

Ім'я користувача:
Олена Бондаренко

Дата перевірки:
29.05.2023 14:59:00 EEST

Дата звіту:
29.05.2023 14:59:55 EEST

ID перевірки:
1015300092

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005535

Назва документа: Tiahnyriadno_Illia_Anatoliiovych_PZ20130

Кількість сторінок: 68 Кількість слів: 6472 Кількість символів: 55082 Розмір файлу: 17.41 MB ID файлу: 1014971838

14% Схожість

Найбільша схожість: 10% з джерелом з Бібліотеки (ID файлу: 1011458481)

11.1% Джерела з Інтернету

401

Сторінка 70

13.2% Джерела з Бібліотеки

194

Сторінка 72

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

1

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»
Кафедра «Комп'ютерні інформаційні технології»

Пояснювальна записка
до кваліфікаційної роботи бакалавра

на тему: «Реалізація взаємодії з користувачем в комп'ютерній грі з використанням Unreal Engine blueprint»
за освітньою програмою: «Інженерія програмного забезпечення»
зі спеціальності: «121 Інженерія програмного забезпечення»
Виконав: студент групи «ПЗ20130»

	<hr/>	/Ілля ТЯГНИРЯДНО/ <hr/>
	(підпис студента)	(Ім'я ПРІЗВИЩЕ)
Керівник:	<hr/>	/доц. Олександр ІВАНОВ/ <hr/>
	(підпис)	(посада, Ім'я ПРІЗВИЩЕ)
Нормоконтролер:	<hr/>	/доц. Світлана Волкова/ <hr/>
	(підпис)	(посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з
праць інших авторів без відповідних посилань.

Студент

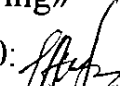
(підпис)

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»
Department «Computer information technology»

Explanatory Note
to Bachelor's Thesis

on the topic: «Implementation of user interaction in a computer game using Unreal Engine blueprints»
according to educational curriculum «Software engineering»
in the Speciality: «121 Software engineering»

Done by the student of the group PZ20130:  /Illia TIAHNYRIADNO/

Scientific Supervisor:

 /Oleksandr IVANOV/

Normative controller:

/Svitlana Volkova/

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»
Department «Computer information technology»

Explanatory Note to Bachelor's Thesis

on the topic: « Implementation of user interaction in a computer game using Unreal Engine blueprints»
according to educational curriculum «Software engineering»
in the Speciality: «121 Software engineering»

Done by the student of the group PZ20130:

/ Illia TIAHNYRIADNO/

Scientific Supervisor:

/Oleksandr IVANOV/

Normative controller:

/Svitlana Volkova/

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Факультет «Комп'ютерні технології і системи»

Кафедра: «Комп'ютерні інформаційні технології»

Рівень вищої освіти: бакалавр

Освітня програма: «Інженерія програмного забезпечення»

Спеціальність: «121 Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри КІТ

_____/Вадим ГОРЯЧКІН/
(підпис)

Дата _____

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

студенту Тягнирядну Іллі Анатолійовичу

1. Тема роботи: «Реалізація взаємодії з користувачем в комп'ютерній грі з використанням Unreal Engine blueprint»

Керівник роботи: Іванов Олександр Петрович, доцент

затверджені наказом № 1209 ст від 07.12.2022

2. Строк подання студентом роботи: _____.____.202_ р.

3. Вихідні дані до роботи: _____

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

Вступ

Збір та аналіз вимог до програмного продукту

Проектування меню наладування

Проектування ігрового інтерфейсу

Проектування ефектів «дебафів»

Розробка меню наладувань

Розробка ігрового інтерфейсу та ефектів «дебафів»

Загальні висновки

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

Презентація

Відео роботи програми

КАЛЕНДАРНИЙ ПЛАН

Стадія	Зміст	Строки виконання
Технічне завдання	Постановка задачі, збір інформації, виріб та обґрунтування критеріїв розробки. Попередній вибір методів рішення задач. Визначення вимог до технічних засобів. Узгодження і затвердження технічного завдання.	31.01.23 -18.02.23
Робочий проект	Програмування та відлагодження програми.	19.02.23 - 20.05.23
	Тестування програми	20.05.23 - 27.05.23
	Розробка, узгодження і затвердження програмної документації.	27.05.23 - 12.06.23
	Подання кваліфікаційної роботи до кафедри	07.06.23
	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	28.06.23

Студент

(підпис)

Ілля ТЯГНИРЯДНО

(Ім'я ПРІЗВИЩЕ)

Керівник роботи

(підпис)

доц. Олександр ІВАНОВ

(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка складається з 8 розділів:

- вступ – в даному розділі описується коротка інформація про ігрову індустрію, сутність розробки, її актуальність. Складається з 1 сторінки;
- збір та аналіз вимог до програмного продукту – у цьому розділі розглядається та описується аналогічні за жанром ігри, їх аналіз, оцінювання, складається постановка подальшої роботи. Складається з 10 сторінок;
- проектування мобільного додатку – у цьому розділі описуються задачі проєкт, його функціональні вимоги, вхідні та вихідні дані, описується вибір мови програмування та компонентно-орієнтований підхід в UE4 Blueprint, проектується меню налаштувань, ігровий інтерфейс, ефекти «дебафів». Складається з 10 сторінок;
- розробка взаємодії з користувачем – у цьому розділі, на основі прототипу, розробляється меню налаштувань, ігровий інтерфейс, ефекти «дебафів». Складається з 15 сторінок;
- загальні висновки – підсумки всієї роботи. Складається з 1 сторінки;
- список використаних джерел – містити бібліографічний список використаної літератури. Складає 1 сторінки;

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ЗБІР ТА АНАЛІЗ ВИМОГ	9
1.1. Опис аналогів ігрових двигунів	9
1.2. Опис взаємодії з користувачем в аналогічних іграх	11
1.3. Постановка задачі.....	18
Висновки до розділу:	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ ДЛЯ ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ	20
2.1. Задачі проекту	20
2.2. Функціональні вимоги	20
2.3. Вхідні та вихідні дані.....	21
2.4. Вибір мови програмування.....	23
2.5. Опис компонентно-орієнтованого підходу в UE4 Blueprint	23
2.6. Проектування меню налаштувань	24
2.7. Проектування ігрового інтерфейсу	26
2.8. Проектування ефектів “дебафів” (debuff) на екрані користувача.....	28
Висновки до розділу	29
РОЗДІЛ 3. РОЗРОБКА ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ	31
3.1. Розробка меню налаштувань.....	31
3.2. Розробка ігрового інтерфейсу	40
3.3. Розробка ефектів "дебафів"	42
Висновки до розділу	46
Список використаних джерел.....	48
Додатки	49
1 ВВЕДЕННЯ	4
2 ПІДСТАВИ ДЛЯ РОЗРОБКИ	5
3 ПРИЗНАЧЕННЯ РОЗРОБКИ.....	6
4 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ	7
4.1 Вимоги до функціональних характеристик	7
4.2 Вимоги до надійності.....	9
4.3 Вимоги експлуатації	9
4.4 Вимоги до складу та параметрів технічних засобів.....	10
4.5 Вимоги до інформаційної та програмної сумісності	10
4.6 Вимоги до маркування і упаковки.....	10
4.7 Вимоги до транспортування та зберігання	11
5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	12
6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ.....	13

7	ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ	14
8	БІБЛІОГРАФІЧНИЙ СПИСОК	15

ВСТУП

За останні роки індустрія комп'ютерних ігор зростає в геометричній прогресії, і мільйони людей у всьому світі насолоджуються широким спектром ігор на різних платформах. Щоб створювати успішні та захоплюючі ігри, дуже важливо забезпечити плавну та інтуїтивно зрозумілу взаємодію з користувачем це так звані UI (User Interface) та UE (User Experience).

Unreal Engine – популярний ігровий движок, який використовується розробниками з усього світу для створення високоякісних ігор. Він пропонує потужну систему візуальних сценаріїв під назвою Blueprints, яка дозволяє дизайнерам створювати ігрову механіку та взаємодію з користувачем без глибоких знань програмування.

Цей дипломний проєкт призначений для демонстрації взаємодії з користувачем в комп'ютерній грі “Nuclear Frost” за допомогою Unreal Engine Blueprints, розгляне ключові концепції та особливості системи Blueprint і продемонструє практичні приклади того, як її можна використовувати для створення різних форм взаємодії користувачів у грі.

Nuclear Frost – світ у грі багато років перебував у стані ядерної зими через вплив ядерної зброї на атмосферу Землі, тому людина зруйнувала свій будинок. Головна мета гри полягає в тому, щоб занурити гравця в атмосферу замороженого світу та боротьби людей за своє існування, а також показати, наскільки може бути небезпечним застосування ядерної зброї та зневага проблем клімату.

Загалом цей дипломний проєкт прагне забезпечити всебічне розуміння ролі взаємодії з користувачем у розробці гри “Nuclear Frost” та продемонструвати можливості системи Blueprint Unreal Engine у створенні UI (User Interface) та UE (User Experience) у відеогрі.

РОЗДІЛ 1. ЗБІР ТА АНАЛІЗ ВИМОГ

1.1. Опис аналогів ігрових двигунів

Існує багато популярних ігрових рушіїв та мов програмування, які розробники ігор можуть використовувати для створення комп'ютерних ігор. Прикладами є такі рушії, як Unity, CryEngine та Godot Engine.

Однак Unreal Engine є більш унікальний, його унікальність полягає в тому, що він має потужну візуальну систему сценаріїв під назвою Blueprints, яка створювати ігрові механіки та взаємодію з користувачем без поглиблених знань програмування.

1.1.1. Аналіз системи візуальних сценаріїв Unity

Unity - це ігровий рушій, який також може надати візуальну систему написання сценаріїв, яка називається PlayMaker, вона має вигляд як на рис. 1.1 та рис. 1.2. За допомогою PlayMaker, як і у випадку з Blueprint Unreal Engine, програмісти та дизайнери можуть створювати ігрову механіку та взаємодію з користувачем без необхідності глибоких знань з програмування. Однак система Blueprint Unreal Engine пропонує більш розширені можливості візуальної розповіді та більш потужний набір функцій, що робить її більш пріоритетним ігровим двигуном

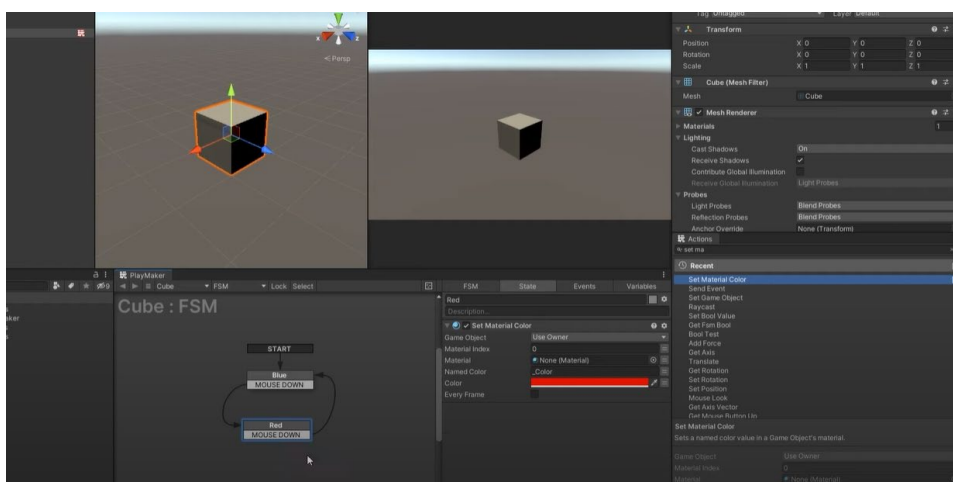


Рис. 1.1. Візуальний вигляд PlayMaker

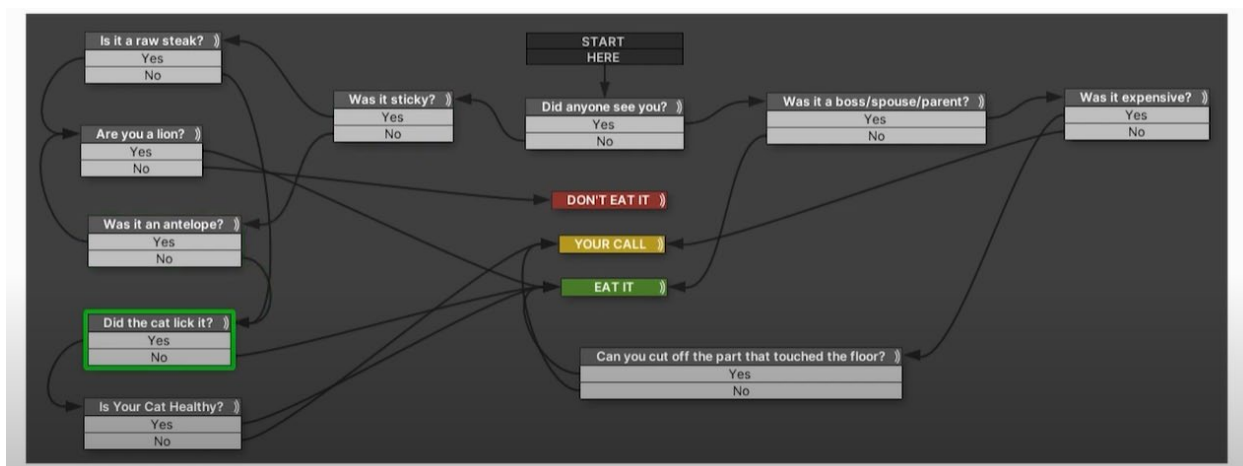


Рис. 1.2. Візуальний вигляд PlayMaker

1.1.2. Аналіз системи візуальних сценаріїв CryEngine

CryEngine - ще один з популярних ігрових рушіїв, який пропонує той самий рівень функціональності, що і Unreal Engine, з візуальною системою програмування під назвою Flow Graph, показаний на рис. 1.3 - 1.5. Однак система Blueprint рушія Unreal Engine пропонує більш інтуїтивно зрозумілий інтерфейс і ширший набір функцій, що робить її кращим вибором для розробників.

