

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 3D гри з використанням методу імітації
«порожнього стільця»**

Виконав студент групи КН-19-1
ШМІДТ Даніїл Олександрович

Керівник роботи:
ГРАФ Марина Сергіївна

Рецензент:
СУГОНЯК Інна Іванівна

Житомир – 2023

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

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: **Шмідт Данііл Олександрович**

Керівник роботи: **Граф Марина Сергіївна**

Тема роботи: **«Розробка 3D гри з використанням методу імітації «порожнього стільця»»,**
затверджена наказом закладу вищої освіти від **«05» січня 2023 р., №10/с**

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

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

Об'єктом дослідження є розробка 3D гри з використанням методу «порожнього стільця».

Предметом дослідження є використання психологічних методів в розробки ігор на Unity.

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

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв
1	М.С. Граф	01.05.23	01.05.23
2	М.С. Граф	12.05.23	12.05.23
3	М.С. Граф	23.05.23	23.05.23

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формулювання технічного завдання. Опрацювання літературних джерел.	2 лютого 2023 – 5 березня 2023	Виконано
2	Проектування структури системи	6 березня 2023 – 25 березня 2023	Виконано
3	Написання програмного коду	26 березня 2023 – 26 квітня 2023	Виконано
4	Тестування системи	27 квітня 2023 – 14 травня 2023	Виконано
5	Оформлення пояснювальної записки	15 травня 2023 – 1 червня 2023	Виконано
6	Попередній захист	1 червня 2023	Виконано

Здобувач вищої освіти **Данііл Шмідт**
Керівник **Марина ГРАФ**

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему «Розробка 3D гри з використанням методу імітації «порожнього стільця» складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 54 сторінках друкованого тексту. Пояснювальна записка має 2 сторінки додатків.

Список використаних джерел містить 12 найменування і займає 1 сторінки. В роботі наведено 26 рисунки. Загальний обсяг роботи – 58 сторінок.

У першому розділі були визначені основні вимоги та критерії до проекту, були оглянуті альтернативні ігрові розробки, було обрано стек технологій для розробки.

У другому розділі було проведено проектування гри з використанням психологічних методів, була розроблена система редагування діалогів.

У третьому розділі був розроблений користувацький інтерфейс, було побудована гра, був описаний порядок використання гри, були протестовані основні функції гри.

Висновок містить в собі результати виконаної роботи по розробці гри з використанням методу імітації «порожнього стільця».

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

ABSTRACT

The explanatory note to the qualification work on the topic "Development of a 3D game using the method of simulating an "empty chair" consists of a list of abbreviations, an introduction, three chapters, conclusions, a list of references and an appendix.

The text part is set out on 54 pages of printed text. The explanatory note has 2 pages of appendices.

The list of references includes 12 titles and occupies 1 page. The paper contains 26 figures. The total volume of the work is 58 pages.

In the first chapter, the main requirements and criteria for the project were defined, alternative game developments were reviewed, and a technology stack was selected for development.

In the second chapter, the game was designed using psychological methods, and a dialog editing system was developed.

In the third section, the user interface was developed, the game was built, the procedure for using the game was described, and the main functions of the game were tested.

The conclusion contains the results of the work done to develop the game using the "empty chair" simulation method.

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

ЗМІСТ

РЕФЕРАТ	Ошибка! Закладка не определена.
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ВИКОРИСТАННЯ ТЕХНІКИ «ПОРОЖНЬОГО СТІЛЬЦЯ»	10
1.1. Постановка задачі	10
1.2. Аналіз аналогів програмного продукту.....	11
1.3. Вибір архітектури гри.	16
Висновки до розділу 1	21
РОЗДІЛ 2. ПРОЕКТУВАННЯ ГРИ З ПСИХОЛОГІЧНОЮ ТЕХНІКОЮ «ПОРОЖНЬОГО СТІЛЬЦЯ»	23
2.1. Визначення варіантів використання та структури системи.	23
2.2. Розробка функціональних алгоритмів роботи програми.....	33
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ СИМУЛЯЦІЇ	45
3.2. Структура інтерфейсу. Інтерфейс та порядок роботи з грою.	45
3.3. Тестування роботи системи	50
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТКИ.....	56

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

U3D – Unity3D.

ОС – Операційна система.

ЕТ – Електричний транспорт.

ТЕД – Тяговий електродвигун.

АХ – Автономний хід.

КМ – Контактна мережа (дроти живлення).

GUI – Graphic User Interface – графічний інтерфейс користувача.

ECS – Entity-Component-System – Сутність-Компонент-Система.

UE4 – Unreal Engine 4.

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

ВСТУП

Основний зміст роботи викладено на 58 сторінках тексту, містить 11 таблиць, 37 ілюстрацій, 2 додатків, налічує 12 найменувань літературних джерел.

Метою кваліфікаційної роботи є створення гри в жанрі головоломки за допомогою рушія Unity з використанням психологічної техніки "порожнього стільця", що дає можливість визначити напрямки поведінки ігрових персонажів.

Абсолютно кожен із нас стикався з ситуацією, коли наші власні думки "схеми" могли заважати нам досягати поставлених цілей у житті. Наша поведінка - це немов типові схеми, за якими рухаються наші думки і на які ми постійно повертаємося.

Для того, щоб змінити звичну поведінку, необхідно усвідомити, як склалося наше звичне мислення і чому ми звикли обирати саме його. З цього можна зробити висновок, що метою було створення застосунку, який допоможе в ігровій формі визначити, які схеми вмикаються в тій чи іншій ситуації та як можна розвинути нові схеми. Для досягнення цього зазвичай використовують одні з таких підходів:

Вправи на основі образів. Методи візуалізації та керованої уяви включають у себе створення уявних образів або сценаріїв для вивчення емоцій, спогадів або бажаних результатів.

Голосовий діалог. Голосовий діалог - це метод, за якого люди беруть участь у розмові між різними аспектами або "голосами" всередині себе. Це може допомогти зрозуміти різні частини особистості та отримати уявлення про суперечливі бажання або мотиви.

Техніка порожнього стільця. Порожній стілець - це техніка, яка передбачає використання порожнього стільця для представлення людини або важливого аспекту себе, з яким у людини є невирішені почуття, конфлікти

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

або незавершені справи. Техніка спрямована на полегшення емоційного вираження, самосвідомості та інтеграції суперечливих емоцій або аспектів особистості. Головною технікою для гри було обрано техніку порожнього стільця.

Метою випускної роботи є проектування та розробка ігри з використанням техніки «порожнього стільця» на рушії Unity.

Встановлена мета обумовлює наступні завдання:

- проведення аналізу роботи з технікою «порожнього стільця»;
- визначення архітектури та загального принципу роботи системи;
- обґрунтування та вибір варіанту реалізації;
- розробка інформаційної системи;
- проектування структурних складових та алгоритмів роботи системи;

Актуальність теми: в іграх завжди використовували психологічні методи для захоплення уваги гравця й отримання певних емоцій від геймплею, але психологічні методи можна використовувати не тільки для таких цілей. Використання психологічних методів у явному вигляді дасть змогу створювати ігрові умовності для створення безпечної атмосфери, які сприяють зниженню стресу, розслабленню. Тому темою дослідження було обрано психологічний метод “порожнього стільця”, призначення якого заохочення позитивних психологічних станів, самоаналіз та емоційна турбота про себе.

Об’єктом дослідження є використання психологічних технік в ігрових формах, а саме техніки «порожнього стільця».

Предмет дослідження є використання психологічних технік під час створення гри та вплив на самого гравця.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Мета дослідження: використання психологічних методик при створенні ігор та їхній вплив на гравця.

Практичне значення полягає в тому, щоб допомогти людині усвідомити, які схеми виявляються в неї і познайомити з тим, як можна розвинути нові схеми поведінки.

Пояснювальна записка до випускної роботи містить 58 сторінках тексту, містить 11 таблиць, 37 ілюстрацій, 2 додатків, налічує 12 найменувань.

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

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ВИКОРИСТАННЯ ТЕХНІКИ «ПОРОЖНЬОГО СТІЛЬЦЯ»

1.1. Постановка задачі

Основне завдання це розробити гру з психологічною технікою "порожнього стільця". Основна задумка гри, полягає в тому, щоб познайомити з принципами роботи техніки. Використання такої гри допоможе поліпшити розуміння основних схем поведінки і наочно відобразити, можливі інші схеми поведінки на які варто звернути увагу; допоможе в навчанні техніці "порожнього стільця" новим психологам. Гра допоможе створювати нові ситуації без шкоди для людини за рахунок того, що вона зможе прожити ситуацію не в реальності.

У випускній роботі необхідно створити гру у вигляді головоломки з використанням психологічної техніки. Максимально точно продумати ігрову ситуацію, а також пов'язані з цим діалогову систему наближену до реальності. Для реалізації поставлених завдань, основними етапами є:

1. Розробити дизайн користувацького інтерфейсу.

1. Створити загальну концепцію дизайну користувацького інтерфейсу
2. За допомогою графічного редактора Adobe Photoshop створити графічні зображення основних елементів. (кнопок)
3. Створити основні сцени гри (головне меню, паузу, збереження та видалення гри, відображення діалогової системи)

2. Проектування та розробка основних ігрових модулів.

1. Модуль керування персонажем.
2. Модуль керування головним меню
3. Модуль керування діалоговою системою

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

4. Модуль керування кат-сценами

3. Моделювання 3D об'єктів за допомогою Blender

1. Меблів

2. Будівель

3. Персонажів

Результатом реалізації поставленого завдання є гра з використання психологічної техніки «порожнього стільця».

1.2. Аналіз аналогів програмного продукту

Психологічні методи ефективно використовувалися відеоіграми протягом багатьох років просто тому, що всі ми живемо в одному й тому самому світі та декодуємо наше оточення, використовуючи здебільшого одні й ті самі фізичні та психічні механізми. Наш життєвий досвід приводить нас у зіткнення з аналогічним діапазоном емоцій, і саме на цю основу ми спираємося, коли створюємо щось, чи то книжка, чи то пісня, чи то гра. Психологічні методи можуть використовуватися з такою метою:

1. Навчання.

2. Терапії.

3. Розважальних цілей.

Переважно психологічні методи відкрито використовують у навчальних іграх, а також у терапіях, коли людині важко відповідати на запитання прямо. У сучасних іграх психологічні методи стали приховувати за текстовими повідомленнями, а також маніпуляцією з ігровою сценою. У випадку з текстовими повідомленнями це відбувається завдяки філософським міркуванням і риторичним запитанням на які гравець відповідає за допомогою механік гри. Якщо ж розглядати маніпуляцію зі сценою, то здебільшого використовуються ігри з світлом, звуковим супроводом, а також побудовою самої локації. На цей час більшість ігор

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

використовують кілька підходів для досягнення цілей. Як основним показником використання текстового посилання з філософським підтекстом можна виділити таку гру як "The Talos Principle". Також є й окремі ігри, які передають емоційний стан головного героя через зміни сприйняття самого рівня через зміну чіткості картинки, а також яскравість освітлення. Прикладом такої гри є серія ігор "Amnesia". В рамках дипломного проекту за основу брали ігри першої категорії, тобто ті, які використовують комбінований підхід, а також відображення методики через діалогові повідомлення. Порівняння схожих ігор можна побачити на порівняльній таблиці 1.1.

