головоломками відповідно до обраної складності	Як і очікувалося
	Натиск на кнопку «Назад»	Повернення до головного меню	Як і очікувалося
Меню «Продовжити гру»	Вибір збереженої складності та натискання на кнопку «Завантаження»	Підвантаження збереженої гри	Як і очікувалося
	Натискання на кнопку «Назад»	Повернення до головного меню	Як і очікувалося
Меню «Налаштування»	Вибір мови «English»	Змінення мови усієї гри на англійську	Як і очікувалося
	Вибір мови «Українська»	Зміна мови усієї гри на українську	Як і очікувалося
	Регулювання гучності музичного супроводу	Зміна гучності музичного супроводу(тихіше або голосніше)	Як і очікувалося
	Зміна розміру екрана гри	Змінення розміру екрана відповідно до встановленого параметру	Як і очікувалося
Локація «Атракціони»	Натиск кнопки «Esc»	Відкривається меню паузи	Як і очікувалося
	Натиск на кнопку підказок	З'являється привид, який каже підказку	Як і очікувалося

	Клік на предмет, який можна підібрати	Предмет додається до інвентаря, інтерфейс оновлюється	Як і очікувалося
	Клік на предмет, який відкриває головоломку	Відкривається інтерфейс головоломки	Як і очікувалося
	Клік на предмет, який викликає фразу головної героїні	З'являється інтерфейс діалогу, де пишеться фраза, і після деякого часу зникає	Як і очікувалося
	Отримання усіх монеток за успішне проходження головоломок	З'являється шматок картини	Як і очікувалося
Локація «Атракціони», головоломка «Відлупцюй бобра»	Клік на кнопку з хрестиком	Головоломка закривається	Як і очікувалося
	Натиск на бобрів в правильному порядку	Бобри зникають, з'являється винагорода - монетка	Як і очікувалося
Локація «Атракціони», головоломка «Тетріс»	Клік на кнопку з хрестиком	Головоломка закривається	Як і очікувалося
	Заповнення фігурами 2 ряди	Успішне проходження головоломки, винагорода - монетка	Як і очікувалося
	Фігури дістають до верхнього краю монітора	Гра розпочинається спочатку	Як і очікувалося
Локація «Атракціони», головоломка «Лабіринт»	Клік на кнопку з хрестиком	Головоломка закривається	Як і очікувалося
	Проходження лабіринту	Отримання винагороди - монетка	Як і очікувалося
Локація «Атракціони»,	Клік на кнопку з хрестиком	Головоломка закривається	Як і очікувалося

головоломка «Танці»	Натискання на стрілочки в правильному порядку	Отримання винагороди - монетка	Як і очікувалося
Локація «Кухня»	Натиск кнопки «Esc»	Відкривається меню паузи	Як і очікувалося
	Натиск на кнопку підказок	З'являється привид, який каже підказку	Як і очікувалося
	Клік на предмет, який можна підібрати	Предмет додається до інвентаря, інтерфейс оновлюється	Як і очікувалося
	Клік на предмет, який відкриває головоломку	Відкривається інтерфейс головоломки	Як і очікувалося
	Клік на предмет, який викликає фразу головної героїні	З'являється інтерфейс діалогу, де пишеться фраза, і після деякого часу зникає	Як і очікувалося
	Приготування правильної їжі для домовичка	З'являється шматок картини	Як і очікувалося
Локація «Кухня», взаємодія з холодильником	Можливість підбирання продуктів	Продукти з'являються в інвентарі	Як і очікувалося
Локація «Кухня», взаємодія продуктів з дошкою для нарізання	Перетягування продуктів до дошки для нарізання	Продукти з'являються в інвентарі, але вже порізані	Як і очікувалося

При тестуванні було виявлено незначні недоліки. Насамперед, була проблема з некоректним відображенням графічних елементів. Були проблеми з розміщенням. Так, наприклад, деякі кнопки були зміщені, а текст на них був нечитабельним. Це було виправлено за допомогою правильно підбраного шрифту, а також кольорової палітри, які, використовуючись разом, доповнювали один одного.