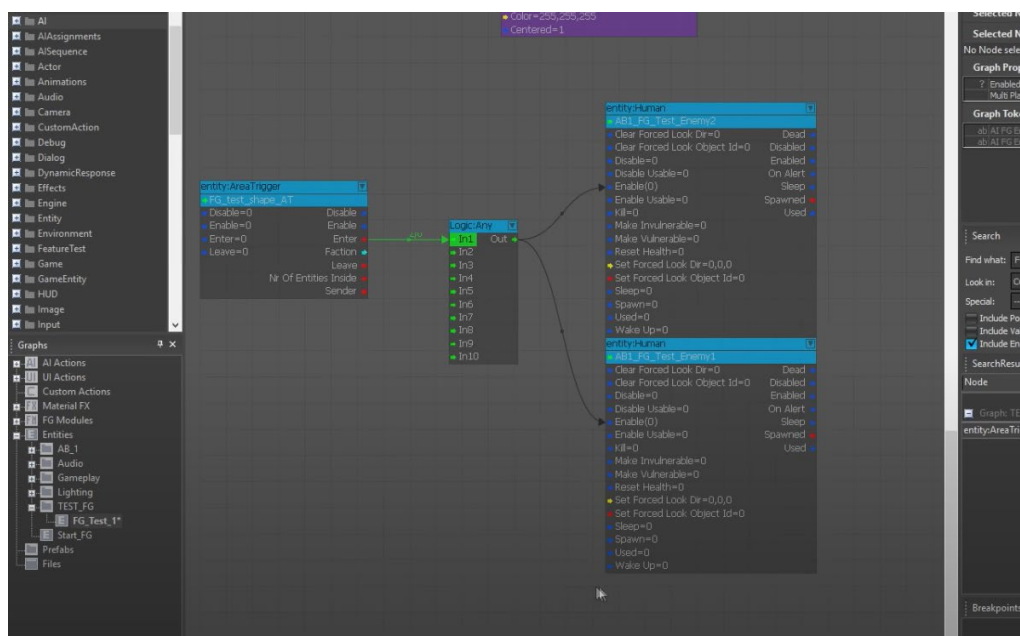


Рис. 1.3. Візуальний вигляд Flow Graph

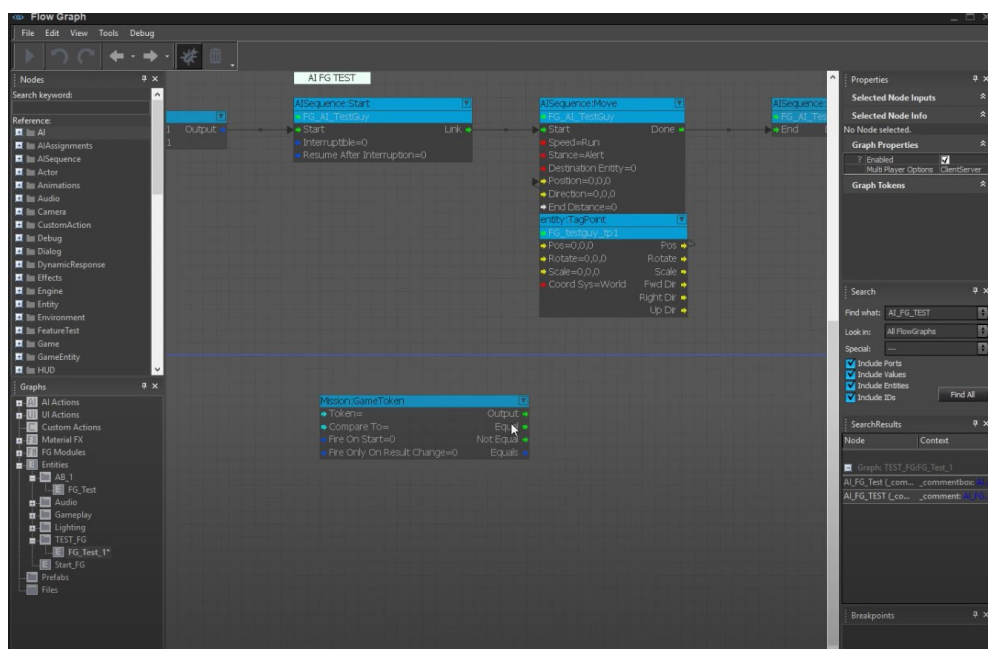


Рис. 1.4. Візуальний вигляд Flow Graph

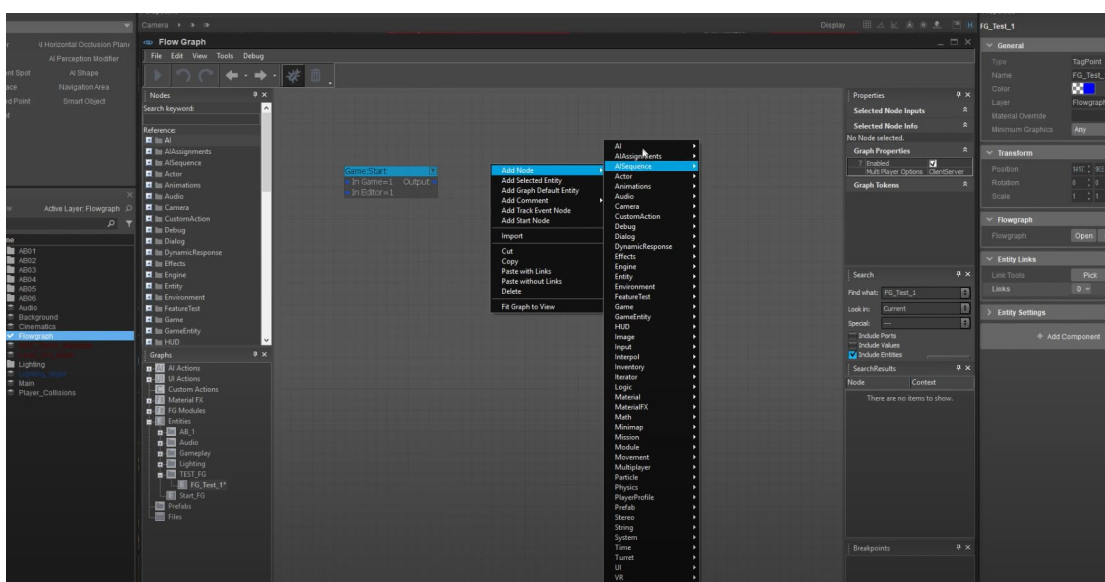


Рис. 1.5. Візуальний вигляд Flow Graph

1.2. Опис взаємодії з користувачем в аналогічних іграх

1.2.1. Аналіз реалізації взаємодії з користувачем у грі Fear the Wolves

"Fear the Wolves" - ця гра є бойовиком у жанрі Battle Royale яка була розроблена в Україні. На початку гри, гравці потрапляють на територію, яка була забруднена радіацією, і гравці повинні боротися один з одним, щоб вижити.

UI (User Interface) та UE (User Experience) у грі Fear the Wolves є важливою частиною ігрового процесу, оскільки вони відкривають доступ до інформації, необхідної для виживання у небезпечній зоні. Деякі особливості UI/UE у грі Fear the Wolves:

– Інвентар – гравці можуть відкрити свій інвентар, щоб побачити свою зброю та спорядження. В інвентарі можна побачити кількість набоїв, медичного приладдя та інших ресурсів, які гравці можуть використовувати для виживання візуальний вигляд можна побачити на рис. 1.6.



Рис. 1.6. Інвентар у грі Fear the Wolves

– Карта – гравці можуть відкрити карту гри, щоб побачити, де вони знаходяться. Карта також показує зони, які незабаром будуть заражені радіацією, що допомагає уникнути їх, візуальний вигляд можна побачити на рис. 1.7.

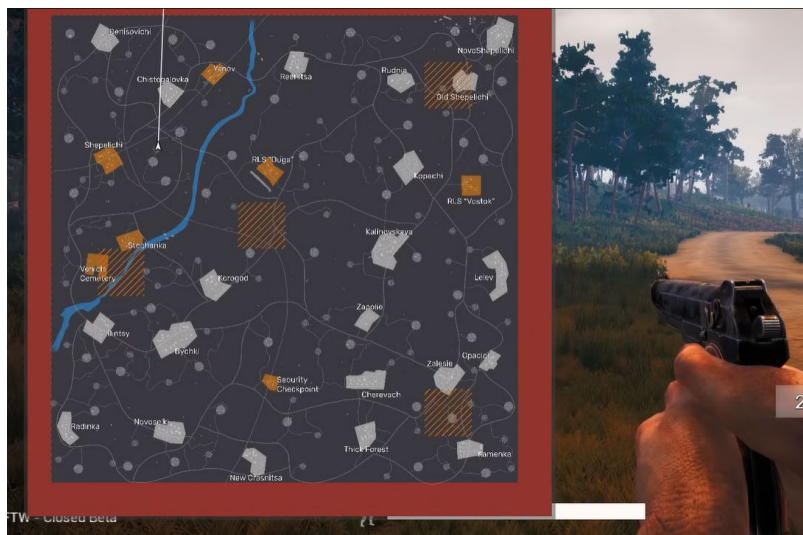


Рис. 1.7. Карта у грі Fear the Wolves

– Показники здоров'я та позитивних або негативних ефектів – гравці можуть побачити свої показники, щоб розуміти, скільки часу у них залишилося до того, як їм потрібно лікування чи їжа, візуальний вигляд можна побачити на рис. 1.8.

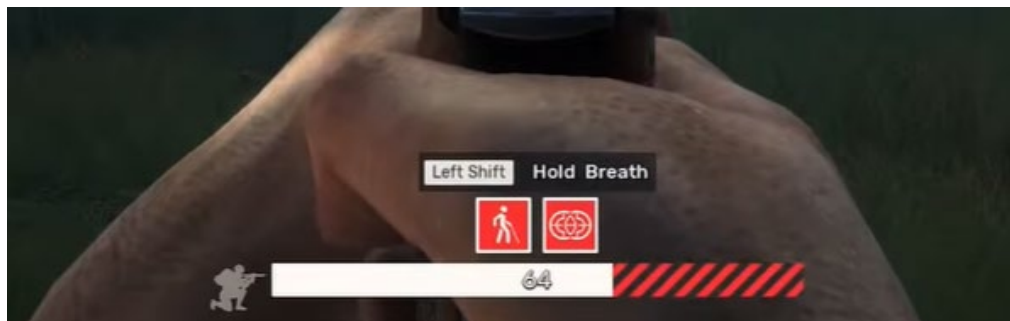


Рис. 1.8. Показники здоров'я та негативних ефектів у грі Fear the Wolves

– Анімація та звук - гра використовує анімацію та звук, щоб допомогти гравцям розуміти, що відбувається у грі. Наприклад, коли гравець отримує шкоду від монстра, екран починає пульсувати і звук стає інтенсивнішим



Рис. 1.9. Екран гравця при отриманні шкоди у грі Fear the Wolves

– Контролери та меню – гравці можуть керувати своїм персонажем за допомогою клавіатури та миші. Вони також можуть відкрити меню налаштувань, щоб налаштувати керування, налаштування звуку та графіки, а також інші параметри гри, візуальний вигляд можна побачити на рис. 1.9.

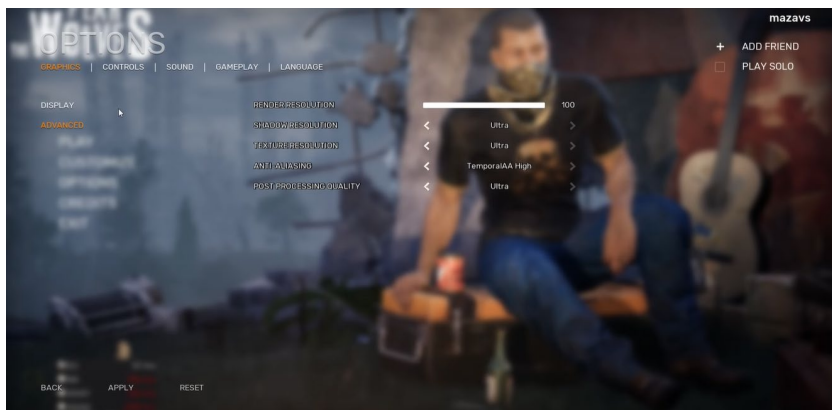


Рис. 1.9. Вікно параметрів у грі Fear the Wolves

1.2.2. Аналіз реалізації взаємодії з користувачем у грі STALKER: Shadow of Chernobyl.

"STALKER: Shadow of Chernobyl" – Ця гра розроблена в Україні та заснована на технологічній катастрофі на Чорнобильській АЕС у 1986 році. У грі гравці грають за "підричника" на ім'я Стрілець, який досліджує Зону - небезпечну територію навколо Чорнобильської АЕС після катастрофи.

UI (інтерфейс користувача) і UE (взаємодія з користувачем) у STALKER: Shadow of Chernobyl є важливою частиною ігрового процесу, оскільки вони допомагають гравцям отримати інформацію, необхідну для виживання в радіоактивній зоні Чорнобильської АЕС.

Приклади деяких функцій UI/UE у STALKER: Shadow of Chernobyl:

– Індикатор здоров'я та захисту – гравці можуть перевіряти показники свого здоров'я та захисту, візуальний вигляд можна побачити на рис. 1.10.



Рис. 1.10. Вікно індикаторів у грі STALKER: Shadow of Chernobyl.

– Інвентар – гравці можуть відкрити свій інвентар, щоб побачити наявну у них зброю та спорядження. Інвентар відображає кількість куль, гранат, аптечок та інших ресурсів, доступних гравцеві для виживання, візуальний вигляд можна побачити на рис. 1.11.



Рис. 1.11. Інвентар у грі STALKER: Shadow of Chernobyl.

– Карта – гравці можуть відкрити карту, щоб побачити, де вони знаходяться та де знаходяться інші персонажі, наприклад інші шахраї та мутанти. На карті також показано області, які можуть бути забруднені радіацією, щоб гравці могли їх уникнути, візуальний вигляд можна побачити на рис. 1.12.



Рис. 1.12. Карта у грі STALKER: Shadow of Chernobyl.

– Радіація – у грі є система радіації, яка відображає кількість радіації, яку отримує гравець у поточній зоні. Це важливо, тому що чим більше радіації в області, тим швидше почне падати здоров'я гравця, візуальний вигляд можна побачити на рис. 1.13.



Рис. 1.13. Радіація у грі STALKER: Shadow of Chernobyl.

– Контролери та меню – гравці можуть керувати своїми персонажами за допомогою клавіатури та миші. Вони також можуть відкрити меню налаштувань, щоб налаштувати елементи керування, налаштування звуку та графіки та інші налаштування гри, візуальний вигляд можна побачити на рис. 1.14.



Рис. 1.14. Меню налаштувань у грі STALKER: Shadow of Chernobyl.

– Підказки та підтримка – гра надає підказки та підтримку гравцям, щоб допомогти їм подолати ігрові труднощі. Наприклад, коли гравець стикається з новим мутантом або квестом, гра може підказати, як краще реагувати на ситуацію або яку стратегію використовувати, візуальний вигляд можна побачити на рис. 1.15.

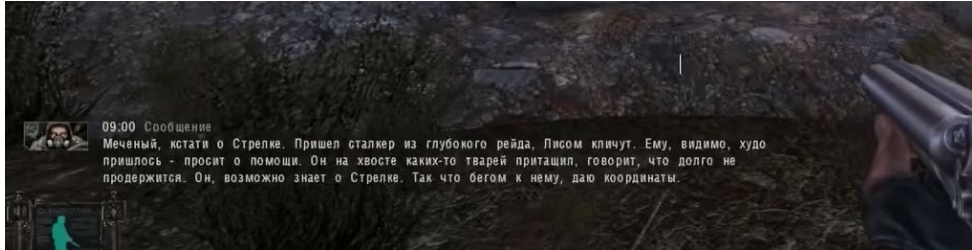


Рис. 1.15. Підказка у грі STALKER: Shadow of Chernobyl.