Таблиця 1.1.

Назва	Фізичний рушій	Платформа	Складність механік	Графічне оформлення
The Talos Principle	Добре	Windows	Легка	Добре
Syberia	Добре	Windows	Легка	Добре
amnesia the dark descent	Добре	Windows	Середня	Добре



Рисунок.1.1. The Talos Principle

The Talos Principle – це філософська головоломка з видом від першої особи від студії Croteam. В основі геймплею полягає продуманий сюжет, який регулярно ставить питання, на які гравець зобов'язаний відповідати,

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

розбавляючи геймплей легкими головоломками. Користуючись простим візуалом та легких механік дозволяє непідготовленому гравцю досить швидко розібратися з усіма основними механіками. Механіки можна поділити на два види інструментів, один з яких блокує перепади та другий, який дозволяє направляти джерела світла в необхідні точки підключення. Сюжет подається трьома способами.

Перший з яких – це спілкування зі штучним інтелектом мета якого є заплутати гравця про усвідомлення ігрового світу.

Друга полягає у пошуку прихованих аудіо повідомлень, які розкривають філософське посилення гри.

Третій це залишені QR-коди з текстовими питаннями на які гравець може відповісти та наступного разу проходячи цю ж локацію побачити відповідь на повідомлення.

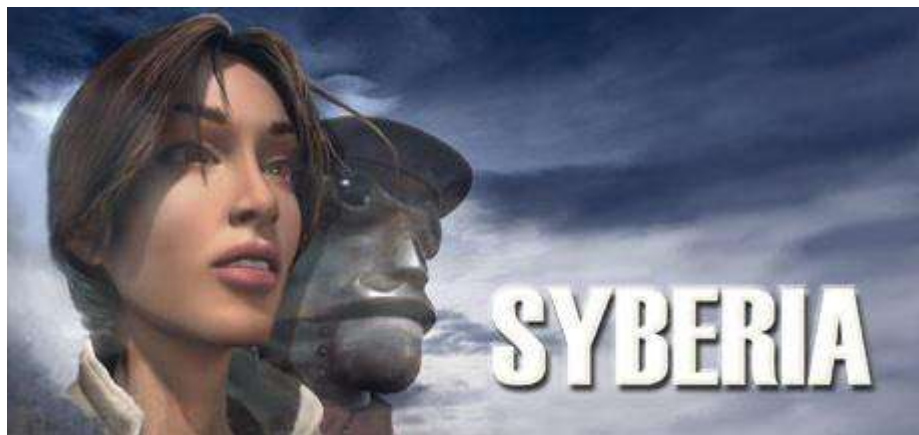


Рисунок.1.2. Syberia

Syberia - пригодницька комп'ютерна гра, створена бельгійським автором Бенуа Сокалем та видана компанією Microïds у 2002 році. В основі гри лежать головоломки, тісно пов'язані з основним сюжетом. Внаслідок того, що гра є представником жанру point-and-click, у неї досить просто почати грати шляхом продуманого візуального інтерфейсу. Предмети, з якими можна взаємодіяти змінюють курсор на різні іконки такі як: лупа, рука, хмара повідомлення. Сюжетне ознайомлення відбувається шляхом

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

діалогової системи, а також деяких записок, які можна отримати, вирішуючи головоломки. Головоломки ж своєю чергою діляться на:

Статичні, тобто ті, які являють собою невеликі міні ігри такі як пазл і, так далі.

Рухомі, які заховані в різних місцях локацій, які можна отримати, відповівши вірно на питання NPC.

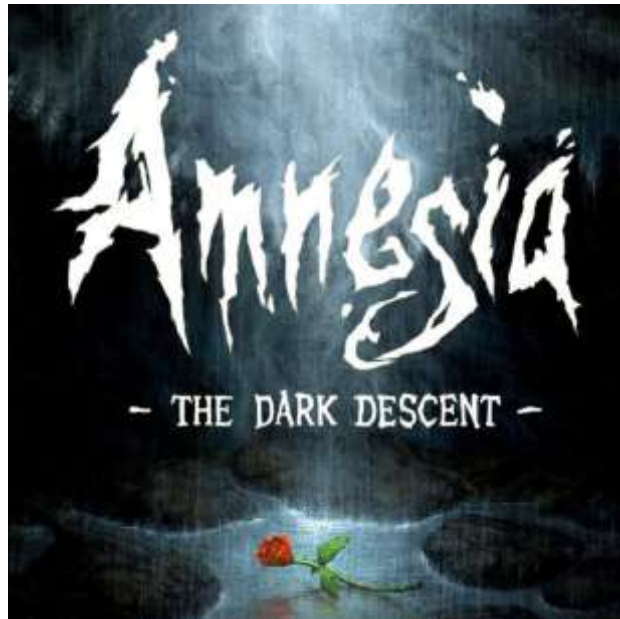


Рисунок.1.3. Amnesia: The Dark Descent

Amnesia: The Dark Descent - це хорор від першої особи на виживання. Це гра про вивчення, життя та занурення в атмосферу нічного кошмару. В основі гри лежать прості головоломки спрямовану на механіку взаємодії зі світом, що дає можливості взаємодії з предметами.

Загалом взаємодії полягає у піднятті та перетягуванні предметів, але візуальний концепт гри змінюється по ходу сюжету. Тобто при знаходженні сюжетних предметів або зустрічі з ворожими NPC сприйняття гравця починає змінюватися внаслідок того, що кольорова палітра гри змінюється. Сюжет доноситься кількома способами:

Перший і досить класичний через записки, залишені у різних місцях на локаціях.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Другий – це внутрішній діалог головного героя самим собою.

Таблиця 1.2. Порівняння ігор на основі того які можливості вони дають.

Назва	Психологічне занурення	Емоційний досвід	Рівень складності головоломок
The Talos Principle	Добрий	Добрий	Добрий
Amnesia: The Dark Descent	Високе	Високий	Низький
Syberia	Низький	Добрий	Добрий

Таблиця 1.3.Мінімальні системні вимоги порівнюваних аналогів

Назва	Operating System	Processor	RAM	Місце на диску
The Talos Principle	Windows 7/8/10	Dual-core 2.0 GHz	2 GB RAM	8 GB
Amnesia: The Dark Descent	Windows XP/Vista/7/8/10	2.0 GHz Intel Core i3 or AMD A6	2 GB RAM	3 GB
Syberia	Windows XP/Vista/7/8	1 GHz Processor	512 MB RAM	1.5 GB

Порівнявши та проаналізувавши ці три гри, можна з упевненістю сказати про те, що кожен із підходів може отримувати різний вплив на емоційний стан гравця незалежно від рівня складності головоломок.

Навіть якщо взяти до уваги, що ігри, які ми порівнюємо, знаходяться в одному напрямку, мають різну реалізацію, а також візуальний і концептуальний підхід для головоломок, що своєю чергою дозволяє визначити, які моделі найкраще підходять для розкриття тих чи інших аспектів.

Аналіз цих ігор показав, що для майбутнього проєкта найкраще підійде комбінування діалогової системи, яка прямо взаємодіє з гравцем, а також головоломок, пов'язаних із сюжетною лінією.

Таким чином нам необхідно продумати сюжет, який буде підходи максимально до реальної історії, візуальна складова має імітувати реальне життя, а також давати можливість самому вибирати які підходи для вирішення

головоломок має обрати гравець. Які додаткові моменти можна продумати, необов'язкові головоломки або додаткових персонажів, які допомагатимуть гравцеві в пошуку рішень для тих чи інших завдань.

1.3. Вибір архітектури гри.

Для створення невеликої гри здебільшого використовують такі патерни:

1. Singleton
2. Observer
3. фабрика
4. State

Аналізуючи патерни, які використовують у подібних проєктах, можемо дійти висновку, що патерн Observer найкраще використовувати з елементами, що пов'язані з обробкою подій, приміром, з поданням користувацького інтерфейсу.

У випадку з патерном Factory його використовують, коли необхідно динамічно створювати об'єкти, як-от процедурна генерація локації або ж створення персонажів, завдяки тому, що в проєкті не планується створення локації способом генерації, а гра має невелику кількість персонажів, додавання цього патерну може спричинити навантаження в розумінні проєкту.

Паттерн State - найкраще підійде, коли необхідно прямо відстежувати стан об'єктів. За рахунок того, що в грі планується діалогова система, в якій необхідно буде вибирати різні варіанти відповідей, такий підхід може полегшити розробку.

Оптимальним варіантом для розроблення було обрано два патерни: перший - це Singleton, а також патерн State.

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

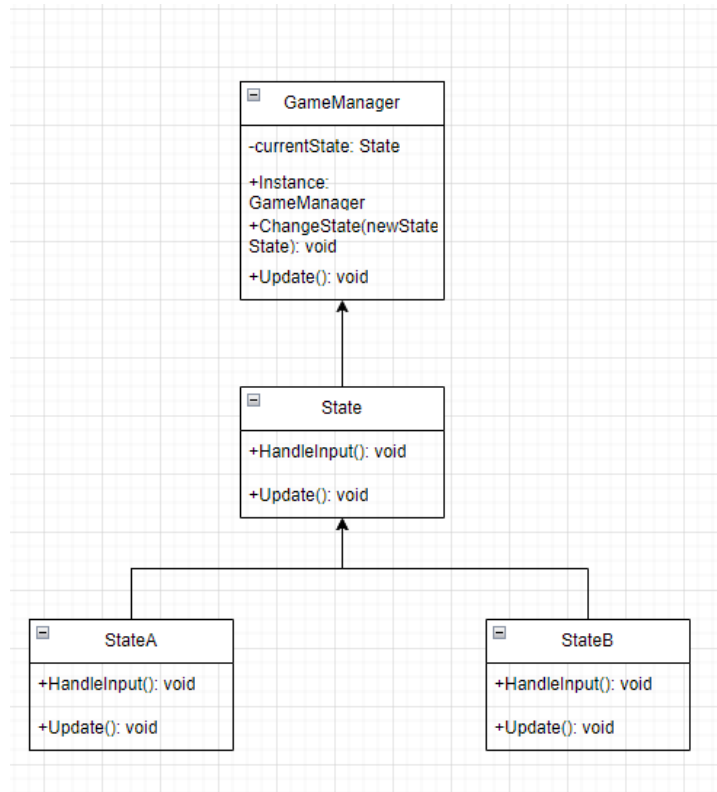


Рисунок.1.4. Схема об'єднання патерну синглтон та стану

GameManager - написаний на Singleton, який у свою чергу відповідає за станом гри.

"currentState" - відповідає змінній, в якій зберігається стан гри.

"Instance" - статична властивість, що відкриває, можливість глобально керувати шаблоном.

"ChangeState(newState: State)" - метод, який дозволяє змінювати стан об'єкта.

"HandleInput()" - тригер обробки подій.

"StateA" та "StateB" - реалізація різних станів гри.

За рахунок того, що патерн Одиночки можна використовувати для важливих компонентів, таких як основний скрипт діалогової системи. Паттерн же Станів буде доповнювати одиночку за рахунок того, що дає можливість відстежувати зміни вибору.

