

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних систем та мереж

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему:

Комп'ютерна система навчання керування
дроном в ігровій формі

Виконав: студент IV курсу, групи СІс-42

спеціальності 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

(підпис)

Шелофастов Б.О.

(прізвище та ініціали)

Керівник

(підпис)

Стадник Н.Б.

(прізвище та ініціали)

Нормоконтроль

(підпис)

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль -2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.
(прізвище та ініціали)

«__» _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

Студенту Шелофастову Богдану Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система навчання керування дроном в ігровій формі

Керівник роботи Стадник Наталія Богданівна., к.т.н..
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «26» 04 2024 року № 4/7-468

2. Термін подання студентом завершеної роботи 24.06. 2024 р.

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ.

1. Аналіз технічного завдання.

2. Проектна частина.

3. Практична частина.

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

1. Будова квадрокоптера

2. Схема роботи квадрокоптера

3. Схема гри та порядок дій при навчанні оператора

4. Скріншот роботи симулятора

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>			

7. Дата видачі завдання _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Ознайомлення з завданням до кваліфікаційної роботи</i>	27.04 – 29.04	Виконано
2.	<i>Розробка технічного завдання</i>	30.04 – 02.05	Виконано
3.	<i>Підбір джерел про ігрові системи керування дронами</i>	03.05 – 08.05	Виконано
4.	<i>Опрацювання літературних джерел</i>	09.05 – 12.05	Виконано
5.	<i>Виконання дослідження щодо розробки системи керування дронами в ігровій формі</i>	13.05 – 16.05	Виконано
6.	<i>Написання програмного коду</i>	17.05 – 20.05	Виконано
7.	<i>Оформлення розділу «Аналіз технічного завдання»</i>	21.05 – 27.05	Виконано
8.	<i>Оформлення розділу «Проектна частина»</i>	28.05 – 31.05	Виконано
9.	<i>Оформлення розділу «Практична частина»</i>	01.06 – 05.06	Виконано
10.	<i>Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»</i>	06.06 – 08.06	Виконано
11.	<i>Оформлення кваліфікаційної роботи</i>	08.06 – 11.06	Виконано
12.	<i>Нормоконтроль</i>	09.06 – 12.06	Виконано
13.	<i>Перевірка на плагіат</i>	11.06 – 14.06	Виконано
14.	<i>Попередній захист кваліфікаційної роботи</i>	14.06 – 17.06	Виконано
15.	<i>Захист кваліфікаційної роботи</i>	25.06	

Студент

_____ (підпис)

Шелофастов Б.О.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Стадник Н.Б.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Комп'ютерна система навчання керування дроном в ігровій формі // Кваліфікаційна робота бакалавра // Шелофастов Богдан Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІс-42 // Тернопіль, 2024 // с. – 55, рис. – 37, табл. – 1, аркушів А1 – 4, бібліогр. – 21.

Ключові слова: АССЕТ, ДРОН, ІГРОВИЙ РУШІЙ, ТЕКСТУРА, SKYBOX, TERRAIN, UNITY

Кваліфікаційна робота присвячена розробці ігрової комп'ютеризованої системи для навчання управлінню дроном.

На основі проведеного аналізу роботи квадрокоптеру було створено концепцію інтерактивного тренування. Розглянуто програмний рушій Unity. Проведено огляд основних його інструментів. При реалізації проєкту застосовано значну кількість інструментів та готових вбудованих бібліотек. Для передачі атмосфери гри використано інструмент Skybox, Розроблено структури гри, побудована схема її роботи, наведено діаграму використання проєкту. Розроблено алгоритм роботи методу пошуку оптимального шляху руху дрону. Реалізовано різні ефекти та анімаційні елементи для надання більшої реалістичності. Проєкт змонтовано у повноцінний програмний продукт та протестовано. Результати проведеного тестування дають змогу стверджувати, що мета виконання роботи досягнута повністю.

Розробка має практичну цінність, дає можливість навчитися керувати дроном в ігровій формі. Може бути застосована як симулятор для навчання операторів дронів.

ANNOTATION

Computer system of learning to control a drone in a game form // Bachelor thesis // Shelofastov Bohdan // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Systems and Nets // Ternopil, 2024 // p.- 55, fig. – 37, table. – 1, Sheets A1 – 4, Ref. – 21.