– Мінікарта – у грі STALKER: Shadow of Chernobyl також є мінікарта, яка допомагає гравцям легше орієнтуватися у світі гри. Мінікарт показує найближчі об'єкти і персонажів, а також маршрут гравця до мети, візуальний вигляд можна побачити на рис. 1.16.



Рис. 1.16. Мінікарта у грі STALKER: Shadow of Chernobyl.

– Анімації та звуки – важливою частиною UI/UE у STALKER: Shadow of Chernobyl є анімації та звуки. Вони допомагають гравцям краще сприймати інформацію та легше сприймати те, що відбувається в грі. Наприклад, звуки кроків та дихання допомагають гравцеві відчувати свою присутність у світі гри, а анімації пошкоджень та поранень показують, як пошкодження впливають на гравця.

В цілому, UI/UE у грі STALKER: Shadow of Chernobyl є важливою частиною ігрового процесу, які допомагають гравцям виживати у постапокаліптичному світі. Вони створюють унікальну атмосферу та допомагають гравцям орієнтуватися у світі гри, надаючи інформацію та підтримку, які необхідні для просування у грі.

1.3. Постановка задачі

Опираючись на аналіз схожих відеоігор, можна сказати, що для реалізації взаємодії з користувачем в комп'ютерній грі “Nuclear Frost” з використанням Unreal Engine blueprint потрібно розробити наступні елементи:

- Меню налаштувань.

Слід розробити меню налаштувань, яке з'являється в грі та дозволяє користувачам змінювати параметри гри, такі як гучність звуку, налаштування екрана, налаштування управління та інші. Це меню повинно бути доступним завжди і мати простий та зрозумілий інтерфейс, щоб гравець зміг швидко і легко змінювати налаштування гри на свій смак.

Для цього потрібно створити окреме вікно з меню, яке з'являється на екрані гри при натисканні клавіші. Це вікно повинно мати кілька вкладок з різними налаштуваннями. На кожній вкладці має бути блок з окремим налаштуванням, яке може змінити гравець.

Пункти меню повинні бути інтуїтивно зрозумілими та розташовані так, щоб користувач міг легко знайти і змінити потрібні йому налаштування.

- Ігровий інтерфейс (game HUD).

Ігровий інтерфейс – це набір візуальних елементів, які надають гравцям інформацію про їхній поточний стан у грі та допомагає їм приймати рішення відповідно до цієї інформації. Крім того, ігровий інтерфейс може містити інші елементи, такі як мінікарту, інвентар, і т.д.

В ігровому інтерфейсі гри будуть присутні такі елементи:

- Показник здоров'я.
- Показник енергії.

- Показник холода.
- Показник радіації.
- Інформація про зброю.
- Компас у верхній частині екрану.
- Ефект “дебафів” (debuff) на екрані користувача.

Задача полягає в візуальному повідомленні гравцю, що на нього діє ефект "дебафів" на екрані користувача. "Дебаф" - це негативний ефект, що може зменшувати рівень життя, сили атаки, рухливості чи інші характеристики.

Висновки до розділу:

Під час опису та аналізу аналогічних ігор, були підмічені головні елементи для взаємодії з користувачем, дані елементи були проаналізовані та додані до списку реалізації у грі. Опираючись на них ми можемо скласти список основних функціональних вимог та описати план для подальшого проєктування та розробки візуальної взаємодії з користувачем.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ ДЛЯ ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ

2.1. Задачі проекту

На основі проведеного опису аналогів ми виділяємо наступні задачі проекту:

- проектування та створення меню налаштувань;
- проектування та реалізація ігрового інтерфейсу;
- проектування та реалізація ефектів “дебафів” (debuff) на екрані користувача.

2.2. Функціональні вимоги

1. Функціональні вимоги для меню налаштувань у грі:
 - має бути доступним із головного меню або під час гри;
 - має містити налаштування звуку, графіки, керування тощо;
 - налаштування звуку повинні включати регулювання гучності музики, звукових ефектів та голосових коментарів;
 - налаштування графіки повинні включати регулювання якості графіки, розширення екрану;
 - налаштування керування повинні включати можливість зміни керування, у тому числі вибору клавіш, налаштування чутливості миші;
 - повинно мати можливість повернутися до значень за умовчанням, якщо гравець не задоволений поточними налаштуваннями;
 - має бути простим та інтуїтивно зрозумілим для гравців усіх рівнів досвіду.
2. Функціональні вимоги для ігрового інтерфейсу у грі:
 - має бути інтуїтивно зрозумілим та зручним для використання для гравців усіх рівнів досвіду;
 - повинен містити інформацію про поточний стан гри, таку як рівень здоров'я гравця, кількість очок досвіду тощо;

- повинен містити елементи управління, які дозволяють гравцеві взаємодіяти з ігровим світом, наприклад, кнопки для атаки або перемикання зброї;
- повинен містити меню налаштувань, що дозволяє гравцеві налаштувати гру під свої уподобання;
- повинен включати підказки та підказки, які допомагають гравцеві орієнтуватися у грі та розуміти, що йому слід робити.

3. Функціональні вимоги для ефектів “дебафів” (debuff) на екрані користувача:

- повинні бути інтуїтивно зрозумілим в будь який момент гри.

На діаграмі прецедентів яка зображена на рис. 2.1, можна побачити як користувач може взаємодіяти з меню налаштувань.

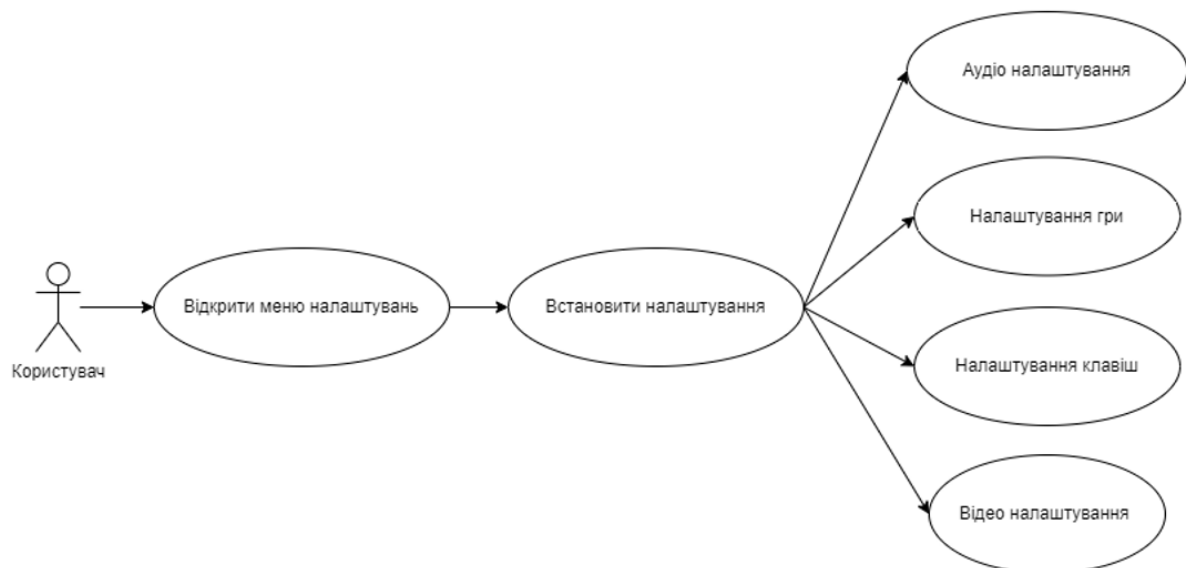


Рис. 2.1. Діаграма прецедентів

2.3. Вхідні та вихідні дані

1. Меню налаштувань

Вхідні дані:

Налаштування звуку:

Загальна гучність, включення/вимкнення музики та звукових ефектів, всі елементи відображені за допомогою Slider у якого діапазон даних від 0 до 100

Налаштування графіки: розширені параметри екрана, якість графіки, включення/вимкнення швидкісних ефектів і т.д. Данні елементи відображені за допомогою combobox або checkbox.

Налаштування керування: призначення клавіш та кнопок.

Налаштування гри: вибір мови гри, елемент відображений за допомогою combobox.

Вихідні дані:

Збережені налаштування: гучність, якість графіки, налаштування керування та налаштування гри.

Повідомлення про зміни налаштувань: якщо користувач вніс зміни в налаштування, то він може отримати повідомлення про збереження змін.

Повернення до головного меню гри: після збереження налаштувань користувач повертається до головного меню гри.

2. Ігровий інтерфейс

Вхідні дані:

Стан гравця: кількість здоров'я, енергії, рівень холоду та радіації.

Дані про місію: поточне завдання, стан поточного завдання.

Вихідні дані:

Оновлення стану: після взаємодії гравця з різними елементами гри, може змінюватися стан гравця, наприклад зменшуватися рівень життя.

Відображення повідомлень: можуть відображатися повідомлення про досягнення гравця, зміни у грі.

3. Ефекти “дебафів” (debuff) на екрані користувача:

Вхідні дані: дебаф, радіація або холод

Час залишку дебафу: це вказує на те, як довго ефект дебафу буде активним на користувачі.

Вихідні дані:

Звукове супроводження: може відтворюватися звуковий ефект, який вказує на наявність певного дебафу.

Анімаційні ефекти: можуть відображатися різні ефекти, блимання екрану, коливання зображення або зміна кольору.

2.4. Вибір мови програмування

При розробці гри в UE4, мова програмування C++ є базовою мовою програмування яка об'єднана з "Blueprint".

Blueprint - це візуальний інтерфейс програмування, який дозволяє швидко створювати складні логічні рішення без будь-яких знань C++. Однак важливо розуміти, що одного Blueprint недостатньо для створення більш гнучких рішень.

За допомогою C++ можна створювати власні класи, функції та об'єкти, що значно спрощує роботу в коді і дозволяє швидко розширювати функціональність гри.

Тобто об'єднання цих двох технологій пропонує більше можливостей для програмування і контролю над проектом, а також більш високу продуктивність.

2.5. Опис компонентно-орієнтованого підходу в UE4 Blueprint

При розробці відеоігор найчастіше використовують компонентно-орієнтований підхід (Component-based approach) він є досить популярним та практичним. Цей підхід, полягає в тому, що програмне забезпечення розбивається на окремі компоненти, які можна використовувати для створення різних функціональних блоків.

У компонентно-орієнтованому підході, кожен із розробників може швидко створити нові функціональні блоки, додаючи до них відповідні компоненти, з можливістю зміни їх параметрів, що забезпечує зручну модульність та легку змінність проекту.

В UE4 Blueprint, компонентно-орієнтований підхід дозволяє використовувати вже готові компоненти та блоки при створенні своїх ігрових об'єктів, які можуть бути модифіковані відповідно до потреб проекту.

Компоненти в UE4 Blueprint можуть бути створені та застосовані для різних об'єктів, таких як персонажі, предмети, та інші.

Наприклад, якщо розробник хоче створити персонажа з можливістю бігу, він може додати до персонажа компонент руху, який містить в собі необхідні функції та налаштування для руху персонажа, потім цей компонент можна застосувати до інших об'єктів, злегка модифікуючи цей. Розробник може додавати та змінювати компоненти в будь-який момент, що дозволяє більш гнучко налаштовувати функціональність проєкту.

Крім того, кожен компонент в UE4 Blueprint може мати свої власні події та функції, які можуть бути викликані в будь-який момент з будь-якої частини проєкту. Це дозволяє створювати та реалізовувати складну логіку при менших зусиллях.

Таким чином, компонентно-орієнтований підхід в UE4 Blueprint є потужним інструментом для створення складних ігрових проєктів. Розробники можуть швидко і легко створювати нові компоненти з функціональних блоків, розширювати і модифікувати їх, а також використовувати вже готові компоненти для підвищення швидкості та надійності своїх проєктів.

2.6. Проектування меню налаштувань

Перший основний крок при проектуванні меню налаштувань є визначення налаштувань: необхідно визначити список налаштувань, які будуть доступні в меню налаштувань, нижче буде приведений список налаштувань які будуть присутні у грі:

- аудіо налаштування;

В аудіо налаштуваннях, у користувача буде можливість, зменшити або додати гучності у звуках інтерфейсу, ефектів, діалогу, музики.

- відео налаштування;

У відео налаштуваннях, користувач зможе змінювати параметри екрана, якість ефектів, обробки екрана, дальність промальовування, якість графіки

- налаштування клавіш;

В меню клавiш, гравець зможе змiнити клавiшi якi були виставленi за замовчуванням, та пiдкорегувати чутливiсть мишi.

– налаштування гри.

В налаштуваннях гри можна вибрати мову iнтерфейсу

Другий крок в проектування є організація меню, у нашому випадку до меню налаштувань ми зможемо перейти з головного меню, меню налаштувань поділиться на два блоки, перша блок буде відображувати список налаштування, а другий блок відображає самі налаштування заданої категорії, на рис. 2.2 можна побачити розташування першої та другої частини.

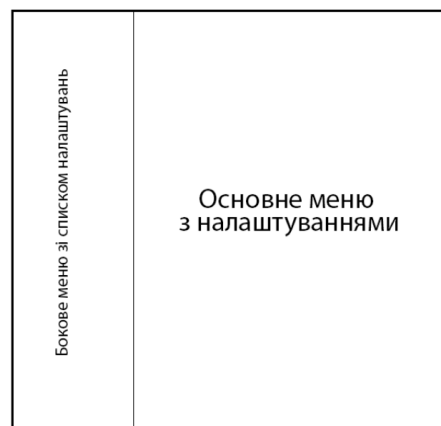


Рис. 2.2. Розташування блоків меню

На рис. 2.3. можна побачити можливість переходу до пунктів меню.

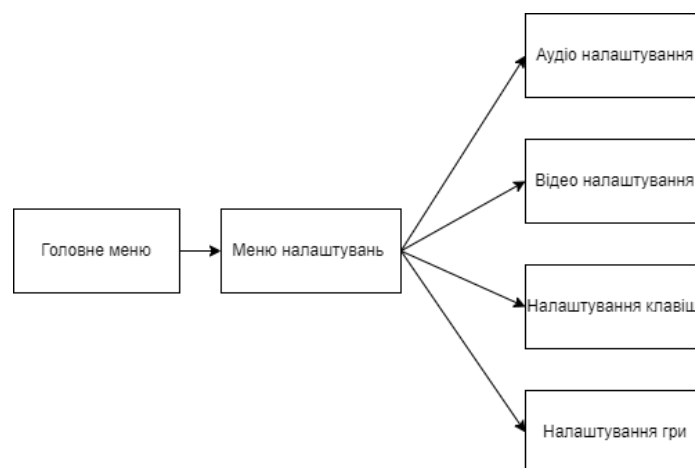


Рис. 2.3. Можливість переходу до пунктів меню

Третій крок є графічний дизайн: після визначення налаштувань та організації меню, необхідно розробити графічний дизайн меню налаштувань.