Якщо ж аналізувати які переваги може дати комбінування двох патернів, таких як Singleton і State, то можна виділити такі аспекти:

1. Центральне управління компонентами. Завдяки Singleton, є можливість створити один клас, який буде глобально доступний і при цьому гарантувати, що він буде створений лише в одному екземплярі. А за рахунок додаткового використання станів це дає можливість спростити переходи між станом об'єктів.

2. Гнучкість. За рахунок патерну Станів ми можемо не задіяти основний скрипт, що написано на Одиночці, і водночас розширювати варіанти реакції на певні події завдяки тому, що патерн Стану закриває всередині себе поведінку на певні тригери.

3. визначеність переходів. За рахунок того, що в Одиночці ми можемо задати правила, які даватимуть інструкції впливу для патерну Стану щодо того, на які події він має реагувати. Такий підхід може поліпшити організованість коду.

Якщо ж підбивати підсумки, то для такого проєкту, в якому основним вузлом є діалогова система, використання двох таких патернів може поліпшити структуру і зручність масштабованості коду.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення.

Наразі рушіями, які активно використовуються для розроблення ігор сьогодні є: Unity, Unreal, GameMaker: Studio, RPGMaker. У кожному з них є свої сильні та слабкі сторони.

Порівнюючи рушії з тим, що необхідно реалізувати в проєкті, виходить такий результат:

1. Unreal - дасть змогу реалізувати чудову візуальну складову завдяки безплатному і готових ассетів, а й через це ж виносить більші вимоги до персональних комп'ютерів як під час розроблення, так і в готовому проєкті.

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

2. GameMaker: Studio - інструмент, який призначений здебільшого для створення 2D-ігор, що дає доволі швидко та зручну систему керування, але вибивається з вимог до проєкту завдяки тому, що візуальне оформлення виходить за рамки 2D-ігор.

3. RPGMaker - чудовий інструмент для створення великих історій внаслідок великої бази готових рішень. Рушій дає змогу розробити випадковий генератор персонажів і редактора карт, але завдяки тому, що більшість асетів і оптимізацій розраховані на ігри в основному 2D з видом зверху або ж графічних новел, що зі свого боку вибивається із завдань проєкту.

4. Unity - кросплатформний рушій, який дає змогу створювати додатки, що працюють на більш ніж 25 різних платформах. Завдяки великій бібліотеці асетів, а також докладній документації дає змогу досить швидко й ефективно розробляти будь-який застосунок. Так само внаслідок невеликих системних вимог додаток легше оптимізувати на велику кількість пристроїв.

Підбиваючи підсумки вибору рушія, найбільше підходить Unity внаслідок таких переваг:

1. Розробка ведеться мовою C#. Внаслідок того, що C# досить давно використовується великою кількістю програмістів, він набрав базу вже готових рішень, які пройшли перевірку часом.

2. Кросплатформеність. Дає змогу використовувати вже створений застосунок на інших пристроях із мінімальними змінами коду та структури застосунку.

3 Бібліотека Unity. За рахунок детальної бібліотеки з інформацією про кожен метод та модуль в Unity з прикладами та порадами щодо того, як використовувати певні структури, дає змогу пришвидшити не лише розуміння цього рушія, а й полегшити подальшу розробку та підтримку.

4. Спільнота. В Unity досить активна спільнота, яка регулярно відповідає на безліч тем на форумах, а також ділитися практичними порадами за допомогою навчальних відеороликів із прикладами про те, як можна реалізувати практично

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

будь-який функціонал. Також працює служба підтримки, яка в разі потреби може допомогти.



Рисунок. 1.5. логотип Blender

Для створення моделей і персонажів, а також текстурування об'єктів було обрано програму Blender. Blender - це безкоштовне програмне забезпечення з відкритим вихідним кодом, що має серйозну потужність. Ба більше, Blender регулярно оновлюється, щоб залишатися технологічно актуальним і мати можливість створювати високоякісні творіння. Завдяки великому набору інструментів для моделювання дає змогу прискорити створення, перетворення і редагування як власних моделей, так і сторонніх асетів.

Окремою перевагою необхідно і підкреслити, можливості для анімації та ріггінгу. Це відбувається завдяки перетворенню скелета та автоматичні прив'язки кісток до моделі персонажа. Також завдяки зручному редактору для налаштування ваги для кісток за допомогою кривих можна отримати більш плавну або ж реалістичну поведінку персонажа.

Завдяки тому, що моделі з блендера можна перетворити у формату obj, який підтримується в Unity, то створення і впровадження в проєкт не викликає жодних труднощів.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок. 1.6. логотип Adobe Photoshop

Для створення візуальних елементів інтерфейсу було вирішено вибрати програму Adobe photoshop. Внаслідок того, що в ній дається широкий спектр можливостей для реалізації будь-яких графічних елементів. Внаслідок великого набору інструментів, які дають можливість маніпулювати формами, кольорами, текстурами. Також внаслідок того, що в Photoshop використовується система шарів, можливо створити складні елементи завдяки об'єднанню декількох зображень в одне. І однією з найсильніших переваг є те, що створені в програмі зображення легко інтегрувати з іншими програмами, що полегшує створення візуальних елементів для інтерфейсів.

Якщо підбивати підсумок, то програми, які будуть використовуватися для реалізації поставлених цілей наступні. Unity 2022.3.0f1, середовище розробки Visual Studio Code, графічний редактор Adobe Photoshop, 3D-редактор Blender

Висновки до розділу 1

Виконане вивчення наявних аналогів ігор, які використовували психологічні моделі всередині себе, дало змогу визначити необхідні аспекти, які необхідно реалізувати в грі. Аналіз сильних і слабких сторін кожного проєкту показав, що головним функціоналом має бути: продуманий інтерфейс,

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

продумана діалогова система, що міститиме в собі можливість вибору варіантів відповідей, реалізована система взаємодії з ігровим світом, можливість із легкістю змінювати або редагувати вже створені текстові конструкції без сильного впровадження в програмне забезпечення.

З додаткових аспектів, які необхідно продумати та реалізовувати, можна виділити систему, яка буде відштовхуючись не від шаблону сюжету, а від потреби самого гравця.

Підбивши підсумок, було розроблено основний напрямок для розробки, обрано інструменти для досягнення поставлених цілей, а також продумано основну архітектуру майбутньої гри.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ ГРИ З ПСИХОЛОГІЧНОЮ ТЕХНІКОЮ «ПОРОЖНЬОГО СТІЛЬЦЯ»

2.1. Визначення варіантів використання та структури системи.

Гра з використанням психологічного методу "порожнього стільця" створюється з метою допомогти людині навчитися бачити шляхи для вирішення конфліктних ситуацій.

Вимоги користувача. Користувачеві необхідні такі функції.

У головному меню.

1. Продовження гри на минулій точці.

2. Можливість змінювати налаштування графіки та перепризначення функціональних клавіш.

У самій грі.

1. Можливість взаємодіяти з механіками.

1.1 Пересування.

1.2 Переміщення предметів.

1.3 Можливість піднімати предмети.

1.3.1 Можливість зберегти предмет в інвентарі.

1.3.2 Можливість викинути предмет з інвентарю.

1.3.3 Можливість вивчити предмет докладніше.

2. Можливість починати та вибирати діалоги.

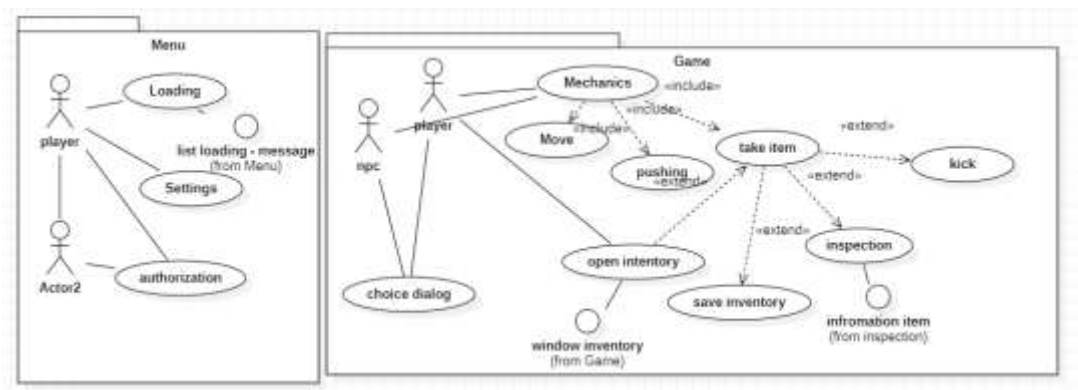


Рисунок.2.1. Варіант використання

Функціональні вимоги:

1. Геймплей: у гравця буде можливість вільно досліджувати локацію, взаємодіяти з предметами, а також брати участь у діалогах з ігровими подіями. Під час взаємодії з ігровими подіями подальші локації змінюють своє відображення.

2. гра створюється з урахуванням того, що діалогова система може змінюватися, тому функціонал повинен передбачати можливість швидкого редагування наявного в грі тексту.

Нефункціональні вимоги.

1. Відчуття.

1.1 Поняття в управлінні гри та взаємодією з ігровим світом повинно проходити для звичайних користувачів не більше 10 хвилин, для досвідчених користувачів від 5 хвилин.

1.2 Час відповіді системи для звичайних запитів не повинен перевищувати 0.016 секунди, а для більш складних запитів - 0.03 секунди.

1.3 Інтерфейс гри має бути простим і зрозумілим, а також усі основні інструменти управління мають бути добре візуалізованими.

1.4 Стабільність. Гра не повинна вилітати або ж видавати помилки під час використання, протягом усього часу, коли гра запущена.

2. Продуктивність. На гру не повинні впливати фактори ззовні.

3. Можливі варіанти використання.

3.1 Після зміни текстових файлів гра повинна автоматично підлаштовувати змінений текст і працювати без збоїв.

3.2 Оновлення версій. Оновлення гри повинні змінюватися за допомогою системи латка, а також зачіпати невелику кількість файлів.