Keywords: ASSET, DRONE, GAME ENGINE, TEXTURE, SKYBOX, TERRAIN, UNITY

The qualification work deals with the creation of a game computerized system for learning how to control a drone.

The concept of interactive training was created on the basis of the analysis of the operation of the quadrocopter. The software driver Unity is considered. An overview of its main tools was conducted. During the implementation of the project, a significant number of tools and ready-made built-in libraries were used. The Skybox tool was used to convey the atmosphere of the game. The structure of the game was developed, a diagram of its operation was built, and a diagram of the use of the project was given. The algorithm of the method of finding the optimal path of the drone movement has been developed. Various effects and animation elements have been implemented to provide greater realism. The project has been assembled into a full-fledged software product and tested. The results of the conducted testing make it possible to state that the purpose of the work has been fully achieved.

The development has practical value, it provides an opportunity to learn how to control a drone in a playful way. Can be used as a simulator for training drone operators.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	11
1.1 Конструкція дрону	11
1.2 Опис розробки	15
1.3 Ігровий рушій	15
1.4 Знайомство з інтерфейсом, основні інструменти	16
1.5 Інструменти створення ландшафту	18
РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА	22
2.1 Особливості представлення моделі дрона.....	22
2.2 Структура гри	24
2.3 SkyBox у Unity	27
2.4 Створення ландшафту гри.....	29
2.5 Налаштування Skybox.....	32
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА	34
3.1 Створення дрону.....	34
3.2 Реалізація Splash Screen	35
3.3 Рух камери і Standard assets.....	36
3.4 Реалізація ефектів.....	39
3.5 Реалізація вікна UI.....	42
3.6 Створення анімації за допомогою інструмента TimeLine	44
3.7 Монтування гри.....	46
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	48
4.1 Стихійні лиха та їх класифікація.	48

					КС КРБ 123.328.00.00 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.	Шелофастов Б.				Літ.	Арк.	Аркушів		
Керівник.	Стадник Н.Б.								
Реценз.					ТНТУ, каф. КС, гр. СІс-42				
Н. Контр.									
Затверд.	Осухівська Г.М								

4.2 Соціальне значення охорони праці	50
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТКИ	
Додаток А Технічне завдання	

					<i>КС КРБ 123.328.00.00 ПЗ</i>	Арк.
						7
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

UI (User interface) – інтерфейс користувача.

Unity – кросплатформовий інструмент для розроблення відеоігор і застосунків, і рушій, на якому вони працюють.

Skybox (Скайбокс) – метод створення фону, щоб рівень відеогри виглядав більшим, ніж він є насправді.

БПЛА – безпілотний літальний апарат.

Ігровий ассет / Ігровий ресурс (Game Asset) – однотипний компонент відеогри, який може бути графічним, звуковим супроводом або скриптом.

Катсцена – визначається як візуальне представлення гри, де відбувається розміщення та організація об'єктів, графічних елементів та логіки гри.

Колайдер – невидима спрощена форма об'єкта, для розрахунку зіткнень з іншими об'єктами.

ОС – операційна система.

ПЗ – програмне забезпечення.

					КС КРБ 123.328.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Ігрова індустрія - один із найбільших сегментів індустрій розваг. Останнім часом індустрія відеоігор, через свою постійно збільшуючу швидкість, стала значно переважати інші різновиди розважального штибу.

Написання відеоігор є складним комплексним процесом, котрий потребує злагодженої спільної праці команди розробників. Фахівці застосовують різноманітні засоби та технології, в тому числі спеціалізовані програмні рушії, 3D моделювання, анімацію, побудова текстури та спецефекти для формування візуально привабливих та реалістичних об'єктів.

Ігрова індустрія володіє значним потенціалом для застосування в різних галузях, зокрема освітні технології, медична сфера, військове мистецтво та ін. Зараз надзвичайно корисними і популярними є різного роду симулятори для навчання пілотів (операторів) літальних апаратів, в т.ч. і БПЛА. В даний час дрони успішно використовуються як для цивільних, так і, особливо, військових потреб. Актуальність таких БПЛА упродовж тривалого часу не викликає сумнівів. Саме тому необхідні фахівці, котрі вміють ними дистанційно керувати.