На рис. 2.4. відображено графічний дизайн бокового меню зі списком налаштувань яке було відображене на рис. 2.2.

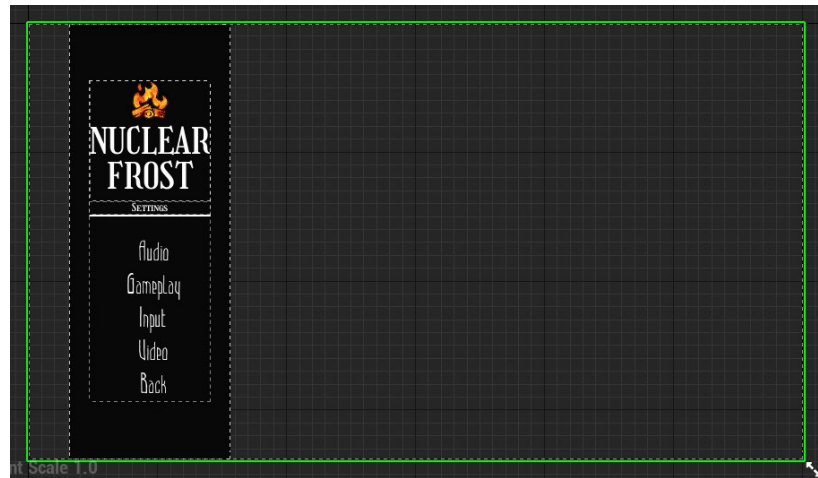


Рис. 2.3. Можливість переходу до пунктів меню

На рис. 2.4. зображено основне меню налаштувань для клавіш.



Рис. 2.4. Основне меню налаштувань

2.7. Проектування ігрового інтерфейсу

Перший крок в проектування ігрового інтерфейсу є визначення елементів які будь візуально допомогти користувачу упродовж всієї гри, такі елементи будуть:

- індикатори здоров'я, холоду, радіації, енергії;
- компас;

- назва зброї;
- кількість патронів у зброї;
- статус місії.

Другий крок є організація інтерфейсу: на цьому кроці потрібно визначити де будуть розташовані кожен із вище перелічених елементів інтерфейсу.

На рис. 2.5. зображено приблизне розташування елементів ігрового інтерфейсу

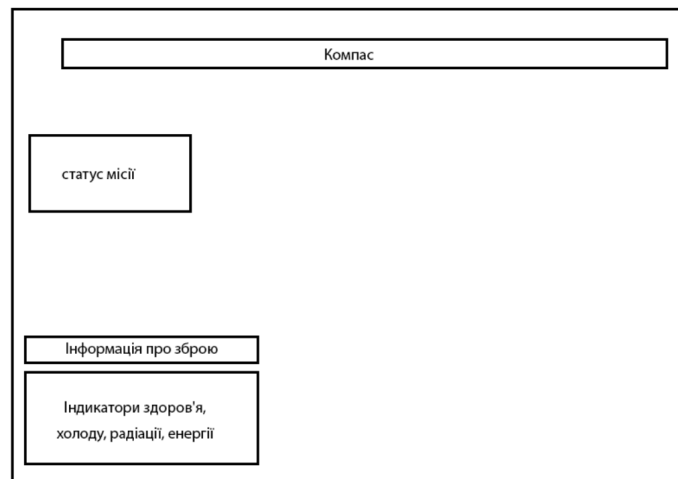


Рис. 2.5. Розташування елементів ігрового інтерфейсу

Після визначення елементів та організації меню, наступний крок полягає в розробці графічного дизайну ігрового інтерфейсу.

На рис. 2.6. зображений графічний вигляд блоку з індикаторами здоров'я, енергії, холоду та радіації.

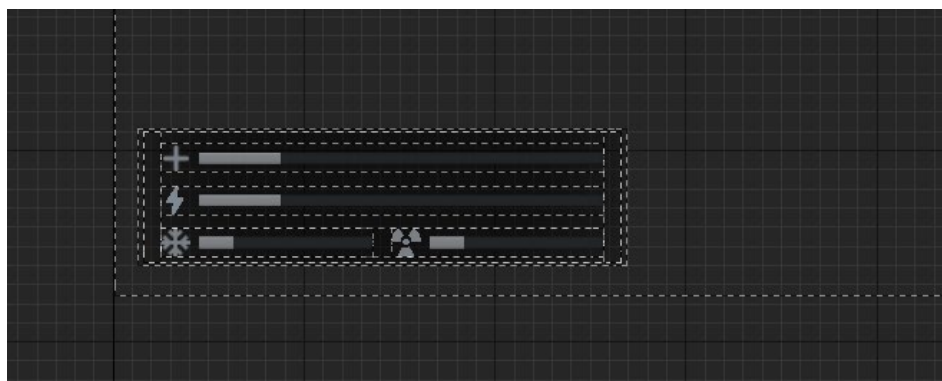


Рис. 2.6. Графічний вигляд блоків з індикаторами

На рис. 2.7. зображений графічний вигляд компаса, даний компас буде допомагати орієнтуватися гравцю за поточною місією

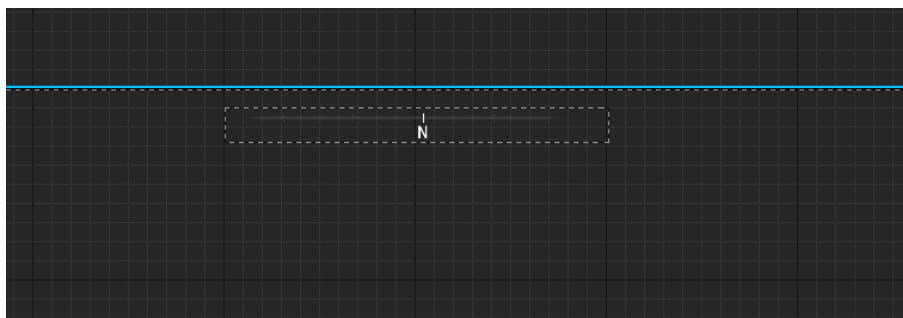


Рис. 2.7. Графічний вигляд компаса

На рис. 2.8. зображений графічний вигляд місії та її прогрес



Рис. 2.8. Графічний вигляд місії та її прогрес

2.8. Проектування ефектів “дебафів” (debuff) на екрані користувача

Одним з підходів до проектування ефектів "дебафів" є використання різних кольорів для кожного типу ефекту. Наприклад, червоний колір можна використовувати для позначення шкоди, яку завдає ворог. Це допоможе гравцям швидко і легко розпізнавати, який саме негативний ефект впливає на них.

Одним із частішим ефектом в даній грі буде ефект холоду приблизний вигляд якого представлений на рис. 2.9.

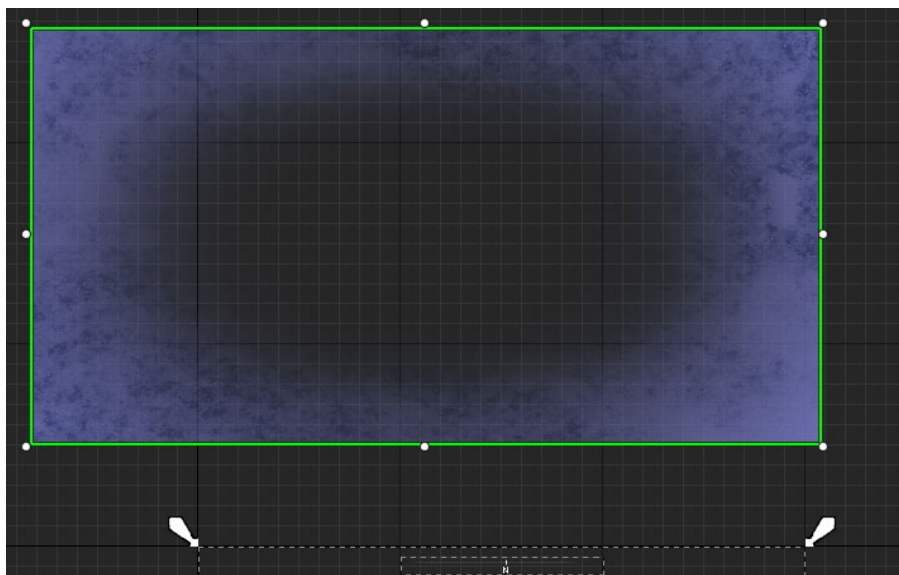


Рис. 2.8. Вигляд ефекту холоду у гравця

Також одним із важливих візуальних ефектів у гравця буде ефект радіації. Для створення ефекту радіації в Unreal Engine 4 було використано кілька методів, включаючи налаштування матеріалів, налаштування частинок, і використання постобробки, завершальний крок створення ефекту, це додати постобробку, таку як гаусове розмиття або насиченість кольорів, щоб створити більш реалістичний ефект радіації. Вигляд цього ефекту буде представлений у розділі 3.

Висновки до розділу

У розділі 2 "Проектування інтерфейсу для взаємодії з користувачем" було розглянуто багато важливих аспектів проектування інтерфейсу.

Спочатку були визначені задачі проекту та сформульовані функціональні вимоги до інтерфейсу, а також було проаналізовано вхідні та вихідні дані. Для розробки було обрано мову програмування, що відповідала вимогам проекту.

Особлива увага приділялася компонентно-орієнтованому підходу в UE4 Blueprint, що дозволило використовувати готові компоненти та прискорити процес розробки.

Після цього були спроектовані меню налаштувань та ігровий інтерфейс, які мають бути зручними та легко зрозумілими для користувачів. Крім цього,

було проектування ефектів "дебафів" на екрані користувача, що покращить візуальний досвід від гри.

В цілому, другий розділ є важливим етапом в розробці проекту та допомагає забезпечити зручну та ефективну взаємодію з користувачем. Використання компонентно-орієнтованого підходу дозволяє зменшити витрати часу та зусиль на розробку інтерфейсу, а ретельне проектування ефектів та інших елементів інтерфейсу забезпечує належний рівень комфорту для користувачів.

РОЗДІЛ 3. РОЗРОБКА ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ

3.1. Розробка меню налаштувань

До меню налаштувань, користувач зможе перейти з головного меню, яке зображене на рис 3.1.

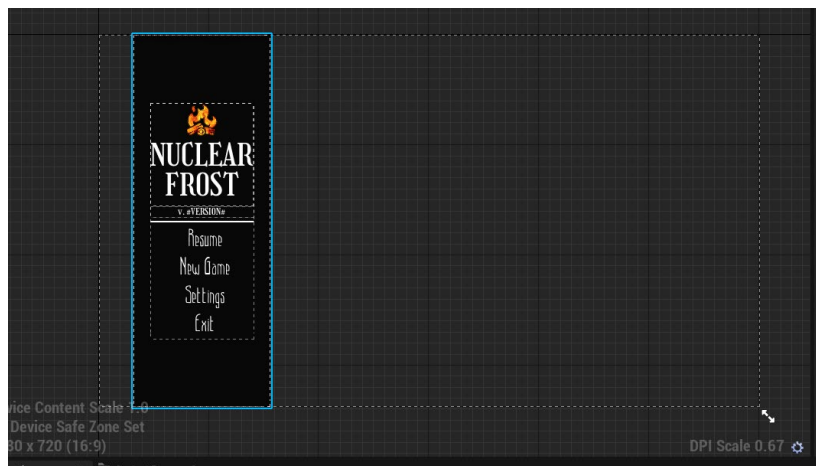


Рис. 3.1. Головне меню гри

При натисканні на кнопку “Settings”, викликається функція «ShowMainSettingsItems», це зображено на рис 3.2.

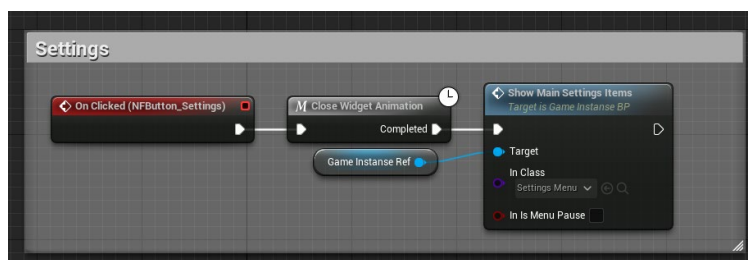


Рис. 3.2. Подія при натисканні кнопки налаштувань

Функція «ShowMainSettingsItems» зберігається у класі GameInstance, це клас, який представляє глобальний об'єкт гри в Unreal Engine, який існує протягом усього часу роботи та служить для зберігання та обробки даних, пов'язаних зі станом гри та ігровими сесіями.

Завдяки цьому класу ми зможемо виконувати функцію «ShowMainSettingsItems» будь-де, тобто ми не будемо прив'язані до конкретного вікна, реалізація функції зображена на рис 3.3.

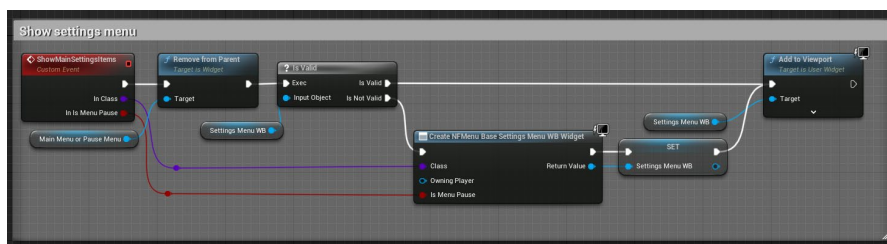


Рис. 3.3. Реалізація функції «ShowMainSettingsItems»

При виконанні функції «ShowMainSettingsItems», у користувача відкривається меню налаштувань яке представлено на рис 3.4.