Висновки до третього розділу

У даному розділі було проведено детальний опис роботи з програмним додатком та його тестування.

Було наведено інформацію про інтерфейс програмного додатку, його основні елементи, та порядок роботи з ним. Було описано, як користувач може взаємодіяти з програмою, які функції та можливості він має, а також які кроки потрібно виконати для досягнення певних результатів. Цей опис дозволяє користувачам ознайомитися з програмним додатком і зрозуміти, як його використовувати.

Було розглянуто процес тестування програмного додатку з метою виявлення помилок, недоліків і впевненості у його працездатності.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У кваліфікаційній роботі було проведено аналіз проблематики, методів та засобів вирішення задачі, пов'язаної з розробкою програмного продукту. Було сформульовано постановку задачі, проведено аналіз аналогів програмного продукту та обґрунтовано вибір інструментальних засобів для створення графічних елементів. Був розроблений пайплайн для виконання завдання.

У другому розділі, було проведено проектування панелей меню з використанням діаграм UML, описано основні етапи реалізації панелей головного меню. Також було розглянуто процес створення графічних матеріалів для панелей меню, включаючи створення персонажів та катсцен за допомогою Adobe Photoshop. Крім того, був розроблений текстовий документ з підказками.

У третьому розділі було описано інтерфейс та порядок роботи з розробленим програмним додатком, а також проведено тестування його функціональності.

В результаті виконаної роботи було успішно розроблено програмний продукт з відповідним інтерфейсом та функціоналом.

В додатках наведено лістинг важливих функції та класів.

Завершивши процес створення звіту, презентації та проекту, була успішно розроблена робоча програма. Ця програма демонструє надійну функціональність та безперебійну роботу, забезпечуючи виконання всіх запланованих функцій незалежно від різних дій персонажа.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Unified Modeling Language - [Електронний ресурс]. Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Unified_Modeling_Language
2. Графічні елементи та ресурси - [Електронний ресурс]. Режим доступу до ресурсу: <https://www.freepik.com/>
3. Adobe Photoshop - [Офіційний веб-сайт]. Режим доступу до ресурсу: <https://www.adobe.com/products/photoshop.html>
4. Adobe Illustrator - [Офіційний веб-сайт]. Режим доступу до ресурсу: <https://www.adobe.com/products/illustrator.html>
5. Основи роботи з Adobe Photoshop - [Електронний ресурс]. Режим доступу до ресурсу: <https://helpx.adobe.com/ua/photoshop/using/workspace-basics.html>
6. Основи роботи з Adobe Illustrator - [Електронний ресурс]. Режим доступу до ресурсу: <https://helpx.adobe.com/ua/illustrator/how-to/ai-basics-fundamentals.html>
7. Техніки інтерфейсного дизайну в Adobe Photoshop - [Електронний ресурс]. Режим доступу до ресурсу: <https://helpx.adobe.com/ua/photoshop/designers.html>
8. Ілюстрація та малюнок в Adobe Illustrator - [Електронний ресурс]. Режим доступу до ресурсу: <https://uk.soringpcrepair.com/how-to-draw-in-illustrator/>
9. Робота з шаровими ефектами в Adobe Photoshop - [Електронний ресурс]. Режим доступу до ресурсу: <https://helpx.adobe.com/ua/photoshop/using/layer-basics.html>
10. Кольорова теорія в графічному дизайні - [Електронний ресурс]. Режим доступу до ресурсу: <https://designtalk.club/teoriya-koloru-korotkyj-kurs-dlya-dyzajneriv-chastyna-1/>
11. Типографіка та композиція в графічному дизайні - [Електронний ресурс]. Режим доступу до ресурсу: <https://cases.media/article/kompoziciya-v-grafichnomu-dizaini>