Успішними операторами дронів стають поціновувачі відеоігор. Цей факт обумовлений тим що, ігри, властиво, повторюють надзвичайно реалістично фізику реальних польотів. Керування БПЛА багато в чому схоже на ігрові програми. Перед початком польоту на справжньому дроні чи мультикоптері, особа повинна пройти курс навчання на симуляторі. Такі польоти дійсно покращують навички операторів, навіть у досвідчених. Хоча для цього ігровий симулятор повинен бути якісним, з реальною передачею фізики та особливостей польоту.

Мета роботи – розробити модель польоту дрону, за допомогою графічного рушія, де користувач міг би навчитися керувати дроном в ігровій формі.

Завдання, необхідні для досягнення даної мети:

- розглянути конструкції дронів;
- ознайомитися з роботою програмного рушія Unity;

					КС КРБ 123.328.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- вивчити можливості його інструментів та їх властивостей;
- розробити структуру гри;
- спроектувати та впровадити всі її складові елементи;
- протестувати розробку на предмет можливості навчання оператора дрона.

					<i>КС КРБ 123.328.00.00 ПЗ</i>	<i>Арк.</i>
						10
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Конструкція дрону

Різні моделі дронів можуть бути різних конструкцій (проста будова призначена, як правило, для розважальної індустрії), проте до їх складу можуть входити і складні елементи, котрі дають змогу перетворити ці БПЛА дійсно у вузькопрофільні пристрої (особливо це актуально для військових потреб) [1]. Внутрішня будова квадрокоптерів може змінюватися в залежності від їх моделей, незважаючи на це, його базові компоненти завжди є постійними [2]. У загальному в кожному дроні можна визначити такі основні частини (рис. 1.1):

- основа (рама);
- акумулятор;
- двигун чи двигуни;
- регулятори обертів;
- пропелер чи пропелери;
- мікроконтролер польоту.

Основна його частина – це рама, до котрої прикріплюються усі решту конструктивних елементів. Основна задача при її виготовленні (з рідких сплавів чи полімерів), що пристрій на її основі повинен бути легким, ударостійким і, за можливості, довговічним та ергономічним.

Квадрокоптер володіє чотирма двигунами із пропелерами, котрі приволять його у рух зі швидкістю, котру генерують регулятори обертів. Через польотний контролер, який опрацьовує команди від оператора, проходить регулювання цих обертів. Пропелери та двигуни є тими елементами, які найчастіше виходять з ладу і найшвидше зношуються, тому що в польоті саме на них попадає основне навантаження.

					КС КРБ 123.328.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Шелофастов Б.				<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник.</i>		Стадник Н.Б.						
<i>Реценз.</i>								
<i>Н. Контр.</i>								
<i>Затверд.</i>		Осухівська Г.М						
						ТНТУ, каф. КС, гр. СІс-42		

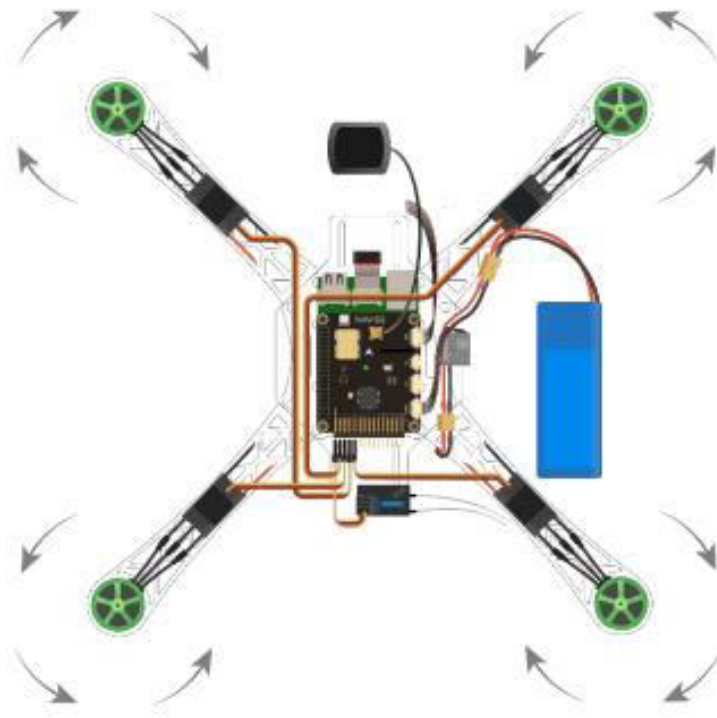


Рисунок 1.1 – Типова будова безпілотника

«Мозком» БПЛА є контролер польоту (рис. 1.2). Через встановлені датчики він може розпізнавати сигнали, котрі поступають від оператора з пульту дистанційного керування. Універсальність пристрою як раз і визначається числом сигналів, що він здатен опрацьовувати.