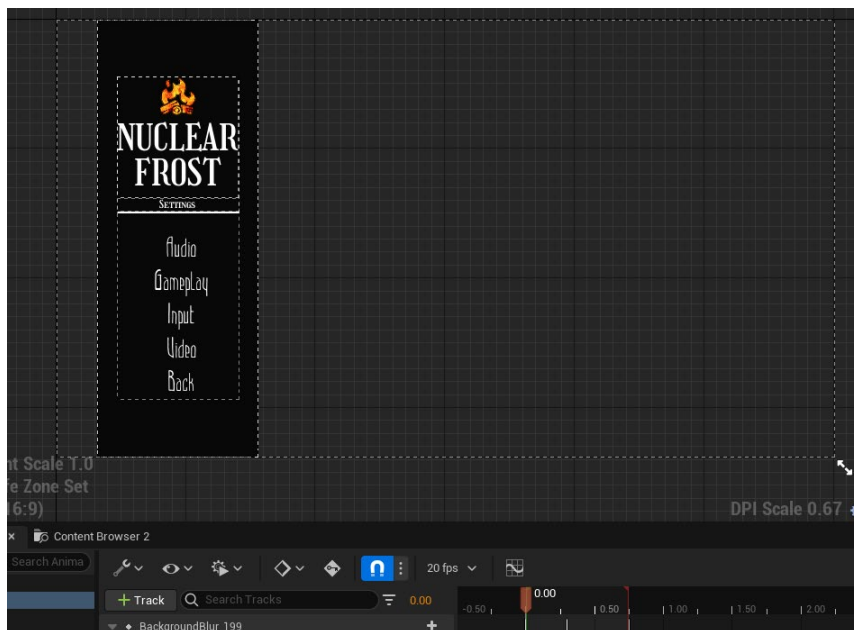


Рис. 3.4. Меню налаштувань

В меню налаштувань, у користувача буде можливість відкрити вікно з певною категорією налаштуванням (аудіо, відео, ігрові та налаштування клавіш), на рис 3.5 та 3.6 представлена подія при натисканні на категорію налаштування та реалізація відкривання вікна.

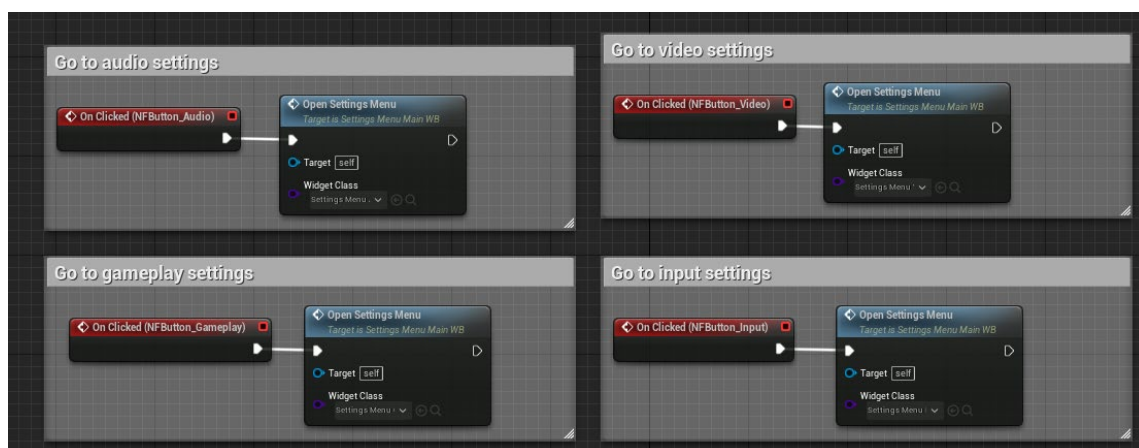


Рис. 3.5. Подія при натисканні кнопки категорії налаштування

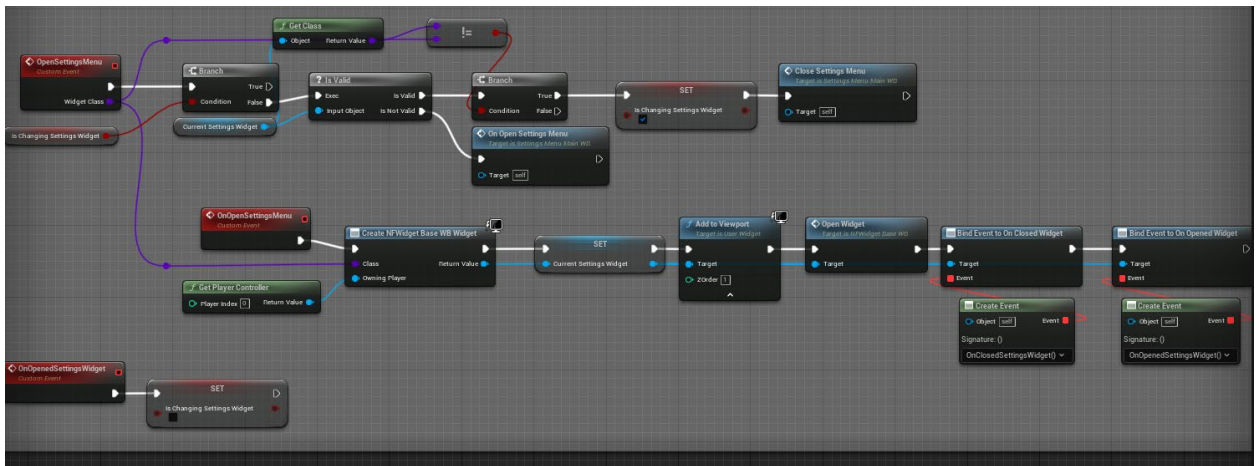


Рис. 3.6. Функція відкриття певного налаштування

Результат виконання функції представлений на рис. 3.7.

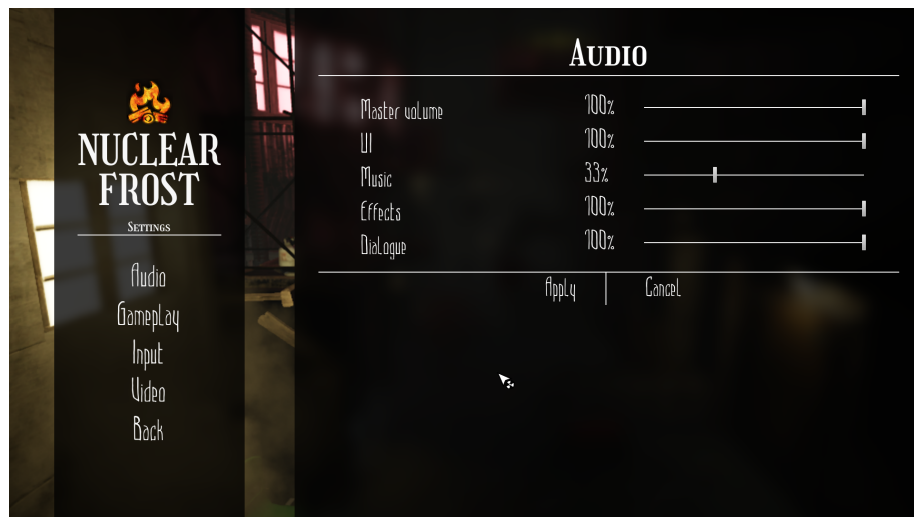


Рис. 3.7. Результат відкриття певного налаштування

3.1.1. Реалізація аудіо налаштувань

На рис 3.8. можна побачити вікно аудіо налаштувань, у якого є 5 повзунків гучності, кнопка «застосувати» та «скасувати».



Рис. 3.8. Вікно аудіо налаштувань

При створенні візуального вигляду, кожного із меню налаштувань, була застосована техніка розбиття на компоненти, кожен із повзунків (slider), полів зі списком (comboBox), строки вводу (input) та інше, були винесені в окремі компоненти (widgets), на рис. 3.9 зображена папка з такими компонентами.

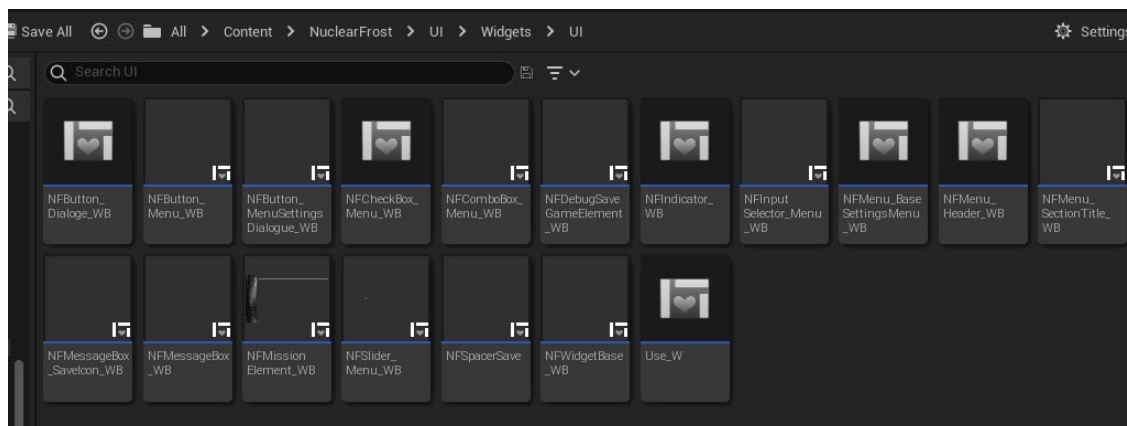


Рис. 3.9. Елементи які були винесені в окремі компоненти (widgets)

Розбиття на окремі компоненти дає змогу налаштовувати та стилізувати цей елемент 1 раз і використовувати вже готовий компонент будь-де.

Наприклад повзунок з текстом з рис. 3.10, можна побачити на рис 3.8.

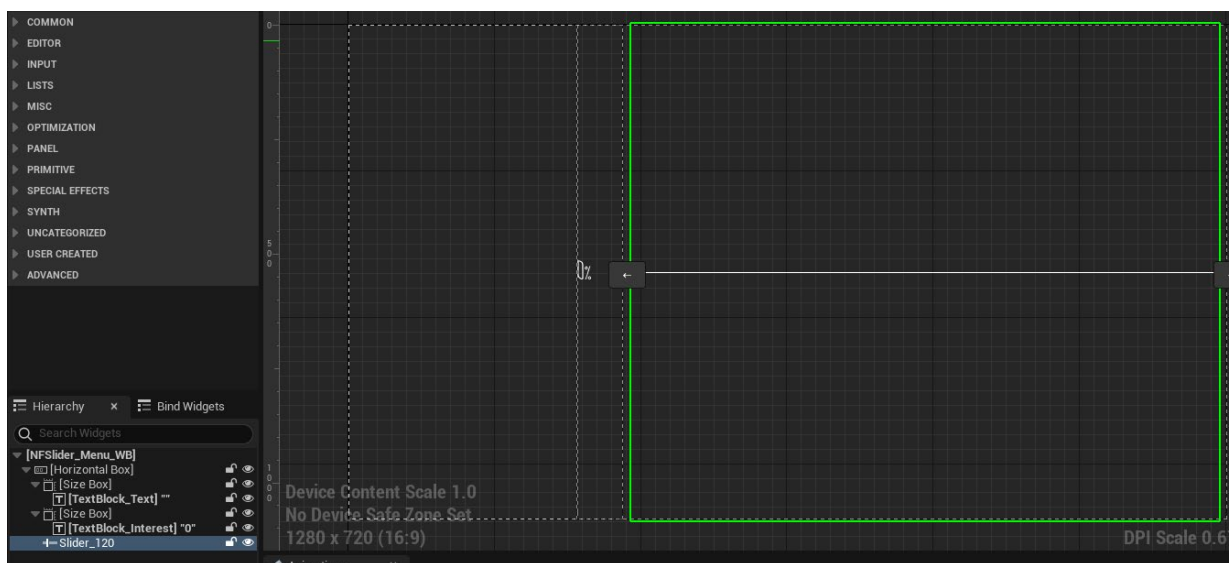


Рис. 3.10. Вигляд компонента slider

На рис. 3.11 представлена ініціалізація текстового поля в цьому компоненті.

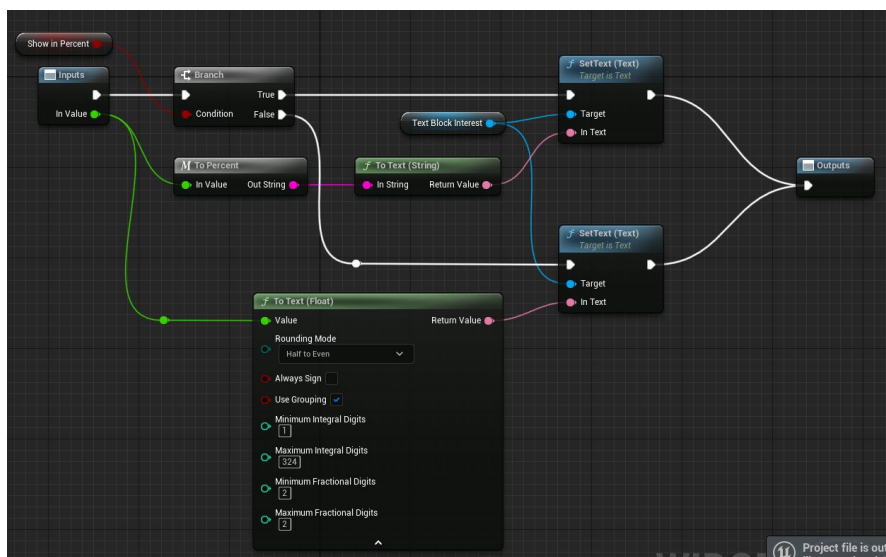


Рис. 3.11. Ініціалізації текстового поля в компоненті slider

Кожного разу, коли гравець викликає вікно з налаштуваннями, потрібно підставити збережені параметри, це називається «ініціалізація», на рис 3.12 представлена функція ініціалізації поточних звукових значень, яка викликається при відкриванні вікна.

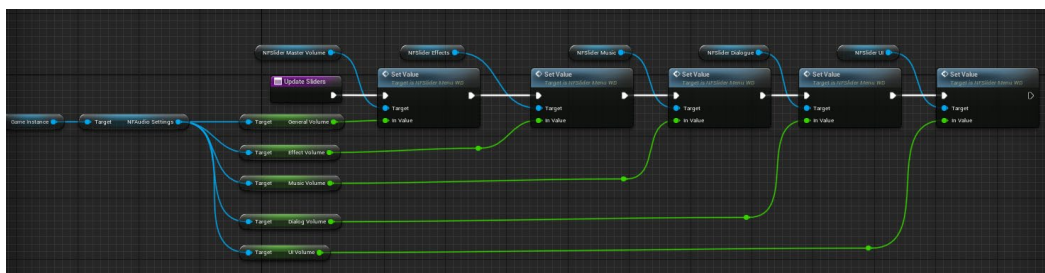


Рис. 3.12. Ініціалізації поточних звукових значень

На рис 3.13 зображена реалізація події зміни повзунка гучності, у якому записується значення повзунка до змінної.