Аналізуючи функціональні вимоги гри, ми можемо скласти структуру класів і підходів, які мають забезпечити систему необхідним функціоналом. На рис.2.2. наведено діаграму класів, які використовуються для керування ігровою сценою (світом).x

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

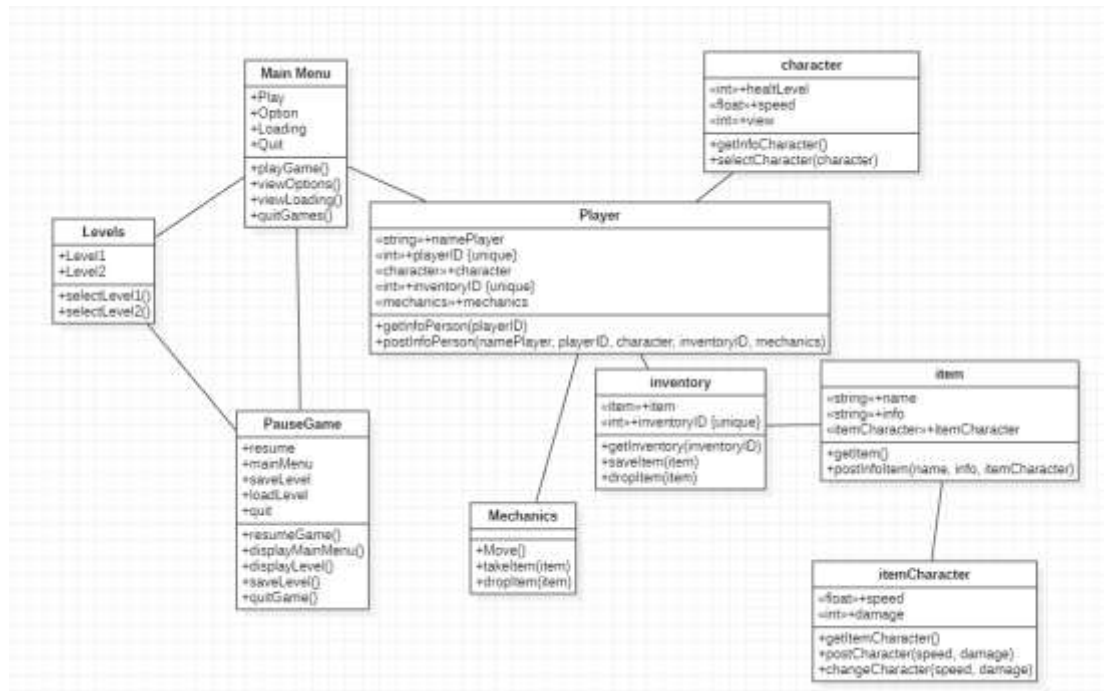


Рисунок. 2.2. Діаграма класів сцени

У даній діаграмі можна виокремити класи, які спроектовано на шаблоні "Фабричного методу" та "Одинака".

Фабричний метод допомагає створювати екземпляри певних класів. Тобто фабричний метод може бути оголошений в одному класі, а його підкласи належать до різних функціоналів. Самі ж підкласи можуть успадковувати майже весь код з базового класу. Таким чином написані такі елементи:

1. загальний клас Player з визначеними йому методами.
2. Дочірні до нього класи інвентарю і механік.
3. загальний клас для створення самого персонажа і властивостей.

Одинак - це породжувальний патерн проєктування, який гарантує, що в класу є тільки один екземпляр і надаємо до нього глобальну точку доступу. У цій діаграмі в такий спосіб були реалізовані функції Mechanics, Main Menu.

Також деякі функції з роботою діалогової системи також використовують патерн "Одинаки".

Хоча патерн має свої недоліки, до яких можна віднести:

1. проблеми з тестуванням.

2. проблеми з роботою з безліччю потоків.

Якщо його використовувати лише для обмеженого функціоналу його використання навпаки допоможе спростити роботу.

Систему було спроектовано на основі кількох патернів, таких як Singleton і State. На рисунку 2.3. представлено діаграму основних класів, які використовуються для управління основними тригерами, діалоговою системою і взаємодією з ігровим світом.

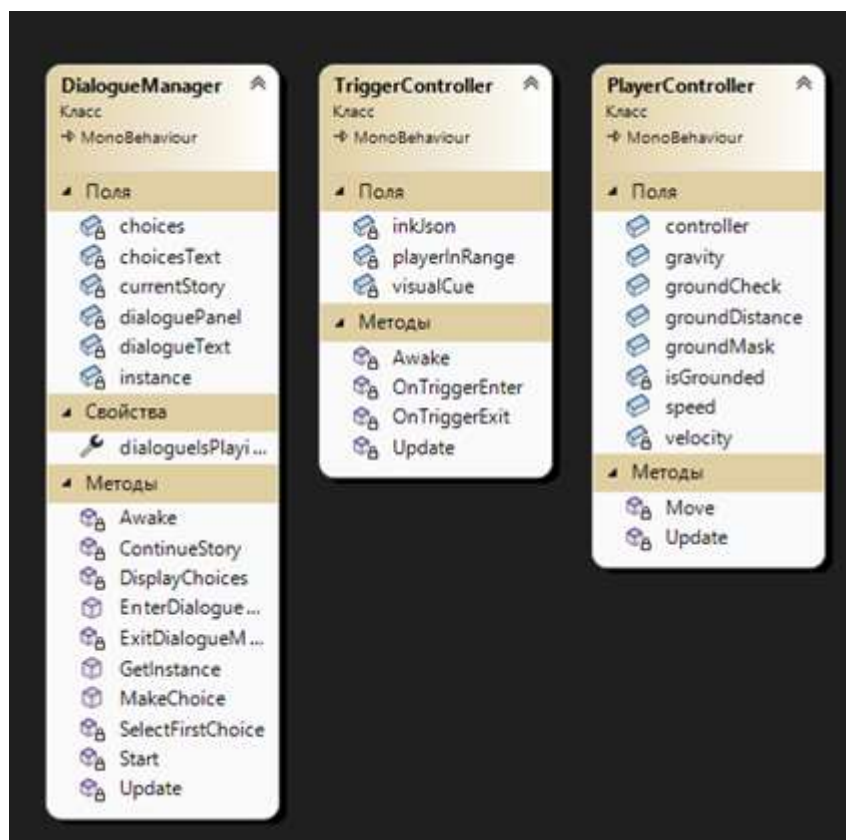


Рисунок. 2.3. Діаграма основних класів.

На цьому рисунку відображено три основні класи. DialogueManager - класовий скрипт, який описує поведінку діалогової системи. У ньому реалізовані наступні компоненти:

choices - це масив об'єктів типу GameObject. Який зберігає в собі кількість можливих виборів у діалозі.

choicesText - це масив об'єктів типу TextMeshProUGUI. Який використовується для відображення тексту вибору.

currentStory - це об'єкт типу Story, який завантажує необхідну історію.

dialoguePanel - це об'єкт типу GameObject, який використовується для відображення отриманої історії на сцені гри.

dialogueText - це об'єкт типу TextMeshProUGUI, який відповідає за візуальне відображення отриманого тексту в грі.

instance - це екземпляр об'єкта. У цьому випадку слугує для реалізації патерну Singleton, який дає змогу створити лише один екземпляр зазначеного типу DialogueManager для всієї програми.

dialogueIsPlaying - це псевдонім для типа структури Boolean. Який слугує для перевірки, чи увімкнено один із діалогів, який блокує можливість повторно запустити діалог, якщо його вже було розпочато.

Awake - метод, який служить для ініціалізації стану гри перед тим, як гру буде завантажено. Він служить для того, щоб ініціювати скрипт DialogueManager.

ContinueStory - метод, який відстежує, чи є можливість продовжити історію. Якщо ж історію неможливо продовжити, викликається метод завершення історії ExitDialogueMode.

DisplayChoices - метод, який реалізує наявність в історії вибору варіантів відповідей у діалозі.

EnterDialogueMode - метод, який служить для того, щоб почати діалогову систему і відобразити початок діалогу.

GetInstance - конструктор класу DialogueManager.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

MakeChoice - метод, який слугує для вибору варіанта відповіді та відображення результату відповіді на зроблений вибір.

SelectFirstChoice - метод, який відповідає за перш промальовування вибору інтерфейсу.

Start - це метод, який відповідає за виклик скрипт безпосередньо перед викликом будь-якого з методів. У цьому випадку служить для ініціалізації основних компонентів і відключення візуальних складових інтерфейсу.

Update - це метод, який викликається кожен кадр. У цьому випадку слугує регулярною перевіркою роботи тригера на запуск діалогової системи.

TriggerController - класовий скрипт, який відповідає за роботу кат-сцен, а також початку діалогової системи.

InkJson - це об'єкт типу TextAsset, який відповідає за завантаження файлів із діалогами в саму гру.

playerInRange - це псевдонім для типа структури Boolean. Який відповідає за перевірку відстані між тригером і гравцем. Для того, щоб початок діалогу починався в певних точках, а не на всій сцені.

visualCue - це об'єкт типу GameObject, який відповідає за відображення візуальної підказки для запуску діалогової системи.

Awake - метод, який служить для ініціалізації скрипт. У цьому випадку вимикає відображення візуальних даних інтерфейсу при завантаженні скрипт.

OnTriggerEnter - метод, який слугує тригером для виконання скрипту в той момент, коли гравець потрапляє в зону дії.

OnTriggerExit - метод, який слугує тригером для виконання скрипту в той момент, коли гравець залишає зону дії.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

Update - це метод, який викликається кожен кадр. У цьому випадку в цьому методі відбувається перевірка, чи гравець перебуває в зоні тригера і чи починає діалог.

PlayerController - це клас, який контролює роботу і взаємодії персонажа з навколишнім світом.

controller - це об'єкт типу CharacterController. Який відповідає за пересування персонажа по сцені.

gravity - це змінна типу float для розрахунку гравітації для персонажа.

groundCheck - це об'єкт типу Transform, який взаємодіє з методом Physics для перевірки, перебуває гравець на землі чи ні.

groundDistance - це змінна типу float. Яка відповідає за те, на якій відстані від землі починає працювати гравітація для персонажа.

groundMask - це об'єкт типу LayerMask. Служить для визначення того, що для гри є землею.

isGrounded - це змінна типу Boolean. Відповідає за отримання результату з методу Physics для перевірки землі.

speed - це змінна типу float. Відповідає за швидкість пересування персонажа по сцені.

velocity - це об'єкт типу Vector3, що відповідає за координати персонажа на сцені.

Move - метод, у якому йде прорахунок пересування персонажа на сцені та зміни координат на самій сцені.

Update - метод, у якому відбувається запуск методу Move і перевірка, чи розпочато діалог у грі.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

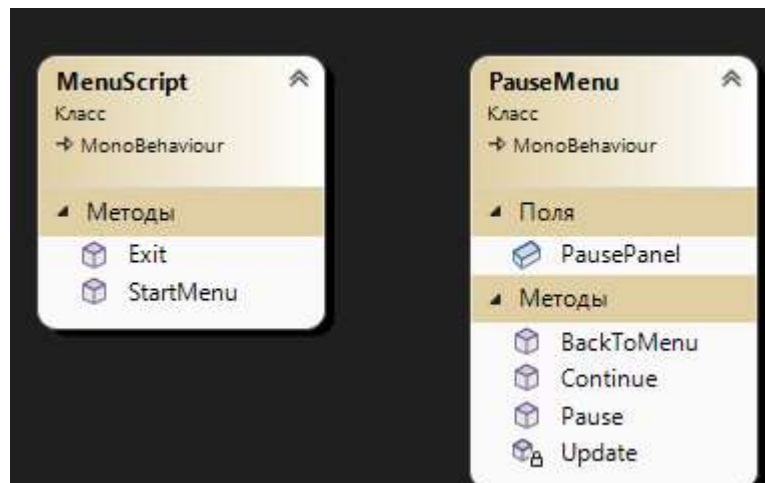


Рисунок.2.4. Основні класи для керування головним меню

Головне меню має в собі кілька скриптів, таких як MenuScript - який здебільшого використовується для переходу між рівнями, а також PauseMenu - який працює безпосередньо в самій грі.

У ці модулі включаються такі елементи як:

- керування елементами інтерфейсу.
- збереження та завантаження гри.
- зміни налаштувань гри.

Таким чином два скрипти ділитимуть різний функціонал у різних станах. MenuScript - взаємодіє з UI елементами в головному меню, коли необхідно лише перемикання між кнопками.