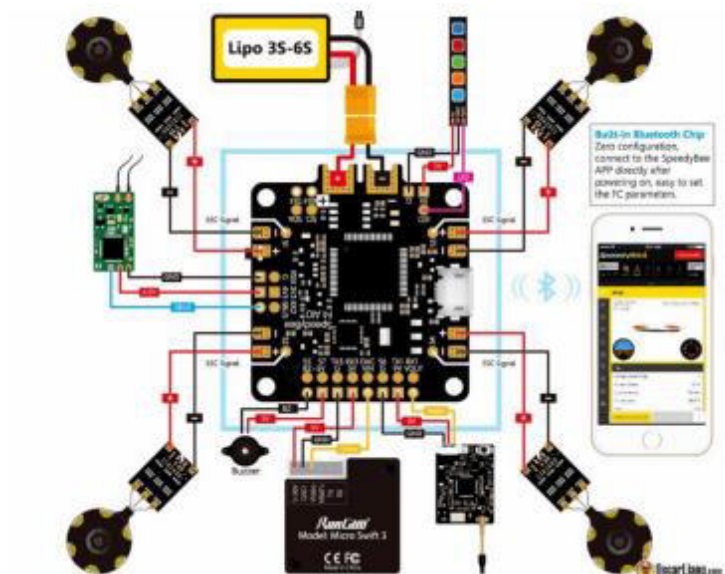


Рисунок 1.2 – Контролер (по центру)

					КС КРБ 123.328.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Контролер для стабільної його роботи потребує віброізоляції. Як правило, чим вона краща, тим стабільніше буде політ квадрокоптера.

Польотний контролер містить комбінації різних датчиків (GPS, лазерного далекоміра, гіроскопу, акселерометра, лідара та ін.), котрі передають йому покази. Завдяки передавачу, що встановлений на рамі, забезпечується зворотній зв'язок. У залежності від показів датчиків оператор може задавати ті чи інші польотні параметри безпілотної. Контролер польоту здатен самостійно змінювати польотні, не потребуючи при цьому операторської допомоги.

Аналогічно до більшості мобільних пристроїв, безпілотної потребують живлення від акумуляторів. Вони йдуть в комплекті з окремим контролером, котрий застосовується як пульт управління БПЛА. Хоча фахівці і стараються спроектувати міцну конструкцію пристрою та створити інтелектуальні системи, проте це не врятує дрон від оператора з малим досвідом роботи.

У залежності від кількості гвинтів і, відповідно, двигунів БПЛА можна класифікувати так [1]:

– бікоптер і трікоптер, мають два та три гвинти відповідно, обертання яких відбувається у протилежних напрямках з метою скомпенсування обертального моменту (рис. 1.3);



Рис. 1.3 – Бікоптер та трікоптер

– квадрокоптер – безпілотної з чотирма двигунами (рис. 1.4), який може нести корисний вантаж;

					КС КРБ 123.328.00.00 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 1.4 – Квадрокоптер

– гексакоптер та октокоптер, містять шість та вісім двигунів відповідно (рис. 1.6). Вони є значно надійніші за попередні, але і дорожчі;



Рис. 1.5 – Гексакоптер і октокоптер

– багатомоторний мультикоптер, має більше восьми двигунів, призначені для виконання спеціальних завдань (рис. 1.6).

					КС КРБ 123.328.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14



Рисунок 1.6 – Багатомоторний мультикоптер

1.2 Опис розробки

Наше завдання — це створення симуляції польоту дрона в тривимірному просторі, геймплей гри буде виглядати так: наявний дрон, який злітає і на шляху його польоту будуть певні перешкоди та ворожі дрони, котрі заважатимуть нашому дрону набирати очки.

Суть гри в тому, щоб пройти локацію, минаючи всі перешкоди і заробити якнайбільше ігрових очок. У самого дрона будуть вбудовані гармати, які допоможуть йому знищувати ворожі дрони. Дизайн гри буде виконаний у космічному стилі, щоб гра була незвичайною, привабливою та ефектною.