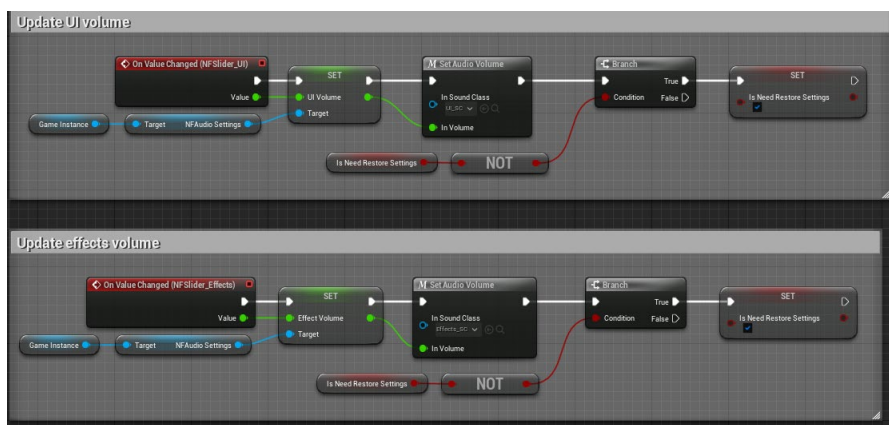


Рис. 3.13. Подія зміни повзунка гучності

3.1.2. Реалізація відео налаштувань

Логіка відкриття відео налаштувань ідентичне відкриттю аудіо налаштувань, на рис. 3.14 та 3.15 представлений візуальний вигляд вікна відео налаштувань.

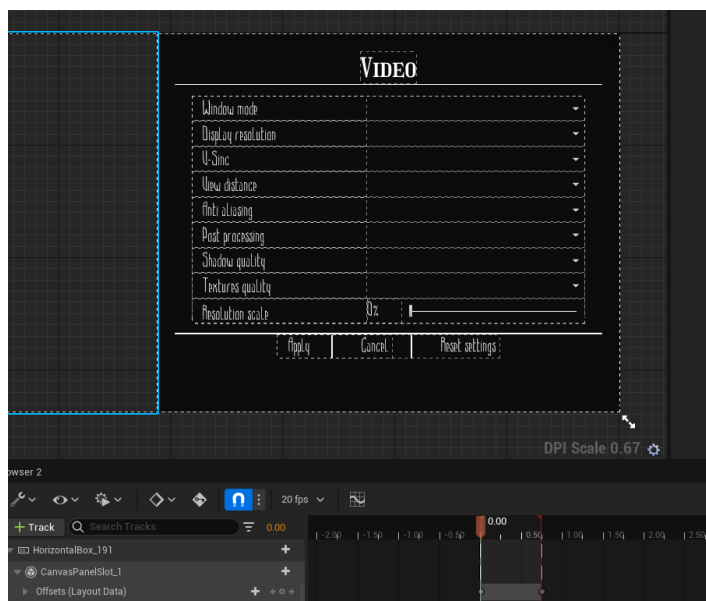


Рис. 3.14. Вікно відео налаштувань у редакторі



Рис. 3.15. Вікно відео налаштувань у грі

При ініціалізації кожного із пунктів налаштувань викликається функція Update в якій ініціалізуються поточні відео значення, метод Update один із яких представлений на рис 3.16.

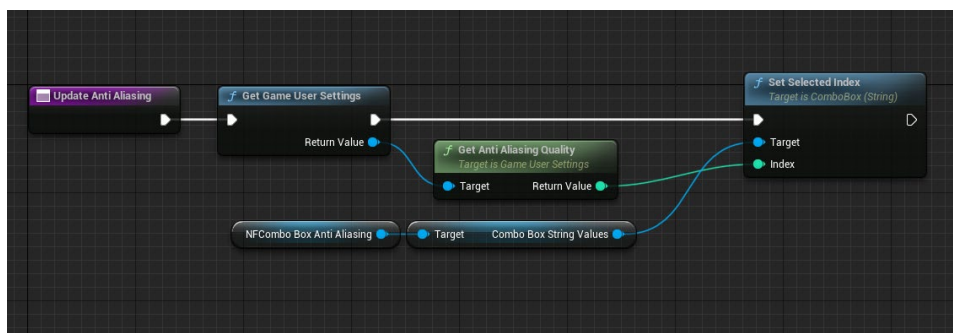


Рис. 3.16. Метод Update

Функція `GetGameUserSettings` являється вбудованою, вона зберігає у себе в середині, основні дані про відео налаштування гравця, на рис. 3.17 зображено перелік зберіганого функціонала відео налаштувань цієї функції, а на рис. 3.18 продемонстровано зміну налаштувань, та збереження їх.

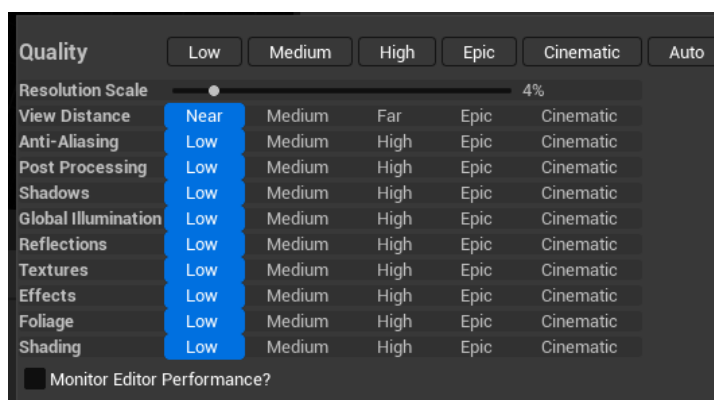


Рис. 3.17. Метод Update

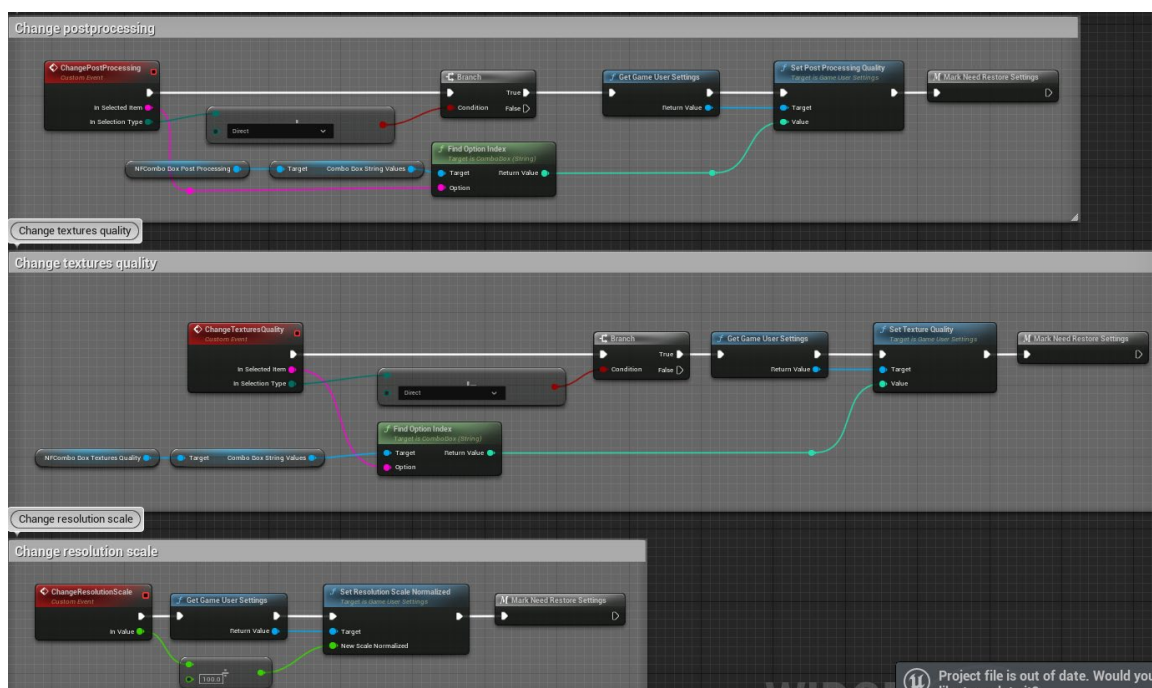


Рис. 3.18. Метод Update

3.1.3. Реалізація налаштувань клавiш

Логіка відкриття вікна не змінилась, на рис. 3.19 та 3.20 представлено вікно налаштування клавiш.

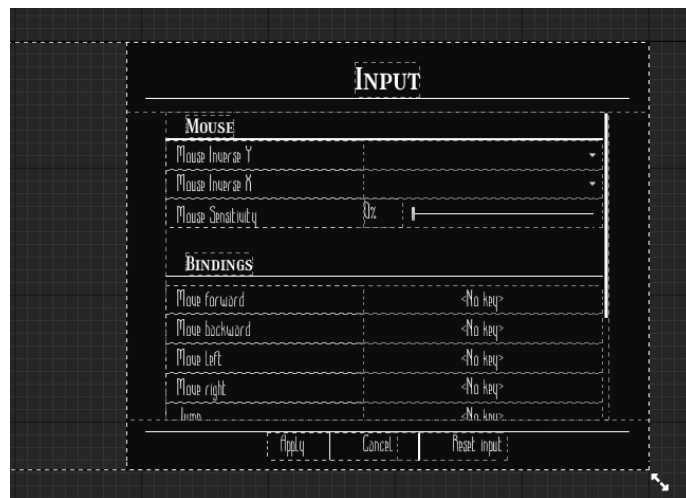


Рис. 3.19. Вікно налаштувань клавiш у редакторі



Рис. 3.20. Вікно налаштувань клавiш у грі

На рис 3.21 представлена реалізація зміни клавiші, методи «Axis Mapping» та «Action Mapping» це вбудовані функції і які записуються клавiші гри на ім'я, імена для клавiш задаються у налаштуваннях проєкт це зображено на рис. 3.22



Рис. 3.21. Реалізація зміни клавіш

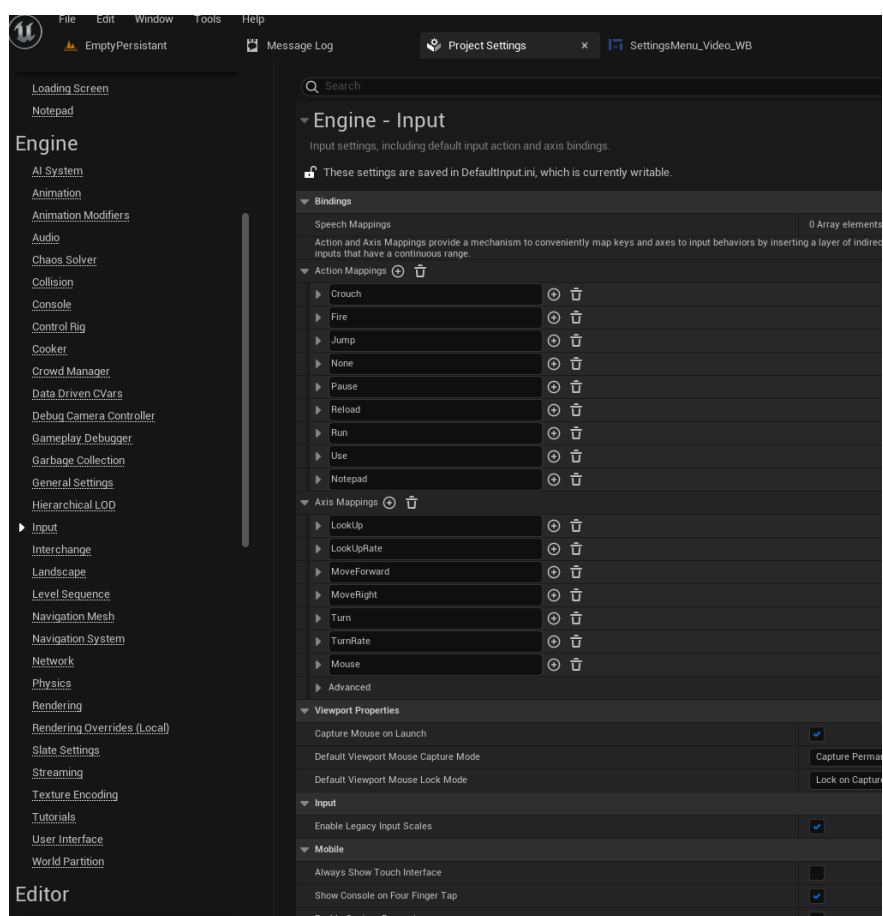


Рис. 3.22. Вигляд імен клавіш в налаштуваннях проекту

3.2. Розробка ігрового інтерфейсу

Ігровий інтерфейс користувача складається з таких елементів інтерфейсу:

1. компас;

Компас призначений для навігації гравця по ланцюжку завдань, тобто, на компасі зображуються мітки, які допомагають гравцю дійти до завдання.

2. панель статусу;

На даній панелі зображуються основні характеристики гравця на поточний час.

3. вікно підказок;

Через вікно підказок гравцю будуть виводитись підказки, якщо гравець довго не може просунутися по завданню.

4. вікно взаємодії;

Дане вікно зображується, якщо гравець навівся на об'єкт з яким можна взаємодіяти.

5. вікно місії;

В даному вікні можна побачити поточну місію та на скільки вона пройдена.

Всі вище перераховані компоненти можна побачити у редакторі на рис 3.23 та у самій грі на рис 3.24.