Своєю чергою PauseMenu - взаємодіє з елементами UI інтерфейсу з перевітками на те, що інші UI інтерфейси не заважають відображенню основних елементів головного меню. Також у ньому реалізовано метод Pause, який зупиняє основний потік гри для того, щоб у гравця була можливість взаємодіяти з UI інтерфейсом без відволікань.

Окремо варто віднести методи роботи з діалоговою системою. Діалогову систему було реалізовано за допомогою мови наративних сценаріїв для ігор "Ink Unity Integratio". За допомогою неї досить просто інтегрувати тексти, діалоги або інші текстові елементи в гру.

Ink - це мова сценаріїв, побудована на ідеї розмітки чистого тексту за допомогою потоку для створення інтерактивних сценаріїв. По суті, її можна використовувати для написання історії в стилі "Обери свій стиль" або розгалуженого діалогового дерева. Але його справжня сила в написанні діалогів з безліччю варіантів і великою кількістю перекомбінацій потоку. Процес написання відбувається за допомогою офіційного редактора Inky для написання ink-скриптів.



Рисунок. 2.5. Інтерфейс редактора Inky.

На рисунку відображено файл з основним текстом. Інформація з файлу відображена на рисунку 2.5.

Сині поля відповідають за початок "історії", а також місце кінця "End".

Завдяки спеціальному редактору ми можемо редагувати історію і бачити результат в правій колонці редактора.

Синтаксис чорнила досить простий. ">" говорить про те, що ми починаємо розповідати історію і підключення додаткових історій для гравця будуть непомітні.

Для створення вибору використовуються спеціальні символи такі як "+", "*". Також синтаксис дає змогу створювати множинний вибір, важливо лише підтримувати вкладеність діалогів.

Наприклад, для того, щоб після вибору був ще один вибір, необхідно ввести кілька "* * *". Як показано на рисунку 2.5. Структура chosen("назва змінної") - створює локальну змінну, яка записує і передається спеціальному методу "chosen", який можна викликати на будь-якому етапі "історії".

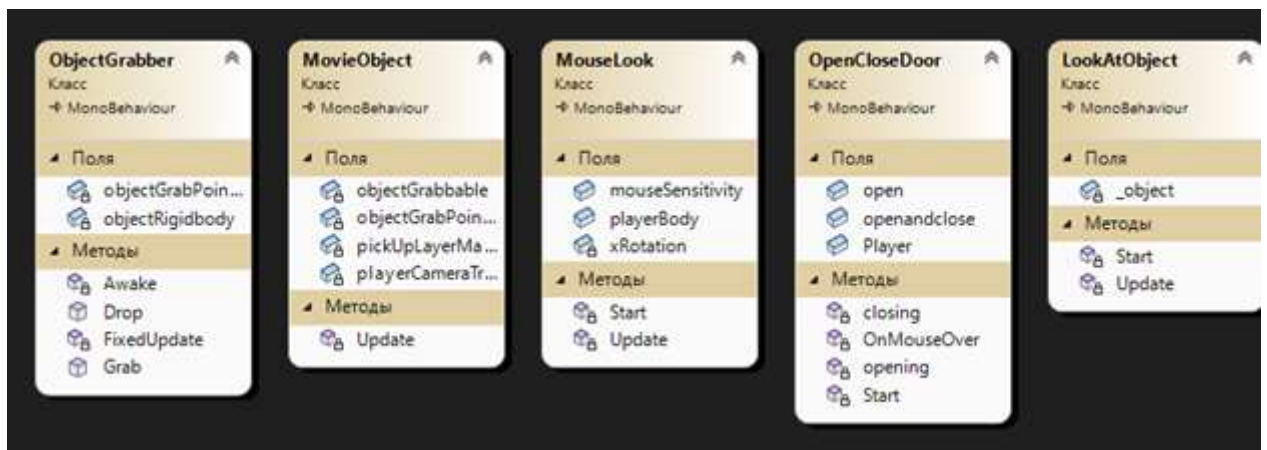


Рисунок. 2.7. Допоміжні скрипти для PlayerController.

Для класу PlayerController були розроблені додаткові класи до яких належать:

ObjectGrabber - клас, який відповідає за можливість підіймати та взаємодіяти з предметами на рівні.

MovieObject - додатковий клас, який слугує помічником для класу "ObjectGrabber", бо відстежує і зберігає в собі об'єкти, пов'язані з відстеженням натискань на клавіші та перевірок на маску об'єктів.

MouseLook - додатковий клас у який винесено основний функціонал роботи з камерою головного персонажа.

OpenCloseDoor - клас, у якому налаштована система взаємодії з елементами, що відчиняють двері в грі.

LookAtObject - скрипт, який допомагає відстежувати об'єктам розташування гравця.

2.2. Розробка функціональних алгоритмів роботи програми.

Для структуризації проєкту було використано один із підходів, який описано в офіційній документації Unity.

Метою, якого є створення структури з каталогів. Хочеться зазначити, що єдиного способу організації проєкт не існує. Але ось деякі принципи, які використовуються:

1. імена каталогів, а також файлів не повинні мати в собі проблеми.
2. Вкладеність у каталогів. У кореновому каталозі зберігаються лише ті файли, які мають у собі схожу структуру.
3. Розмежування каталогів. Створені ресурси знаходяться в окремому місці від тих, що були додані ззовні.

Структуру файлової системи в Unity можна налаштовувати відповідно до завдань проєкту і використання принципів структуризації проєкту дає змогу спростити етап розроблення та масштабування проєкту.

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

відправлення. На цій діаграмі також відображено функції, які спрацьовують при виконанні певних умов. Детальний опис параметрів наведено у таблиці 2.1.

Таблиця 2.1

інформація про елементи у діаграмі послідовності

Назва атрибуту	Призначення
Main Menu	Скрипт роботи з головним меню.
Levels	Скрипт роботи з рівнями та доступності до них.
PauseGame	Скрипт, що відповідає за повернення до розділу Main Menu, а також за роботу в ігровому сценарії безпосередньо.
Player	Назва основного класу гравця. У ньому зберігаються необхідні методи для правильної роботи.
Character	У ньому зберігаються основні характеристики головного гравця.
Inventory	Основний скрипт для обробки інвентарю гравця.
Item	Основний клас для обробки подій, пов'язаних із предметами на сцені.

ItemCharacter	У ньому зберігаються основні властивості отриманих предметів.
Mechanics	Клас, який відповідає за основні можливості в грі.

Також під час розроблення алгоритмів було створено діаграму компонентів.

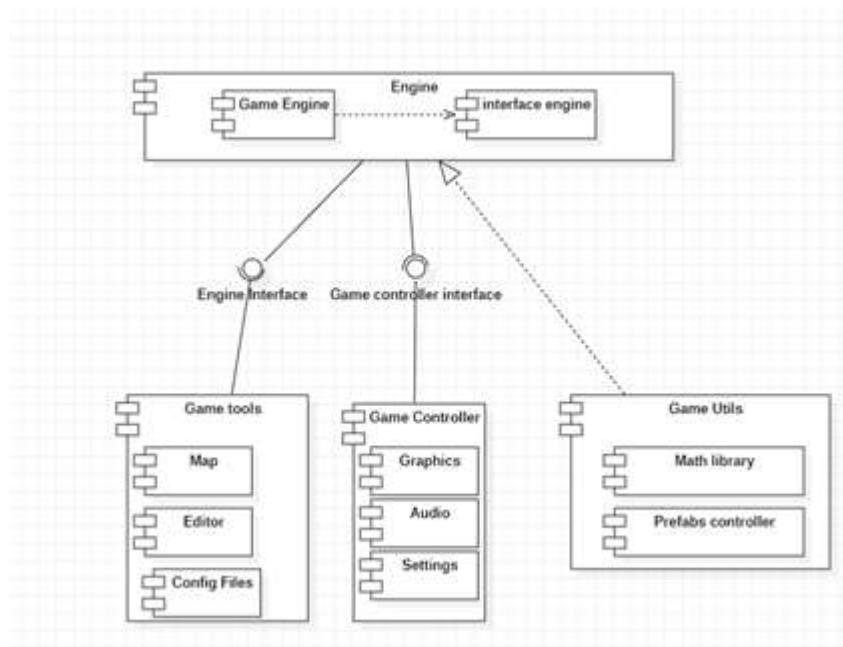


Рисунок. 2.10. Діаграма компонентів

Діаграма компонентів - діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонентів відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного

забезпечення може бути представлено як компоненту. Деякі компоненти існують під час компіляції, деякі - під час компонування, а деякі під час роботи програми.

На діаграмі відображені такі елементи:

- Компоненти – фізично існуюча частина системи, яка забезпечує реалізацію класів і відносин, а також функціональної поведінки модельованої програмної системи. На даній діаграмі це:

- Map – інструмент роботи з картою;
- Editor – компонент, який забезпечує зміни;
- Config file – компонент, який зберігає основні налаштування;
- Graphics – компоненті, контролю налаштуванням графіки;
- Audio – компоненті, контролю налаштуванням аудіо;
- Settings – зміни менш значних параметрів програми;
- Math library – компонент, що відповідає за логіку гри;
- Prefabs controller – компоненти, який керує використанням префабів;
- Game engine – компонент, який відповідає за основні можливості додатку;
- Interface engine – компонент керування інтерфейсом ядра.

2.3. Проектування та реалізація алгоритмів роботи системи

Основними модулем і елементами є діалогова система. Цей модуль відповідає за відображення текстових повідомлень у кат-сценах, а також у спілкуванні з головним персонажем. До нього входить:

1. відображення тексту в кат-сцені.
2. зміна UI інтерфейсу залежно від розміру історії.
3. Керування історією.
4. Керування вибором у діалоговій системі.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

5. Відстеження наявності історії в цій локації.

6. Керування ігровою сценою (завмирання гри) під час програвання діалогу.

7. Керування швидкістю друку текстового повідомлення.

При створенні системи найбільше зусиль прикладалось, щоб створити просту і функціональну діалогову систему, яку можна змінювати за потребою. Перед початком діалогової системи необхідно підтримувати кілька умов, основний процес показано на Рисунку 2.11.

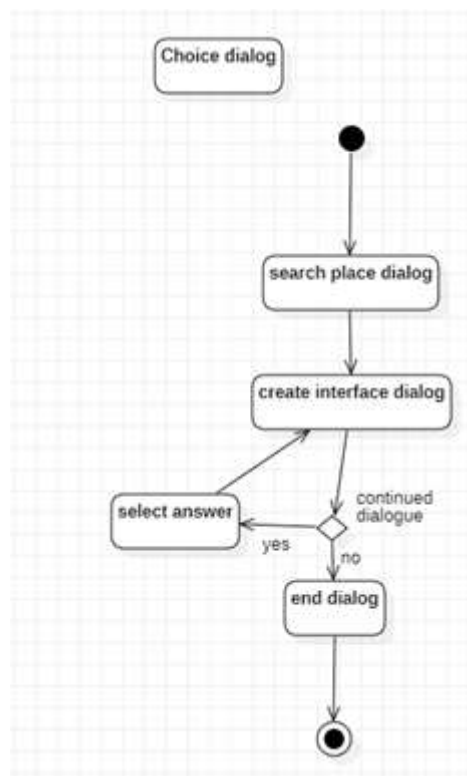


Рисунок. 2.11. Основна схема роботи с діалогової системою.