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

12. Растрова та векторна графіка: основні відмінності - [Електронний ресурс]. Режим доступу до ресурсу: <https://poradumo.com.ua/231727-chim-vidrizniayetsia-rastrova-grafika-vid-vektornoyi/>
13. Принципи композиції в графічному дизайні - [Електронний ресурс]. Режим доступу до ресурсу: <https://www.arthuss.com.ua/books-blog/kompozic>
14. Техніки колірного графічного дизайну - [Електронний ресурс]. Режим доступу до ресурсу: <https://kupibo.com.ua/grafichnyj-dyzajn-robota-z-kolirnym-krugom/>
15. Інструменти та функції Adobe Photoshop для редагування зображень - [Електронний ресурс]. Режим доступу до ресурсу: <https://helpx.adobe.com/ua/photoshop/using/tools.html>
16. Як створити та налаштувати пайплайн - [Електронний ресурс]. Режим доступу до ресурсу: <https://snov.io/blog/how-to-manage-your-sales-pipeline/>
17. Практичні поради для графічного дизайнера - [Електронний ресурс]. Режим доступу до ресурсу: <https://eds.ua/blog/article/graphic-design-porady-dlya-novachkiv-ta-prosunityh>
18. Створення текстур та патернів в Adobe Photoshop - [Електронний ресурс]. Режим доступу до ресурсу: <https://helpx.adobe.com/ua/photoshop/using/creating-patterns.html>
19. Інструменти та функції Adobe Illustrator для створення векторної графіки - [Електронний ресурс]. Режим доступу до ресурсу: <https://helpx.adobe.com/ua/illustrator/using/tools-in-illustrator.html>
20. Підготовка фігури до анімації в фотошопі - [Електронний ресурс]. Режим доступу до ресурсу: <https://helpx.adobe.com/ua/photoshop/using/creating-frame-animations.html>

ДОДАТКИ

Найбільш важливі функції та класи

Файл BtnSwitchLang.cs:

```
private void Start()
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class BtnSwitchLang : MonoBehaviour
{
    [SerializeField]
    private LocalizationManager localizationManager;

    public void switchLang(string language)
    {
        localizationManager.CurrentLanguage = language;
        Debug.Log("Now language is " + localizationManager.CurrentLanguage);
    }
}
```

Файл LocalizationData.cs:

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

[System.Serializable]
public class LocalizationData
{
    public LocalizationItem[] items;
}

[System.Serializable]
public class LocalizationItem
{
    public string key;
    public string value;
}
```

Файл LocalizationManager.cs:

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

public class LocalizationManager : MonoBehaviour
{
    private string currentLanguage;
    private Dictionary<string, string> localizedText;
    public static bool isReady = false;

    public delegate void ChangeLangText();
    public event ChangeLangText OnLanguageChanged;

    void Awake()
    {
        if (!PlayerPrefs.HasKey("Language"))
```

```

    {
        if (Application.systemLanguage == SystemLanguage.Ukrainian)
        {
            PlayerPrefs.SetString("Language", "Ukrainian");
        }
        else
        {
            PlayerPrefs.SetString("Language", "English");
        }
    }
    currentLanguage = PlayerPrefs.GetString("Language");

    LoadLocalizedText(currentLanguage);
}

public void LoadLocalizedText(string langName)
{
    string path = Application.streamingAssetsPath + "/Languages/" +
langName + ".json";

    string dataAsJson;

    if (Application.platform == RuntimePlatform.Android)
    {
        WWW reader = new WWW(path);
        while (!reader.isDone) { }

        dataAsJson = reader.text;
    }
    else
    {
        dataAsJson = File.ReadAllText(path);
    }

    LocalizationData loadedData =
JsonUtility.FromJson<LocalizationData>(dataAsJson);

    localizedText = new Dictionary<string, string>();
    for (int i = 0; i < loadedData.items.Length; i++)
    {
        localizedText.Add(loadedData.items[i].key,
loadedData.items[i].value);
    }

    PlayerPrefs.SetString("Language", langName);
    isReady = true;
    OnLanguageChanged?.Invoke();
}

public string GetLocalizedValue(string key)
{
    if (localizedText.ContainsKey(key))
    {
        return localizedText[key];
    }
    else