Для розробки системи навчання дроном буде використано графічний рушій Unity [3], мова програмування C#, що дозволить написати скрипти для навчальної системи [4]. C# - є сучасною об'єктно-орієнтованою мовою, входить до сімейства мов C, і буде добре знайомою тому, хто вже колись працював із C, C++, Java чи JavaScript [5].

1.3 Ігровий рушій

На даний момент найбільш популярним графічним рушієм є Unity [6], його простота і кросплатформеність привернула велику кількість розробників ігор,

					КС КРБ 123.328.00.00 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

котрі тільки починають свій шлях. Unity - це кросплатформовий ігровий рушій, в якому об'єднані різноманітні програмні інструментами, котрі застосовуються при створенні ПЗ. Він містить і текстовий редактор, і компілятор, і відлагоджувач і т.д. Водночас через його застосування, процес створення ігор в Unity є максимально спрощеним та комфортним, а мультиплатформенність рушія дає змогу гравцям охопити якнайбільшу кількість ігрових платформ та ОС [7].

Ще одна з переваг графічного рушія Unity - це надзвичайно велика бібліотека асетів та плагінів, при допомозі яких властиво процес розробки гри власне можна значно пришвидшити. Вони можуть імпортуватися та експортуватися, можна додавати в гру цілі заготовки - рівні, ворогів, патерни поведінки штучного інтелекту тощо. Unity поширюється безкоштовно, що відчиняє перед незалежними фахівцями двері до індустрії ігор.

Безперечно, наявні обмеження: безкоштовна версія рушія демонструє логотип Unity перед початком гри, а сам проект, властиво, котрий створений при її допомозі, не повинен давати прибуток розробнику більше \$100 тисяч на рік [8].

1.4 Знайомство з інтерфейсом, основні інструменти

Для розробки нашої гри розберемо основні поняття про графічний рушій Unity.

Unity використовує мову програмування C# для створення скриптів, тобто для створення нашої гри потрібно знати мову програмування C#, знання цієї мови допоможе створювати скрипти без особливих проблем.

При створенні нового проекту в Unity з'являється вікно програми, де ми можемо додавати об'єкти для своєї гри, у верхній лівій частині цього поля є такі інструменти.

Tool - дозволяє пересуватися по сцені, на якій збирається сама гра.

Move tool - за допомогою даного інструменту можна пересувати об'єкти в сцені.

Rotate Tool – обертає обраний об'єкт у тривимірному просторі.

					КС КРБ 123.328.00.00 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Scale Tool – інструмент змінює розмір об'єкта.

Rect Tool – інструмент може використовуватися для переміщення, зміни розміру об'єкта [9].

Інтерфейс вікна рушія Unity наведено на рис. 1.7.

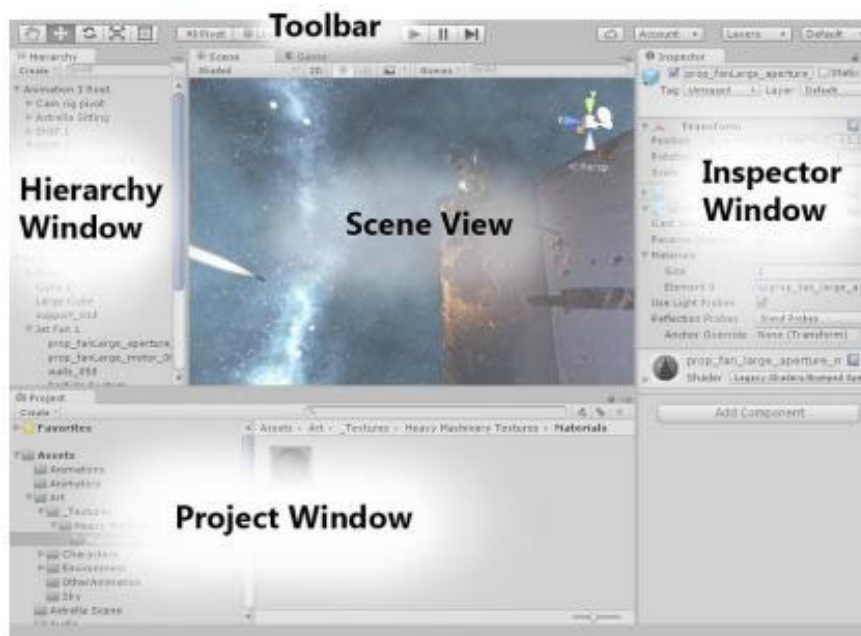


Рисунок 1.7 – Вікно розробки

У лівій частині є назва сцени і які об'єкти зберігає у собі ця сцена (рис. 1.8), в даному разі зберігається камера і освітлення.