Насамперед гравцеві необхідно знайти спеціальний маркер, який відображається в правому верхньому куті. Він повідомляє про те, що в цій локації дана можливість прослухати діалог.

Після запуску діалогової системи створюється ігровий інтерфейс із завантаженою історією і варіантами відповідей, якщо ж вони присутні. У

гравця є вибір продовжити історію або ж відповісти на запитання, а також гравець може закінчити діалог.

У разі ж якщо гравець вирішує відповісти на поставлені запитання або ж продовжити діалог. Система змінює вже побудований інтерфейс з умовою, які були прописані в історію.

Таблиця 2.2

Опис функцій роботи з діалоговою системою

Назва атрибуту	Призначення
Search place dialog	функція, яка щоразу перевіряє наявність тригера для відображення значка присутності діалогу на об'єкті.
Create interface dialog	функція, яка відповідає за промальовування інтерфейсу, а також за завантаження історії з INK файлу. У ньому ж відбувається і створення варіантів відповідей.
Select answer	тригер, який обробляє завантаження історії залежно від того, яку відповідь було обрано.
End dialog	вивантажує діалогову систему з файлів завантаження після завершення діалогу.

Був також додатково написаний скрипт, який відповідає за обробку взаємодії з ігровими предметами на сцені, що показано на малюнку 2.12.

Процеси взаємодії з навколишнім світом постійно спрацьовують, поки гравець перебуває на сцені, лише на моменти роботи в кат-сценах така взаємодія вимикається. Наприклад, гравець, пересуваючись сценою за

допомогою курсору, може піднімати предмети. Після натискання на клавішу мишки відбувається підняття предмета на певну висоту перед гравцем, предмет вважатиметься піднятим, поки гравець повторно не натисне на клавішу мишки.

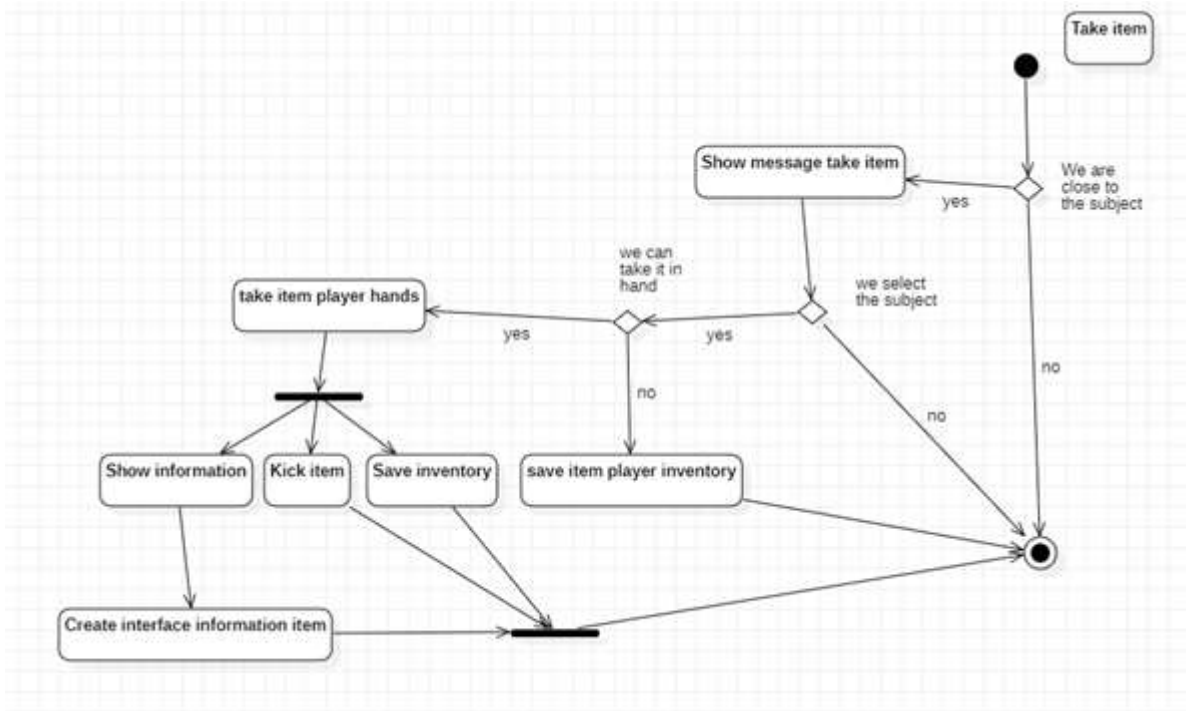


Рисунок. 2.12. Основна схема роботи з предметами

Таблиця 2.3

Опис функцій роботи з предметами на сцені

Назва атрибуту	Призначення
Show message take item	функція, яка відображає можливість підняття предмета з умовою того, що гравець перебуває поруч із самим предметом. У цій же функції відбувається і перевірка чи навів гравець на предмет. Коли ж гравець наведе на предмет курсор і натисне клавішу мишки, запуститься перевірка, чи є цей предмет допустимим для підняття.

Take item player hands	функція, що перевіряє наявність предмета в руках. Якщо ж у гравця були порожні руки, підбирає предмет і записує його в перший активний слот, що своєю чергою дає можливість спрацювання наступним функціям.
Kick item	функція, яка відповідає за кидок предмета.
Save inventory	функція, що відповідає за збереження предмета з активного слота в інвентарі.
Create interface informati on item	функція, яка завантажує отриману історію або ж опис предмета на сцені.

Таким чином було розглянуто основні алгоритми та функції роботи з системами.

2.4. Реалізація принципу побудови діалогів

Система діалогів повинна мати загальну структуру побудови для того, щоб створити систему, яка буде якомога точніше передавати підходи, що використовуються в психологічних методиках. Для цього використовується "Базова модель" когнітивно поведінкової терапії та зокрема робота з "критиком". Пояснення до Рисунку 2.13 наведено в таблиці 2.4.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

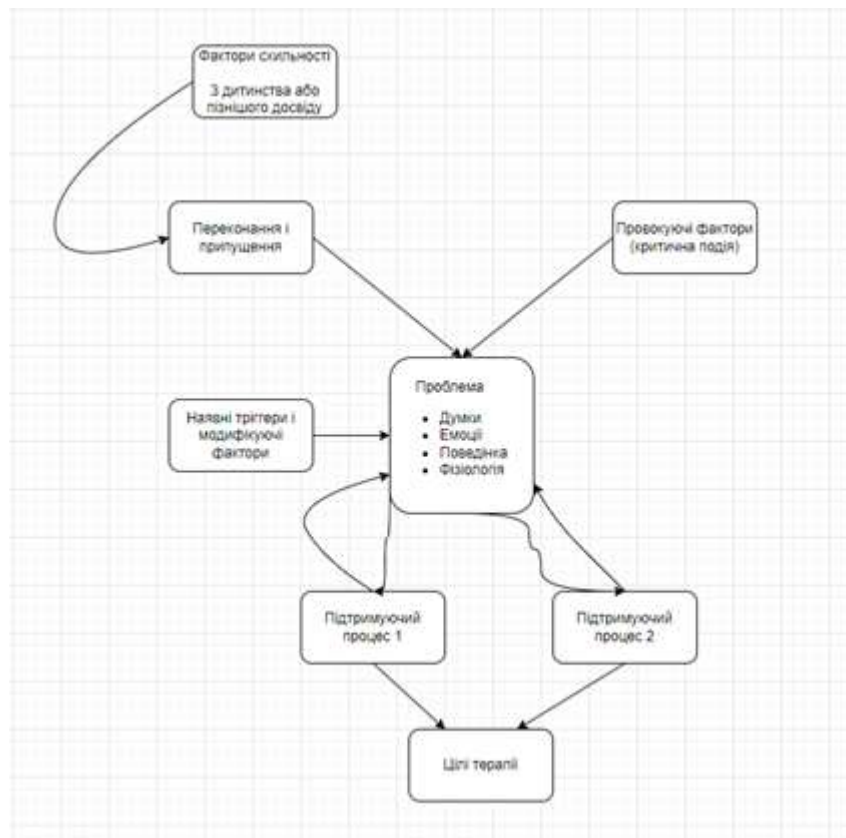


Рисунок. 2.13. Загальна модель тривожних розладів та індивідуального формулювання випадку у КПТ

Таблиця 2.4.

пояснення моделі тривожних розладів

Назва атрибуту	Призначення
Фактори схильності. З дитинства або пізнішого досвіду	це історії з минулого, які впливають на сьогоднішній досвід.
Переконання і припущення	Це уявлення про те, що людина думає про себе, оточуючих і про життя загалом

Провокуючі фактори (критична подія)	Це події або ж ситуації, які провокують виникнення проблеми зараз.
Наявні триггери і модифікуючі фактори	Це події, які можуть збільшувати та послаблювати вплив ситуації на проблему.
Підтримуючий процес	Це психологічні процеси через що існує проблема.
Цілі терапії	Розв'язання проблеми
Проблема	Це ситуації, які турбують людину найчастіше.

На основі схем і рисунків, що були відображені, було побудовано схему, яка наведена в додатку Б. Ця схема використовується для побудови діалогів у самій грі.

Метою створення схеми слугує збереження спрямованості за психологічним методом, підтримання людини в стабільній формі, а також систематичному підході до розв'язання ситуацій, що виникають, які передбачені в ігрових ситуаціях.

Такий підхід дає змогу покращити ігровий досвід, а також дасть змогу визначити можливі проблеми та варіанти вирішення безпосередньо гравцем.

Висновки до розділу 2

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

Етап проектування гри з використання психологічної техніки "порожнього стільця" базується на реалізації діалогової системи, що безпосередньо пов'язані як із самою технікою, так і ігровими механіками. Взаємодія відбувається за допомогою створених скриптів.

Для кращого розуміння основних подій у грі були створені додаткові діаграми послідовностей і розроблений основний підхід для структуризації підходів гри. Також було розроблено основні підходи роботи з компонентами.

Були розглянуті основні підходи в реалізації алгоритмів і наведені приклади роботи поточних систем і основних методів під час використання. В останньому пункті відображено підходи, які використовуються в психологічній техніці, пояснення структури підходів у терапії.

На основі даних підходів було розроблено алгоритм, який використовується для побудови діалогової системи та створення варіантів відповідей. Детально описані події, які збільшують ефект впливу на гравця.

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

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ СИМУЛЯЦІЇ

Додаток було розроблено на рушії Unity з використанням графічних кат-сцен, прорахованих на двигуні. Після компіляції всіх основних скриптів та стисканні необхідних асетів та скриптів. Отримані файли можна використовувати практичних на будь-яких пристроях.

Тестування проходило під операційну систему Windows, додаток має комфортно працювати на системах, які підходять під мінімальні вимоги:

- процесор із тактовою частотою 2.4 ГГц або вище;
- 2 ОЗП;
- 1 ГБ вільного місця на жорсткому диску;
- Графічний пристрій на 1 ГБ з підтримкою DirectX9.