```

```

        {
            throw new Exception("Localized text with key \"" + key + "\" not found");
        }
    }

    public string CurrentLanguage
    {
        get
        {
            return currentLanguage;
        }
        set
        {
            PlayerPrefs.SetString("Language", value);
            currentLanguage = PlayerPrefs.GetString("Language");
            LoadLocalizedText(currentLanguage);
        }
    }

    public bool IsReady
    {
        get
        {
            return isReady;
        }
    }
}

```

Файл LocalizedSprite.cs:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class LocalizedSprite : MonoBehaviour
{
    [SerializeField] GameObject ukrainianSprite;
    [SerializeField] GameObject englishSprite;

    private LocalizationManager localizationManager;

    void Awake()
    {
        if (localizationManager == null)
        {
            localizationManager =
                GameObject.FindObjectOfType<LocalizationManager>().GetComponent<LocalizationManager>();
        }

        localizationManager.OnLanguageChanged += UpdateText;
    }

    void Start()
    {
        UpdateText();
    }
}

```

```

    }

    private void OnDestroy()
    {
        localizationManager.OnLanguageChanged -= UpdateText;
    }

    virtual public void UpdateText()
    {
        if (gameObject == null) return;

        if (localizationManager == null)
        {
            localizationManager =
                GameObject.FindGameObjectWithTag("LocalizationManager").GetComponent<LocalizationManager>();
        }

        string currentLanguage = PlayerPrefs.GetString("Language");

        ukrainianSprite.SetActive(false);
        englishSprite.SetActive(false);

        if (currentLanguage == "Ukrainian")
        {
            ukrainianSprite.SetActive(true);
        }
        else if (currentLanguage == "English")
        {
            englishSprite.SetActive(true);
        }
    }
}

```

Файл LocalizedText.cs:

```

using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.UI;

public class LocalizedText : MonoBehaviour
{
    [SerializeField]
    public string key;

    private LocalizationManager localizationManager;
    private Text text;
    private TMP_Text textTMP;

    void Awake()
    {
        if (localizationManager == null)
        {

```

```

        localizationManager =
GameObject.FindObjectOfType<LocalizationManager>().GetComponent<LocalizationMa
nager>();
    }
    if (text == null)
    {
        if (GetComponent<Text>())
            text = GetComponent<Text>();
        if (GetComponent<TMP_Text>())
            textTMP = GetComponent<TMP_Text>();
    }
    localizationManager.OnLanguageChanged += UpdateText;
}

void Start()
{
    UpdateText();
}

private void OnDestroy()
{
    localizationManager.OnLanguageChanged -= UpdateText;
}

virtual public void UpdateText()
{
    if (gameObject == null) return;

    if (localizationManager == null)
    {
        localizationManager =
GameObject.FindGameObjectWithTag("LocalizationManager").GetComponent<Localizat
ionManager>();
    }
    if (text == null)
    {
        if (GetComponent<Text>())
            text = GetComponent<Text>();
        if (GetComponent<TMP_Text>())
            textTMP = GetComponent<TMP_Text>();
    }
    if (textTMP)
        textTMP.text = localizationManager.GetLocalizedValue(key);
    else if (text)
        text.text = localizationManager.GetLocalizedValue(key);
    }
}

```

Файл LoadGameToggles.cs:

```

using System.Collections;
using System.Collections.Generic;
using System.IO;
using TMPro;
using UnityEngine;
using UnityEngine.UI;

```

```

public class LoadGameToggles : MonoBehaviour
{
    [SerializeField]
    private Toggle toggleEasy;

    [SerializeField]
    private Toggle toggleNormal;

    [SerializeField]
    private Toggle toggleHard;

    [SerializeField]
    private TMP_Text noSavedGame;