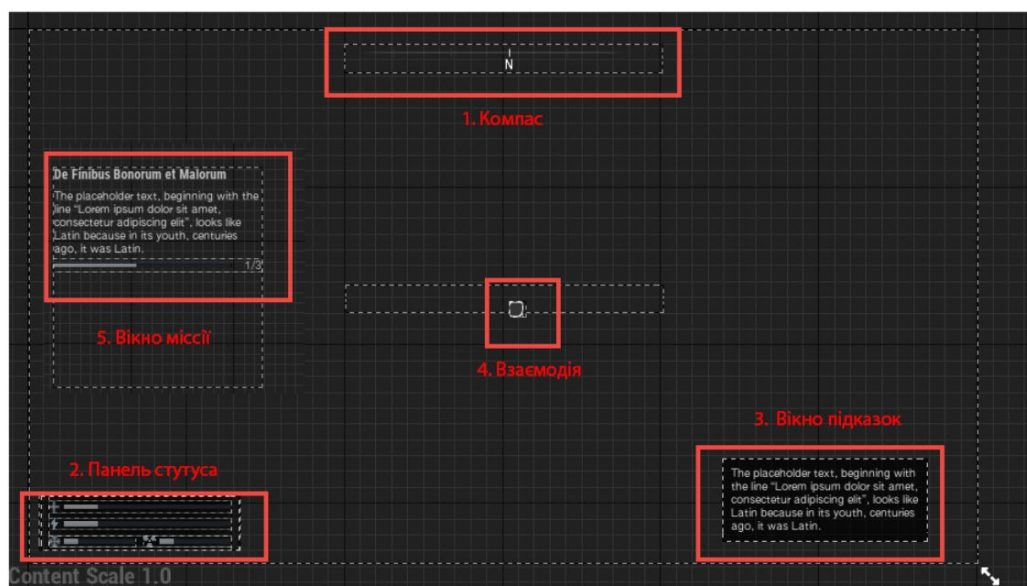
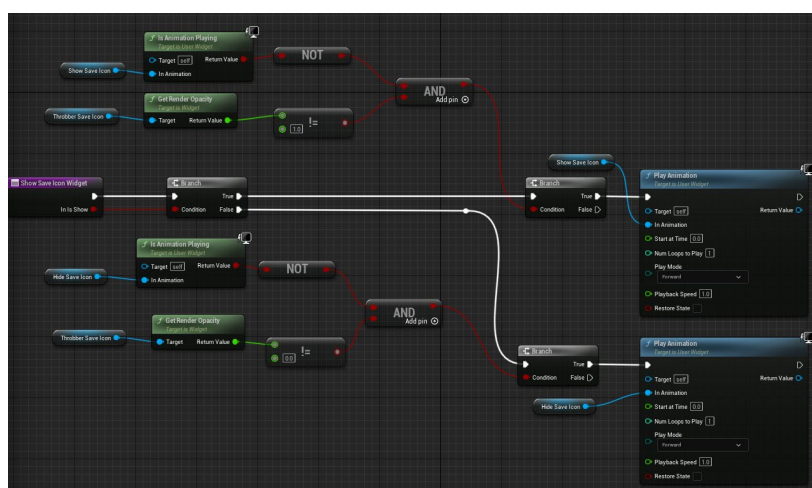
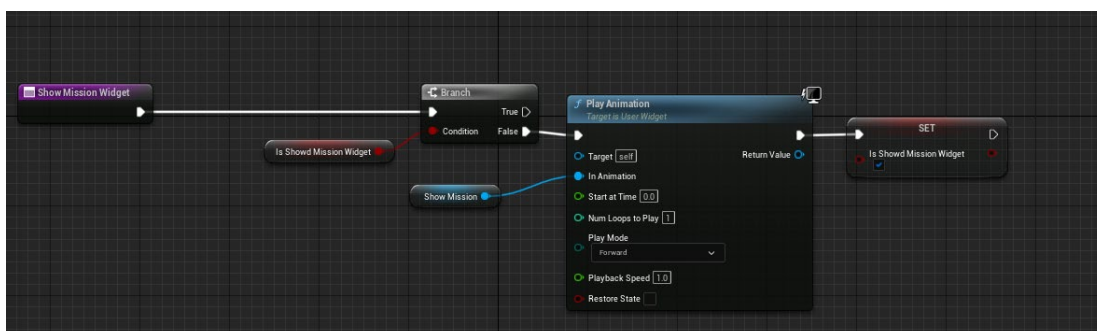
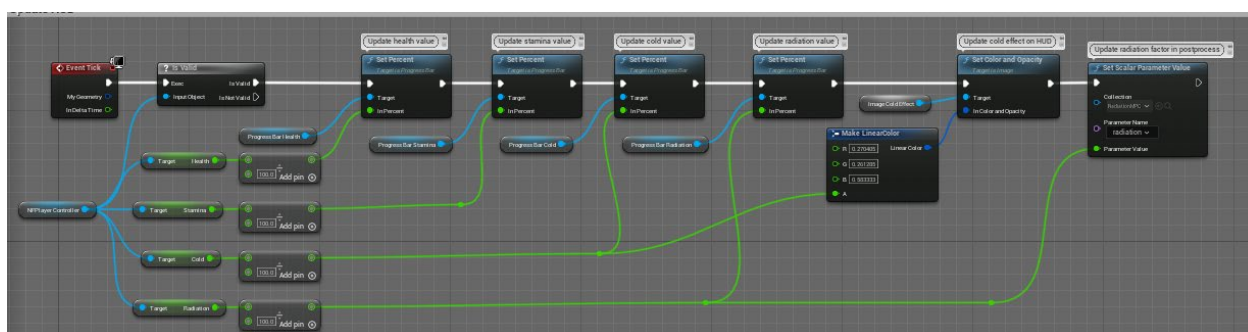


Рис. 3.23. Ігровий інтерфейс у редакторі



Рис. 3.24. Ігровий інтерфейс у грі

На рис. 3.25 представлена ініціалізація компонентів інтерфейсу.



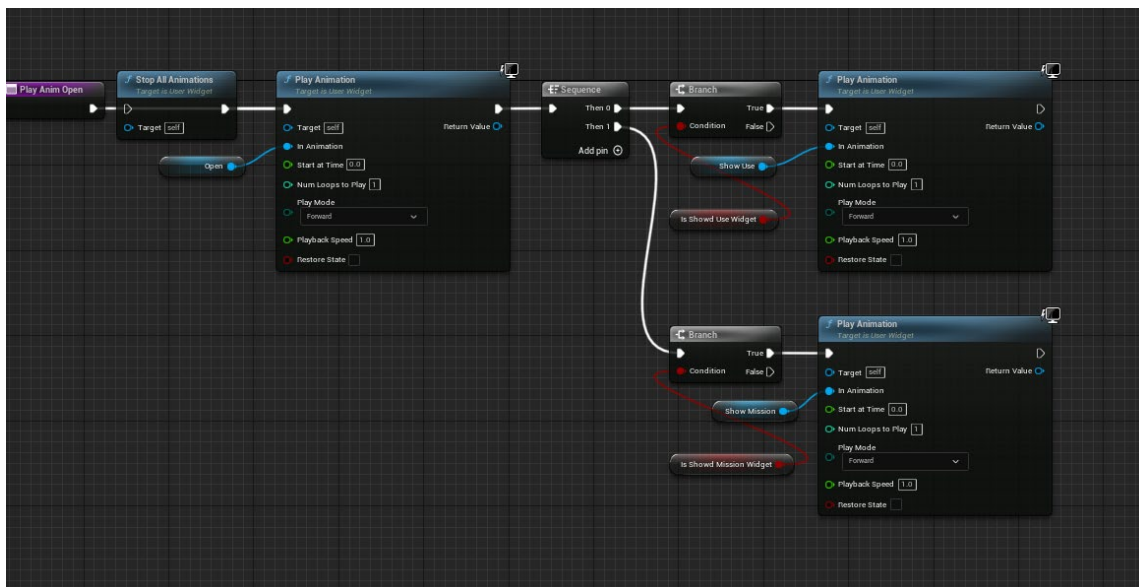


Рис. 3.25. Ініціалізація компонентів ігрового інтерфейсу

3.3. Розробка ефектів "дебафів"

3.1. Ефект холоду

Щоб реалізувати ефект холоду на екрані користувача, для економії часу, було прийнято рішення, накладати спеціальне зображення холоду і в залежності від кількості холоду, буде змінюватись прозорість зображення це зображено на рис 3.28, на рис 3.26 представлено зображення холоду, а на рис 3.27 як цей ефект проявляється у грі.

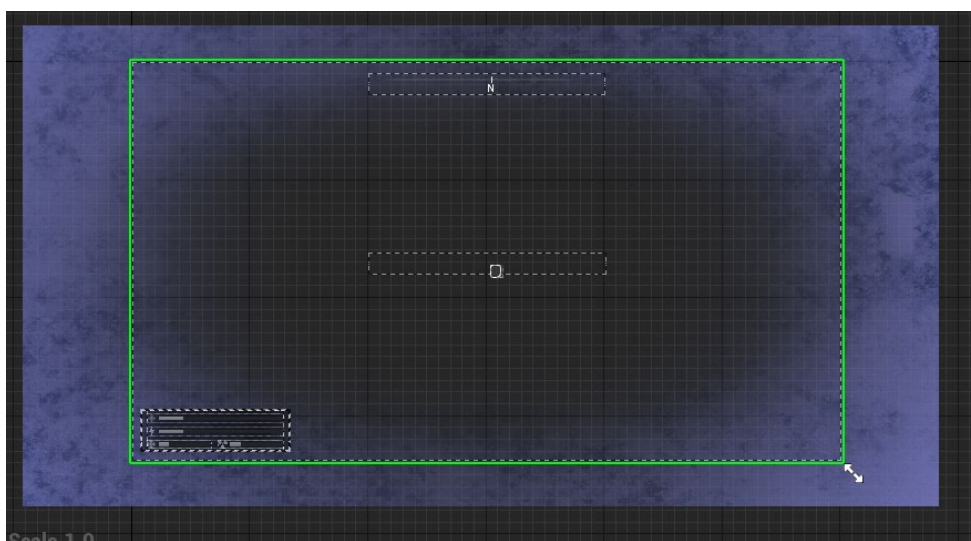


Рис. 3.26 Зображення ефекту холоду у редакторі



Рис. 3.27. Ефект холода у грі

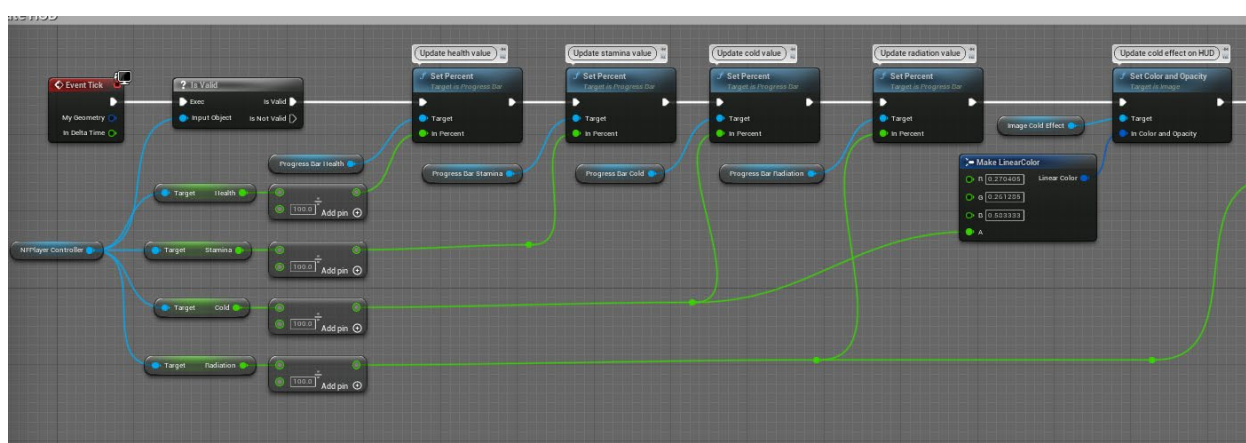


Рис. 3.27. Реалізація ефекта холода у грі

3.2. Ефект радіації

У даному проєкті, ефект радіації буде відображати глітч-ефект, для його створення в Unreal Engine, за допомогою матеріалу, можна використовувати різні техніки, такі як текстурні та тимчасові спотворення, анімації та маскування і т.д.

Першим етапом створення буде створення зони в якій буде відбуватися цей ефект, для цього там потрібно створити Post Process Volume, та перенести його на мапу, як це зображено на рис. 3.22.



Рис. 3.22. Створення Post Process Volume

Другим кроком є створення матеріалу для ефекту радіації, бо створюємо матеріал для Post Process, потрібно вказати тип матеріалу, це зображено на рис 3.23

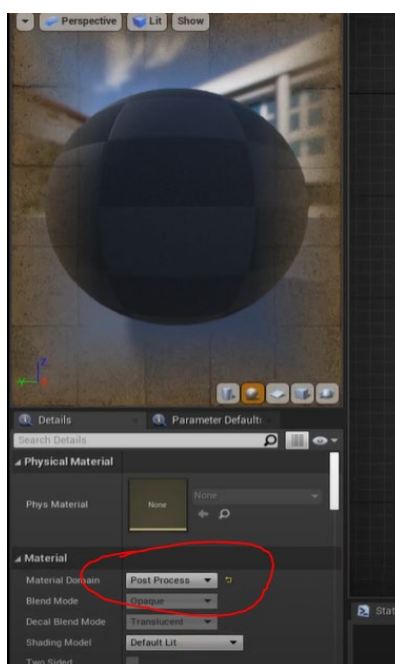


Рис. 3.23. Налаштування матеріалу для Post Process.

В наступному етапі додаємо до матеріалу змінні, які будуть відповідати за насиченість кольору та інтенсивності самого ефекту, дані кольори додається за допомогою множення до текстури екрана, це можна побачити на рис. 3.24 саме це буде додавати нам глітч ефект радіації

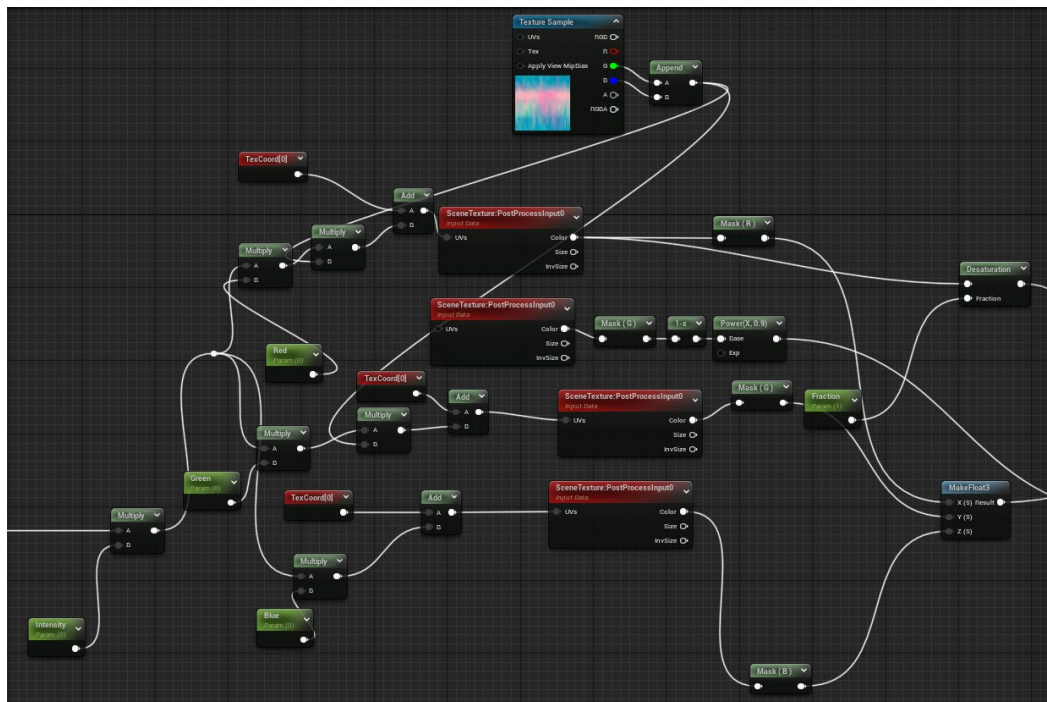


Рис. 3.23. Створення матеріалу для глітч-ефекту

На рис. 3.24 представлений створений ефект радіації у грі



Рис. 3.23. Ефект радіації у грі

Висновки до розділу

У розділі 3 була проведена розробка взаємодії з користувачем в грі, використовуючи Unreal Engine 4 (UE4) Blueprint - вбудований візуальний скриптовий інструмент для розробки функціональності гри без використання коду.

В результаті було розроблено наступні елементи:

- Меню налаштувань.