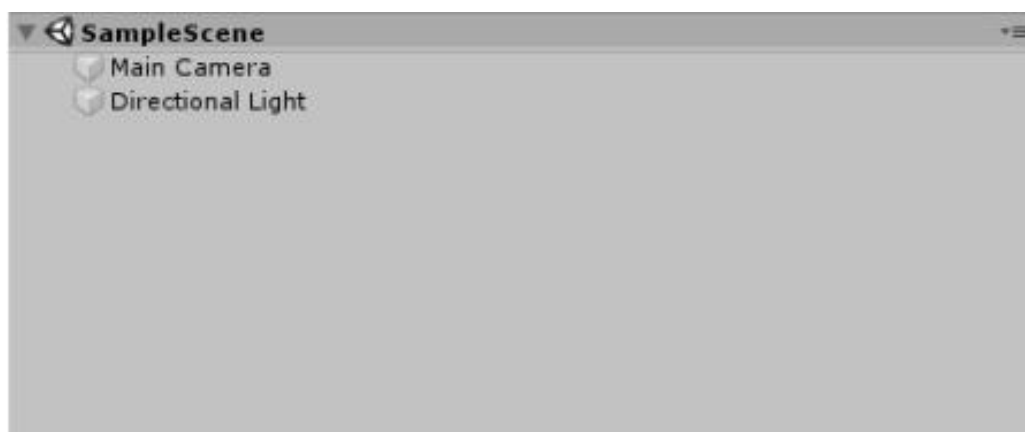


Рисунок 1.8 – Вікно сцени

Scripts (сценарії) – це окремі послідовності дій, створені для

					КС КРБ 123.328.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

автоматичного виконання завдання.

Сцена в Unity використовується для зберігання ігрових об'єктів, для ігрового процесу, так само сцен може бути багато в залежності від потреби самої гри [10].

Prefabs - це особливий тип асетів, що дозволяє зберігати весь GameObject з усіма компонентами та значеннями властивостей.

Materials - це визначення того, як повинна відображатися поверхня, включаючи посилання на текстури, інформацію про мозаїку, колірні відтінки та багато іншого.

Main Camera - це камера, що відображає ігрове поле, який бачить гравець. Для того, щоб запустити проект, використовується кнопка Play, щоб зупинити – кнопка Pause.

У правій частині знаходиться вікно Inspector. Цей елемент дозволяє змінювати властивості самого об'єкта, а також змінювати його положення, розмір, додавати скрипти і т.д.

Розберемо сам Inspector [10].

Transform відповідає за позицію об'єкта у тривимірному просторі.

Mesh Filter - зберігає властивість і передає параметр Mesh Render.

Mesh Render- відображає об'єкт.

Box Collider – визначають форму об'єкта з метою фізичних зіткнень.

1.5 Інструменти створення ландшафту

Для створення ландшафту використовується такий інструмент як Terrain, компонент terrain надає усі інструменти, необхідні для редагування ландшафту (рис. 1.9). Усі панелі інструментів, за винятком інструмента розміщення дерев та панелі налаштувань, надає набір «пензлів» та параметри для налаштування розміру та прозорості пензля, так само як інструменти малювання зі звичайного редактора зображень. Це дозволяє створювати ландшафт подібно до інструменту малювання [11].

					<i>КС КРБ 123.328.00.00 ПЗ</i>	Арк.
						18
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