Система з мінімальним необхідними параметрами достатньо для того, щоб гра працювала на частоті кадрів (60 fps). Для всіх систем, які вищі за мінімальні вимоги, цей показник може збільшуватися. Тобто для всіх систем, які не старше 2-6 років, що зі свого боку підходять для більшості сучасних комп'ютерів.

3.2. Структура інтерфейсу. Інтерфейс та порядок роботи з грою.

Після запуску гри користувач опиняється в головному меню, в якому він може змінити налаштування гри, завантажити гру, якщо вона була збережена, почати нову і вийти з гри.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок. 3.1. Головне меню.

При натисканні на "Старт" - гравця переноситься на найперший рівень, кнопка "Завантажити" - активна лише в тому випадку, якщо гравець потрапляв на чек-поінти. При натисканні на неї гравець починає з того, чек-поінта в якому перебував. Кнопка "Налаштування" - відкриває підменю із загальними налаштуваннями гри. Кнопка "Вихід" - виконує закриття гри.

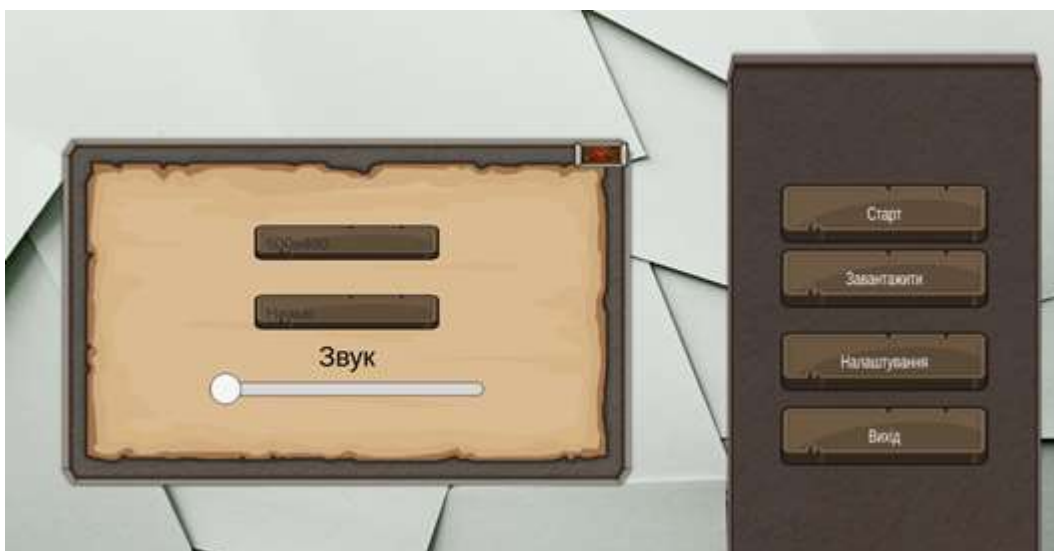


Рисунок. 3.2. Меню налаштувань

Після того як гравець натисне на кнопку "настройки" йому стануть доступні 3 види налаштувань.

1. Налаштування роздільної здатності екрана. Наразі передбачено налаштування під роздільну здатність 1920x1080, 1024x800, 600x400

2. Налаштування керування графікою. Реалізовано 3 види налаштувань. Високі, середні та низькі.

3. налаштування звуку. Це налаштування дає змогу повністю прибрати звук із гри.

Після того як гравець натисне на кнопку "Старт". Гра автоматично завантажить перший рівень (Рисунок. 3.3.).

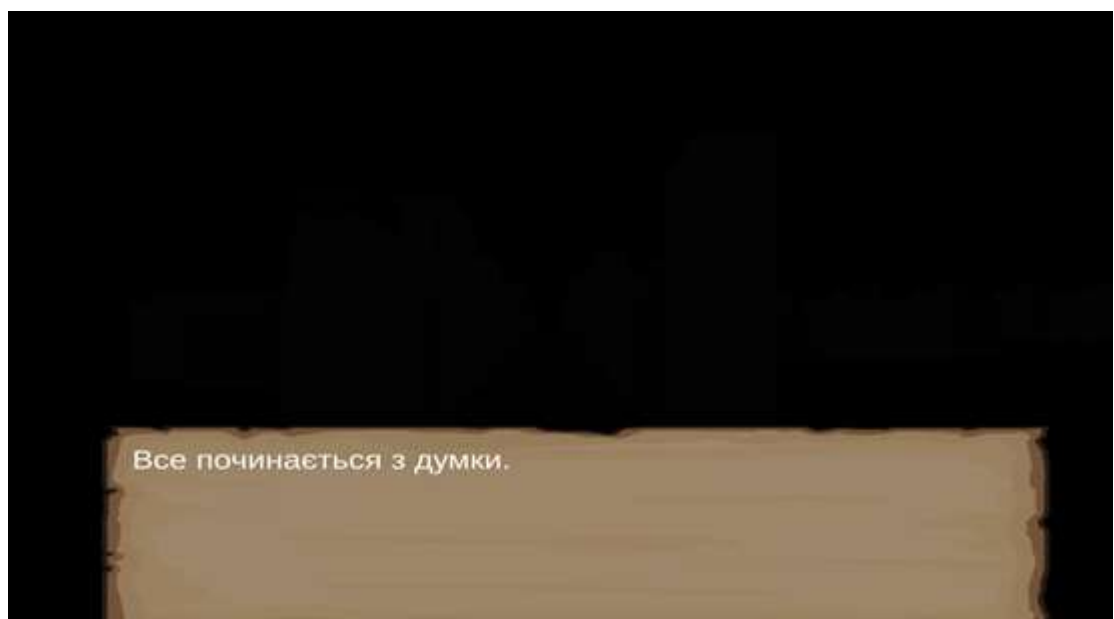


Рисунок. 3.3. Програвання кат-сцени на першому рівні

Якщо ж гравець уперше запускає гру, то після початкової кат-сцени, перш ніж почати гру, буде ввімкнено невеликий відеоролик із навчанням з основним управлінням. Метою відеоролика є познайомити гравця з усім управлінням

Рисунок 3.4.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок. 3.4. Інформаційний модуль



Рисунок. 3.5. Підказка про те, що доступний діалог

Якщо ж гравець у процесі підійде до місця, в якому доступний діалог, у правому верхньому кутку буде відображена підказка Рисунок.3.5. Для того, щоб її запустити гравцеві необхідно натиснути "Е".

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

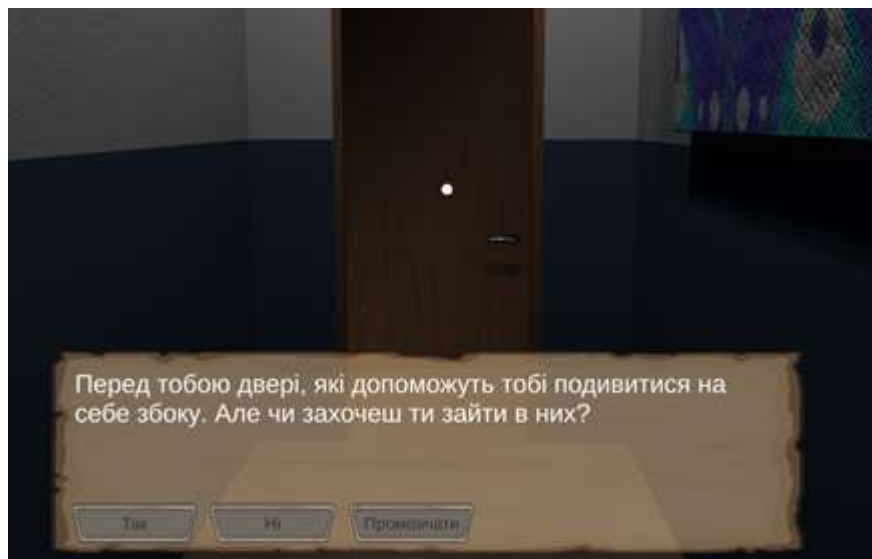


Рисунок. 3.6. Відкритий діалог

Якщо ж гравець відкрив діалог. Ігрове поле блокується, гравець досі може використовувати мишку і клавіатуру, але саме пересування гравця та ігрові події буде зупинено. Наразі у гравця є можливість контролювати, як довго буде відображатися текст. Для того, щоб продовжити гравцеві необхідно натиснути на клавішу "Space". Якщо ж гравець хоче вибрати один із варіантів відповідей у діалозі. Йому необхідно за допомогою клавіш пересування вибрати необхідний діалог і натиснути на "Space".

Також у грі присутня можливість поставити гру на паузу Рисунок. 3.7. Для того, щоб це зробити гравцеві необхідно натиснути на клавішу "Esc". Після цього відкриється меню, в якому гравець може зберегти свій прогрес. Завантажитися з останньої точки, відновити гру і повернутися в головне меню.

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

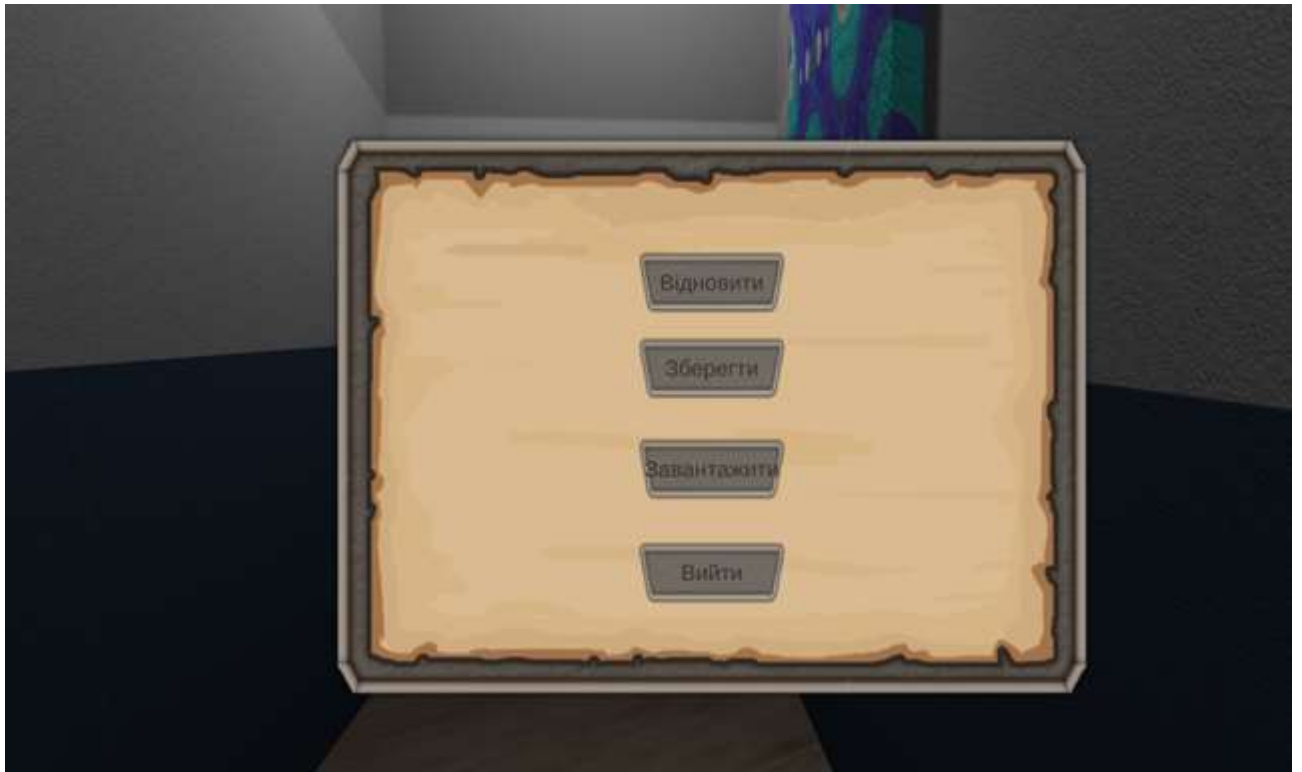


Рисунок. 3.7. Меню паузи.