Було створено меню налаштувань, яке дозволяє гравцям змінювати різні параметри гри, такі як налаштування звуку, графіки, управління тощо. Це дозволяє гравцям налаштовувати гру на свій власний смак, забезпечуючи більше контролю над геймплеєм.

- Ігровий інтерфейс.

Було розроблено ігровий інтерфейс, що включає різноманітні елементи, такі як відображення життєвих показників гравця, рівня енергії, інформації про завдання тощо. Цей інтерфейс допомагає гравцям взаємодіяти з грою, отримувати необхідну інформацію та контролювати стан свого персонажу.

- Ефекти "дебафів".

Були розроблені ефекти "дебафів", такі як холод та ефект радіації, які можуть впливати на персонажа гравця під час гри.

В результаті розробки цих елементів взаємодії з користувачем, гра стала більш привабливою та динамічною, дозволяючи гравцям налаштовувати гру на свій власний смак

Загальні висновки

Дипломна робота присвячена реалізації взаємодії з користувачем в комп'ютерній грі NuclearFrost з використанням Unreal Engine blueprint.

У роботі був проведений збір та аналіз вимог до ігрового двигуна, описано взаємодію з користувачем в аналогічних іграх та поставлено задачі проектування інтерфейсу для взаємодії з користувачем.

Для реалізації взаємодії з користувачем були визначені функціональні вимоги та вхідні та вихідні дані. Було вибрано мову програмування Unreal Engine blueprint та описано компонентно-орієнтований підхід до розробки ігрового інтерфейсу.

Розроблено меню налаштувань, ігровий інтерфейс та ефекти "дебафів" на екрані користувача. Було проведена розробка меню налаштувань, розробка ігрового інтерфейсу та розробка ефектів "дебафів".

Результатом роботи є реалізована взаємодія з користувачем в комп'ютерній грі NuclearFrost з використанням Unreal Engine blueprint, що дозволяє користувачам налаштовувати гру під свої потреби та отримувати візуальні ефекти "дебафів". Отримані результати можуть бути використані для подальшого розвитку гри та покращення взаємодії з користувачем.

Список використаних джерел

1. Офіційна документація Unreal Engine [Електронний ресурс]
<https://docs.unrealengine.com>
2. Гід з Матеріалів у Unreal Engine [Електронний ресурс]
<https://awesometuts.com/blog/materials-unreal-engine/>
3. Відео туторіали з Unreal Engine blueprint [Електронний ресурс]
<https://www.youtube.com/watch?v=TaAwcm7YLq4&list=PLYpWZxEb36SgBqLDYYjNSHamyyWW9XCRa>
4. Langone, J. (2018). Mastering Unreal Engine 4: Build high-performance AAA games with UE 4. Packt Publishing.
5. Блог. Unreal Engine: що варто знати новачку [Електронний ресурс]
<https://vokigames.com/ua/unreal-engine-scho-var-to-znatu-novachku-pt-to-zanadto-skladne-pz-na-yakomu-stvoruutsya-skhedevru/>

Додатки

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖУЮ

Проректор Українського державного
університету науки і технологій

Анатолій РАДКЕВИЧ

18.02.23

РЕАЛІЗАЦІЯ ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ В КОМП'ЮТЕРНІЙ ГРІ З ВИКОРИСТАННЯМ UNREAL ENGINE BLUEPRINT

Технічне завдання

ЛИСТ ЗАТВЕРДЖЕННЯ

1116130. 01326-01-ЛЗ

Представники

підприємства-розробника

Завідувач кафедри КІТ

Вадим ГОРЯЧКІН

18.02.23

Керівник розробки

Олександр ІВАНОВ

18.02.23

Виконавець

Ілля ТЯГНИРЯДНО

18.02.23

Норм-контролер

Світлана ВОЛКОВА

18.02.23

2023

ЗАТВЕРДЖЕНО

1116130.01326-01-ЛЗ

**РЕАЛІЗАЦІЯ ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ В КОМП'ЮТЕРНІЙ ГРІ З
ВИКОРИСТАННЯМ UNREAL ENGINE BLUEPRINT**

Технічне завдання

1116130.01326-01

Листів 15

ЗМІСТ

1	Введення	Помилка! Закладку не визначено.
2	Підстави для розробки	Помилка! Закладку не визначено.
3	Призначення розробки	Помилка! Закладку не визначено.
4	Вимоги до програмного продукту ...	Помилка! Закладку не визначено.
4.1	Вимоги до функціональних характеристик	7
4.2	Вимоги до надійності	9
4.3	Вимоги експлуатації	9
4.4	Вимоги до складу та параметрів технічних засобів	10
4.5	Вимоги до інформаційної та програмної сумісності	10
4.6	Вимоги до маркування і упаковки	10
4.7	Вимоги до транспортування та зберігання	11
5	Вимоги до програмної документації	Помилка! Закладку не визначено.
6	Стадії та етапи розробки	Помилка! Закладку не визначено.
7	Порядок і контроль приймання	Помилка! Закладку не визначено.
8	Бібліографічний список	Помилка! Закладку не визначено.

1 ВВЕДЕННЯ

Цей проєкт прагне забезпечити всебічне розуміння ролі взаємодії з користувачем у розробці комп'ютерної гри-бойовика від першого обличчя, де доведеться протистояти загрозам ядерної земі та мутантам “Nuclear Frost” та продемонструвати можливості системи Blueprint Unreal Engine у створенні UI (User Interface) та UE (User Experience) у відеогрі.

Головна мета гри полягає в тому, щоб занурити гравця в атмосферу замерзлого світу та боротьби людей за своє існування, а також показати, наскільки може бути небезпечним застосування ядерної зброї та зневага проблем клімату.

Елементи які повинні бути реалізовані для взаємодії з користувачем в комп'ютерній грі “Nuclear Frost” :

- Створення та реалізація меню налаштувань;
- створення та реалізація ігрового інтерфейсу;
- створення та реалізація ефектів “дебафів” (debuff) на екрані користувача.

2 ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є наказ від 07.12.22 №1209ст ректора Українського державного університету науки і технологій “Про призначення наукових керівників та затвердження тем бакалаврських робіт” за спеціальністю 121 “Інженерія програмного забезпечення» факультету “Комп’ютерних технологій і систем” по кафедрі “Комп’ютерні інформаційні технології”.

Тема дипломної роботи - “РЕАЛІЗАЦІЯ ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ В КОМП’ЮТЕРНІЙ ГРІ З ВИКОРИСТАННЯМ UNREAL ENGINE BLUEPRINT ”. Керівник - доцент Іванов О. П.

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Функціональне призначення полягає в розробці та імплементації функціональності взаємодії гравців з комп'ютерною грою, створеною з використанням Unreal Engine Blueprint. Ця функціональність забезпечує зручний та інтуїтивно зрозумілий інтерфейс, який дозволить гравцям ефективно взаємодіяти з грою, щоб забезпечити найкращий геймплей та досвід користувача.

Експлуатаційне призначення – це забезпечити користувачам зрозумілий, простий інтерфейс та функціонал цього інтерфейсу, що повинно підвищити досвід користувача у грі

4 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

4.1 Вимоги до функціональних характеристик

Програмний продукт повинний бути забезпечений такими інтерфейсами користувача як:

- Меню налаштувань.

Цей інтерфейс з'являється в грі та дозволяє користувачам змінювати параметри гри, такі як гучність звуку, налаштування екрана, налаштування управління та інші. Це меню повинно бути доступним завжди і мати простий та зрозумілий інтерфейс, щоб гравець зміг швидко і легко змінювати налаштування гри на свій смак.

Пункти меню повинні бути інтуїтивно зрозумілими та розташовані так, щоб користувач міг легко знайти і змінити потрібні йому налаштування.

- Ігровий інтерфейс (game HUD)

Повинен надавати гравцям інформацію про їхній поточний стан у грі та допомагає їм приймати рішення відповідно до цієї інформації. Крім того, ігровий інтерфейс може містити інші елементи, такі як мінікарту, інвентар, і т.д.

В ігровому інтерфейсі гри будуть присутні такі елементи:

- Показник здоров'я.
- Показник енергії.
- Показник холода.
- Показник радіації.
- Інформація про зброю.
- Компас у верхній частині екрану .
- Ефект “дебафів” (debuff) на екрані користувача

Задача полягає в візуальному повідомленні гравцю, що на нього діє ефект "дебафів", тобто негативний ефект, що може зменшувати рівень життя, сили атаки, рухливості чи інші характеристики.

Вимоги для вхідних та вихідних даних для меню налаштувань.

Вхідні дані:

Налаштування звуку:

Загальна гучність, включення/вимкнення музики та звукових ефектів, всі елементи відображені за допомогою Slider у якого діапазон даних від 0 до 100

Налаштування графіки: розширені параметри екрана, якість графіки, включення/вимкнення швидкісних ефектів і т.д. Данні елементи відображені за допомогою combobox або checkbox.

Налаштування керування: призначення клавіш та кнопок.

Налаштування гри: вибір мови гри, елемент відображений за допомогою combobox

Вихідні дані:

Збережені налаштування: гучність, якість графіки, налаштування керування та налаштування гри.

Повідомлення про зміни налаштувань: якщо користувач вніс зміни в налаштування, то він може отримати повідомлення про збереження змін.

Повернення до головного меню гри: після збереження налаштувань користувач повертається до головного меню гри.

Вимоги для вхідних та вихідних даних для ігрового інтерфейсу

Вхідні дані:

Стан гравця: кількість здоров'я, енергії, рівень холоду та радіації.

Дані про місію: поточне завдання, стан поточного завдання.

Вихідні дані:

Оновлення стану: після взаємодії гравця з різними елементами гри, може змінюватися стан гравця, наприклад зменшуватися рівень життя.

Відображення повідомлень: можуть відображатися повідомлення про досягнення гравця, зміни у грі.

Вимоги для вхідних та вихідних даних для відображення ефекту “дебафів” (debuff) на екрані користувача:

Вхідні дані: дебаф, радіація або холод

Час залишку дебафу: це вказує на те, як довго ефект дебафу буде активним на користувачі.

Вихідні дані:

Звукове супроводження: може відтворюватися звуковий ефект, який вказує на наявність певного дебафу.

Анімаційні ефекти: можуть відображатися різні ефекти, блимання екрану, коливання зображення або зміна кольору.

4.2 Вимоги до надійності

Вимоги до надійності наступні: Наявність архівної копії даних про стан гравця та стан на момент збереження гри на зовнішньому носії;

4.3 Вимоги експлуатації

Програмний продукт повинен використовуватись у приміщеннях які відповідають умовам роботи ЕОМ, а саме мають такі кліматичні, санітарні та гігієнічні умови, які відповідають ДНАОП 0.00-1.13-99 (див. табл. 1).

Таблиця 1. Кліматичні умови

Пора року	Категорія робіт згідно з ГОСТ 12.01-005-88	Температура повітря, град.С	Відносна вологість повітря, %	Швидкість руху повітря, м/с
		Оптимальна	Оптимальна	Оптимальна
Холодна	легка-1-а	22-24	40-60	0,1
	легка-1-б	21-23	40-60	0,1
Тепла	легка-1-а	23-25	40-60	0,1
	легка-1-б	22-24	40-60	0,2

Працювати з програмою може людина, що має навички роботи з комп'ютером та досягла повноліття, тобто більше 18-ти років та ознайомлена з керівництвом користувача програмного продукту.

4.4 Вимоги до складу та параметрів технічних засобів

Продукт, що розробляється повинен використовуватись на ЕОМ, що мають наступні характеристики:

- Операційна система: Windows 10 64-bits OS.
- Процесор: Intel Core i3.
- Оперативна пам'ять: 4 GB RAM.
- Відеокарта: NVIDIA GTX 1650.
- DirectX: Version 12.
- Внутрішня пам'ять: 1 GB.

4.5 Вимоги до інформаційної та програмної сумісності

Програмний продукт розробляється для операційної системи Windows 10.

4.6 Вимоги до маркування і упаковки

Упаковка програмного продукту, включаючи документацію повинна бути захищена від пошкоджень різного роду (механічних, кліматичних).

На упаковці повинно бути вказана назва продукту, номер версії, мінімальні системні вимоги.

На зворотній стороні упаковки вказується розробник та його юридична адреса.

<p style="text-align: center;">Програмний продукт</p> <p style="text-align: center;">РЕАЛІЗАЦІЯ ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ В КОМП'ЮТЕРНІЙ ГРІ З ВИКОРИСТАННЯМ UNREAL ENGINE BLUEPRINT</p> <p style="text-align: center;">Мінімальні системні вимоги:</p> <ul style="list-style-type: none"> - Операційна система: Windows 10 64-bits OS - Процесор: Intel Core i3 - Оперативна пам'ять: 4 GB RAM - Відеокарта: NVIDIA GTX 1650 - DirectX: Version 12 - Внутрішня пам'ять: 1 GB 	<p style="text-align: center;">Програмний продукт</p> <p style="text-align: center;">РЕАЛІЗАЦІЯ ВЗАЄМОДІЇ З КОРИСТУВАЧЕМ В КОМП'ЮТЕРНІЙ ГРІ З ВИКОРИСТАННЯМ UNREAL ENGINE BLUEPRINT</p> <p style="text-align: center;">Розробник: Тягнирядно І.А. Кафедра: "КІТ", УДУНТ</p> <p style="text-align: center;">м. Дніпро, вул. Лазаряна 2</p> <p style="text-align: center;">2023</p>
---	--

4.7 Вимоги до транспортування та зберігання

Транспортування повинне забезпечувати збереження програмного продукту його цілісність і запобігання несанкціонованого доступу до нього. Програмний виріб міститься на фізичному носії та переданий через microUSB порт.

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу документації мають входити:

- специфікація;
- текст програми;
- опис програми;
- керівництво користувача.

Вся документація програмного додатку повинна задовольняти вимоги до програмної документації.

.

6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Таблиці 1. – Стадії та етапи розробки

Стадія	Зміст	Строки виконання
Технічне завдання	Постановка задачі, збір інформації, вибір та обґрунтування критеріїв розробки. Попередній вибір методів рішення задач. Визначення вимог до технічних засобів. Узгодження і затвердження технічного завдання.	05.12.22 -18.02.23
Робочий проект	Програмування та відлагодження програми.	12.02.23 - 22.05.23
	Тестування програми	23.05.23 - 30.05.23
	Розробка, узгодження і затвердження програмної документації.	30.05.23 - 15.06.23

7 ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ

Контроль за виконанням роботи здійснює керівник розробки доц. Іванов О. П.

Прийом здійснюється комісією у складі:

- Горячкін В. М. (керівник підрозділу);
- Іванов О. П. (керівник розробки).

8 БІБЛІОГРАФІЧНИЙ СПИСОК

1. Івченко, Ю.М. Основи стандартизації програмних систем: методичні вказівки до дипломного проектування та лабораторних робіт/уклад.: Ю.М. Івченко, В. І. Шинкаренко, В. Г. Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна, 2009. - 38 с