    [SerializeField]
    private Button loadButton;

    private void Start()
    {
        string dirEasy = Application.persistentDataPath + "/Easy";
        string dirNormal = Application.persistentDataPath + "/Normal";
        string dirHard = Application.persistentDataPath + "/Hard";

        if (!Directory.Exists(dirEasy))
        {
            toggleEasy.gameObject.SetActive(false);
            toggleEasy.isOn = false;
        }
        if (!Directory.Exists(dirNormal))
        {
            toggleNormal.gameObject.SetActive(false);
            toggleNormal.isOn = false;
        }
        if (!Directory.Exists(dirHard))
        {
            toggleHard.gameObject.SetActive(false);
            toggleHard.isOn = false;
        }

        if(! (Directory.Exists(dirEasy) || Directory.Exists(dirNormal) ||
Directory.Exists(dirHard)) )
        {
            noSavedGame.gameObject.SetActive(true);
            loadButton.gameObject.SetActive(false);
        }
    }
}

```

Файл PauseMenu.cs:

```

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

public class PauseMenu : MonoBehaviour
{

```

```

    public bool GameIsPaused = false;
    public bool GameCanBePaused = true;

    public GameObject pauseMenuUI;
    public GameObject[] objects;

    public void SetCanBePaused(bool newBool)
    {
        GameCanBePaused = newBool;
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Escape))
        {
            if (GameIsPaused)
            {
                Debug.Log("Resume");
                Resume();
            }
            else if (GameCanBePaused)
            {
                Pause();
            }
        }
    }

    public void Resume()
    {
        GameIsPaused = false;
        pauseMenuUI.SetActive(false);
        foreach (GameObject obj in objects)
            obj.SetActive(true);
        Time.timeScale = 1f;
    }

    void Pause()
    {
        pauseMenuUI.SetActive(true);
        foreach (GameObject obj in objects)
            obj.SetActive(false);
        Time.timeScale = 0f;
        GameIsPaused = true;
    }

    public void LoadMenu()
    {
        Resume();
        SceneManager.LoadScene(0);
    }
}

```

Файл Settings.cs:

```

using System;
using System.Collections;
using System.Collections.Generic;

```

```

using TMPro;
using UnityEngine;
using UnityEngine.Audio;
using UnityEngine.UI;
using System.Linq;

public class Settings : MonoBehaviour
{
    public Toggle FullscreenToggle;
    public Slider VolumeSlider;

    public void Awake()
    {
        SetAllSettings();
    }

    public void SetAllSettings()
    {
        if (PlayerPrefs.HasKey("Volume"))
        {
            Debug.Log("Set volume");
            VolumeSlider.value = PlayerPrefs.GetFloat("Volume");
        }

        if (PlayerPrefs.HasKey("isFullscreen"))
        {
            FullscreenToggle.isOn = (PlayerPrefs.GetInt("isFullscreen") == 1);
        }
    }

    public void SetFullscreen(bool isFullscreen)
    {
        Screen.fullScreen = isFullscreen;
        if (isFullscreen)
            PlayerPrefs.SetInt("isFullscreen", 1);
        else
        {
            PlayerPrefs.SetInt("isFullscreen", 0);
        }
    }

    public void SetVolume(float volume)
    {
        AudioListener.volume = volume;
        PlayerPrefs.SetFloat("Volume", volume);
    }
}

```

Файл StartNewGame.cs:

```

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

public class StartNewGame : MonoBehaviour
{

```

```
private string difficulty = "Easy";

public void SetDifficulty(string name)
{
    difficulty = name;
    Debug.Log(difficulty);
}

public void NewGame(int sceneId)
{
    PlayerPrefs.SetString("Difficulty", difficulty);

    LoadScene(sceneId);
    PlayerPrefs.SetInt("SaveGame", sceneId);
}

public void LoadScene(int sceneNumber)
{
    SceneManager.LoadScene(sceneNumber);
}
}
```