3.3. Тестування роботи системи

Тестування ігрового додатка відбувалося за допомогою сценарію "Test Case". Перевага використання полягає в тому, що в ньому є набір вхідних значень, передумов виконання, очікуваних результатів і постумов виконання.

Далі представлені тест-кейси взаємодії з предметами з діалоговою системою і кат-сценами.

Взаємодія з дверима, щоб відкрити та закрити їх:

Вимоги: Персонаж гравця перебуває біля зачинених дверей.

Кроки:

1. Персонаж підходить до дверей.
2. Персонаж наводить курсор на двері.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Персонаж натискає ліву кнопку миші.
4. Персонаж відпускає ліву кнопку миші.

Очікувані результати: Дверь має відчинитися.

Фактичний результат: Дверь відчинилась

Взаємодія з об'єктами

Вимоги: Персонаж перебуває поруч із невеликим об'єктом.

Кроки:

1. Персонаж наводить курсор на об'єкт.
2. Персонаж натискає ліву кнопку миші.
3. Персонаж відпускає ліву кнопку миші.
4. Персонаж переміщує курсор в інше місце.
5. Персонаж натискає ліву кнопку миші.
6. Персонаж відпускає ліву кнопку миші.

Очікувані результати: Персонаж може підняти невеликі предмети.

Фактичний результат: Персонаж підняв невеликий предмет.

Взаємодія декількох об'єктів

Вимоги: Персонаж повинен підняти невеликий предмет і спробувати взяти ще один.

Кроки:

1. Персонаж наводить курсор на інший об'єкт.
2. Персонаж натискає ліву кнопку миші.
3. Персонаж відпускає ліву кнопку миші.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Очікувані результати: Гравець повинен упустити піднятий раніше предмет.

Фактичний результат: Гравець упустив піднятий предмет.

Висновки до розділу 3

У цьому розділі були складені мінімальні системні вимоги для комфортного використання програми. Також були відображені основні, можливості використання гри.

Гра має в собі реалізовану систему, яка дозволяє взаємодіяти з ігровим світом, а також з елементами інтерфейсу такими як: головне меню, меню паузи.

Також у процесі тестування було протестовано основні механіки роботи додатка, використовуючи скомпільований додаток за допомогою тест-кейсів. Невеликі помилки в роботі логіки застосунку було знайдено та виправлено, наразі ж застосунок працює стабільно та без збоїв.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час виконання роботи було спроектовано та реалізовано основні системи для управління з діалоговою системою з урахуванням психологічного методу "порожнього стільця". Для досягнення результату було виконано такі кроки:

1. Під час проведення аналізу вимог до застосунку дозволив визначити основні аспекти роботи з методом "порожнього стільця", визначити необхідні вимоги з погляду ігрових механік, а також їхні функціональні можливості. Вивчення аналогів використання психологічних прийомів в інших застосунках дало змогу визначити, що варіанти впливу на гравця досить широкі, але дослідження показали, що для досягнення поставленого завдання необхідно докласти основних зусиль до опрацювання діалогової системи, зручності роботи з UI інтерфейсом, також було помічено, що на гравця позитивно впливає наявність навчальних елементів, які знайомлять гравця з основними механіками гри. Поставлені вище вимоги дали змогу визначитися з основними інструментами реалізаціями, якими стали Unity, Adobe Photoshop, Blender і Ink мова наративних сценаріїв.
2. Були проаналізовані можливі варіанти використання та підходів. Для цього були створені діаграми використання та на їхніх основі були розроблені діаграми класів для аналізу структури додатка.
3. Для правильного відображення психологічного методу, було вивчено основні підходи в терапії. На основі використовуваних схем, було складено діаграму сценарію для діалогової системи.
4. Були опрацьовані основні аспекти взаємодії основних класів і класів, що допомагають.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

5. Одним з основних аспектів роботи був опис використання "мови наративних сценаріїв" з відображенням синтаксису мови, її переваг і варіантів використання на готовому додатку.
6. Реалізований додаток має в собі готові набори для використання діалогової системи з можливістю швидкої зміни під необхідні потреби. Розділи користувацького інтерфейсу такі як: головне меню, меню паузи та можливості налаштування.
7. Також було продемонстровано основні механіки гри, відображено кілька кат-сцен і проведено тестування додатка, яке допомогло виявити помилки, які вдалося виправити.

Створений додаток є готовим для використання. Додаток може допомогти в навчанні психологічного прийому, навчанню нових варіантів поведінки.

У майбутньому застосунок можна розширювати, доповнюючи новими механіками або ж використовуючи не тільки один психологічний метод. Внаслідок обраної архітектури та комплексу введення файлової системи подальший розвиток та масштабування відбуватиметься легше. Додаток не замінює консультацію у психологів, але може допомогти відчутти на собі один із підходів, які використовують.

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

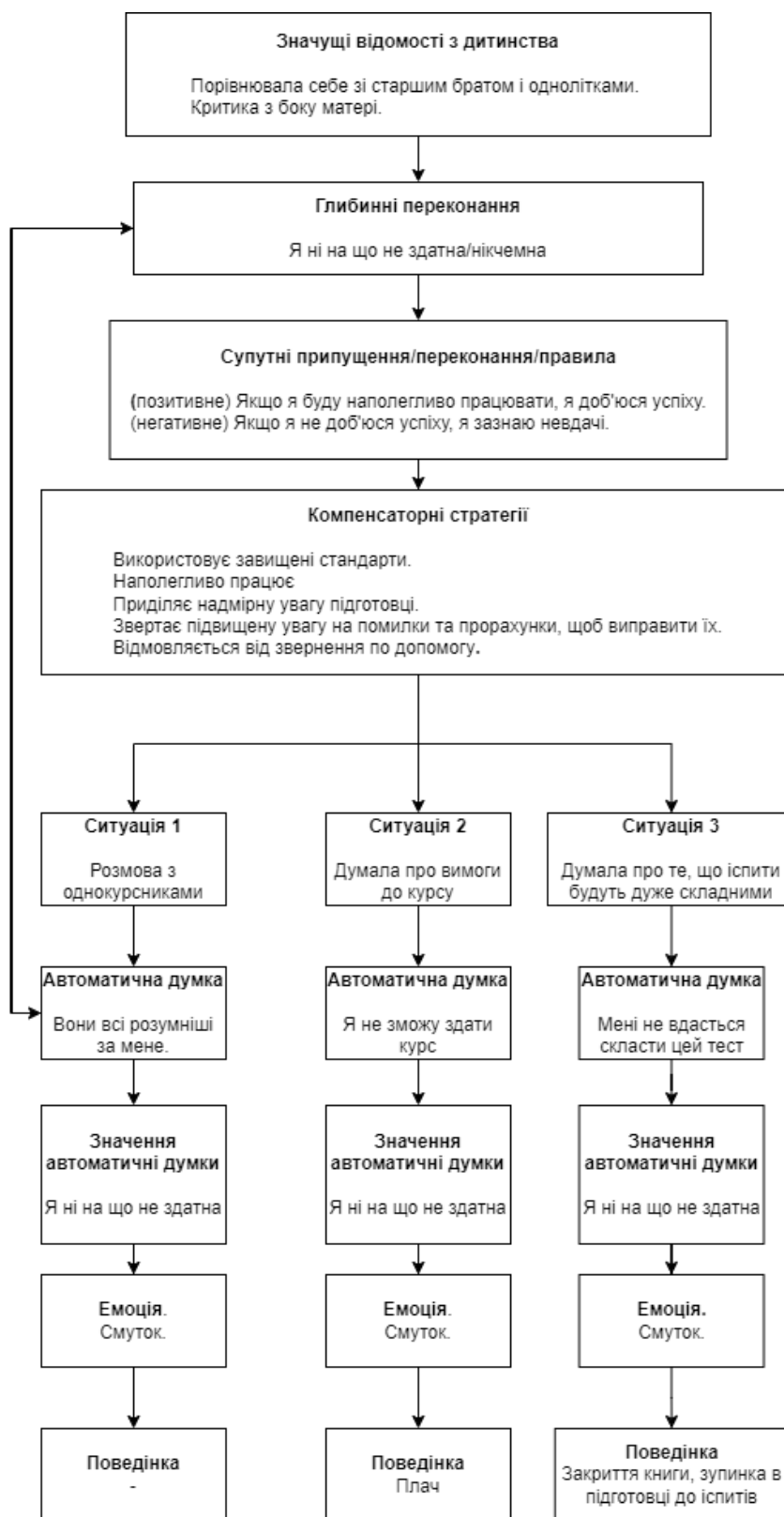
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вестбрук Д., Кеннерлі Г., Кірк Дж. Вступ у когнітивно-поведінкову терапію. 978-ме вид. Львів : СВІЧАДО, 2014. 420 с.
2. Гаєс С. Звільнений розум. 978-ме вид. Київ : publishing, 2022. 496 с.
3. Сергієнко О. Якщо ви надмірно переживаєте за все. Львів : СВІЧАДО, 2018. 100 с.
4. Станчишин В. Стіни в моїй голові. Київ : Віхола, 2021. 208 с.
5. Best practices for organizing your Unity project | Unity. *Unity*. URL: <https://unity.com/how-to/organizing-your-project>.
6. Class Diagram - StarUML documentation. *Introduction - StarUML documentation*. URL: <https://docs.staruml.io/working-with-uml-diagrams/class-diagram>.
7. Cutscenes and trailers with timeline and cinemachine - unity learn. *Unity Learn*. URL: <https://learn.unity.com/project/cutscenes-and-trailers-with-timeline-and-cinemachine>.
8. It's all in your mind: visual psychology and perception in game design. *Game Developer*. URL: <https://www.gamedeveloper.com/design/it-s-all-in-your-mind-visual-psychology-and-perception-in-game-design>.
9. Singleton. *Refactoring and Design Patterns*. URL: <https://refactoring.guru/design-patterns/singleton>.
10. State. *Refactoring and Design Patterns*. URL: <https://refactoring.guru/design-patterns/state>.
11. Unity - scripting API: animator. *Unity - Manual: Unity User Manual 2022.3 (LTS)*. URL: <https://docs.unity3d.com/ScriptReference/Animator.html>.
12. Writing web-based interactive fiction with ink. *inkle*. URL: <https://www.inklestudios.com/ink/web-tutorial/>

ДОДАТКИ

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

приклад заповненої діаграмою під час терапії



Діаграма підходу для діалогової системи