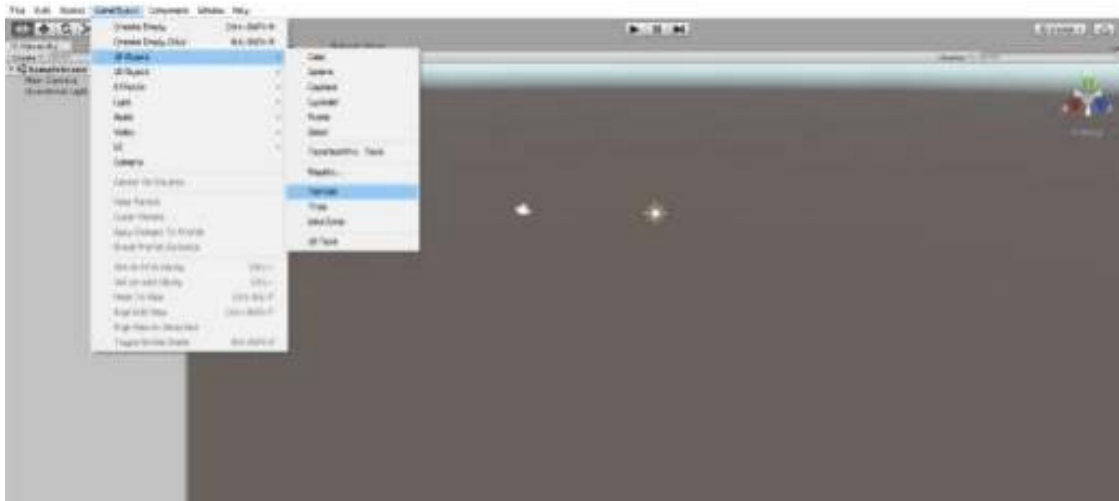


Рисунок 1.9 – Створення terrain

Для того, щоб створити Terrain потрібно зайти в розділ GameObject – 3D Object – Terrain. Як бачимо на рис. 1.10 ми створили об'єкт terrain.

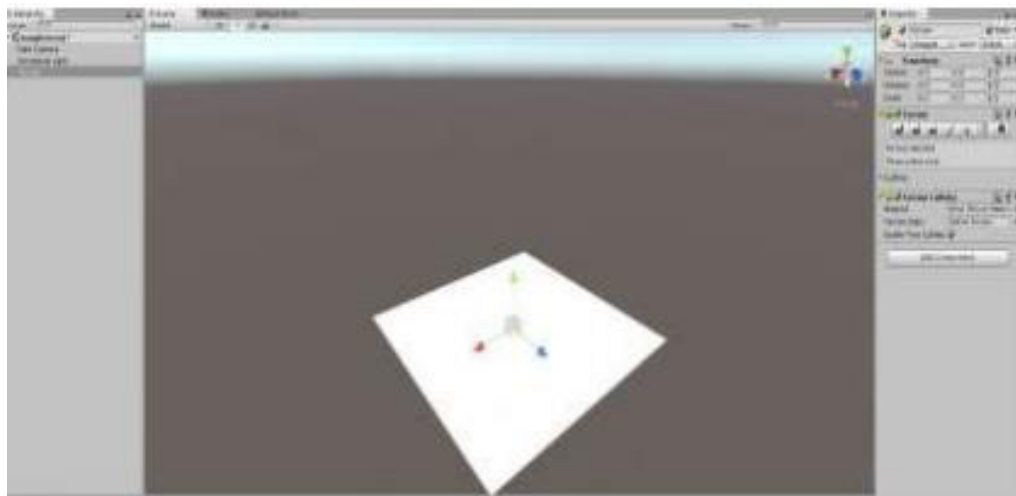


Рисунок 1.10 – Terrain

Розглянемо властивості інструменту Terrain. Вони поділяються на 5 розділів.

Base Terrain знаходиться кілька властивостей та параметрів. Такі як операція Draw включає або вимикає рендеринг ландшафту, Pixel Error визначає значення точності перетворення між картами трейну. Вищі значення становлять меншу точність, а менші значення будуть надмірної обробки. Base Map dist –

максимальна відстань, на якій текстури ландшафту будуть відображатися з повною роздільною здатністю. Cast Shadows дозволяє вмикати та вимикати тіні. Materials тут встановлюється матеріал, який використовуватиметься для рендерингу ландшафту. Це вплине на те, як інтерпретуються колірні канали текстури ландшафту. Тут можна вибрати [11]:

- Built In Standard представляє матеріал, заснований на фізичному рендерингу, введеному в Unity 5. Якщо вибрано цей параметр, то для кожного шару накладання можна використовувати одну текстуру для альbedo та згладжування;

- Built In Legacy Diffuse є застарілим вбудованим матеріалом ландшафту з попередніх релізів Unity (версії 4 і раніше). Він використовує модель освітлення Ламберта та має додаткову підтримку звичайної карти;

- Built In Legacy Specular використовує модель освітлення Blinn-Phong має додаткову підтримку звичайної картки. Цей параметр також дозволяє встановити загальний колір відображення та блиск для ландшафту.

Опція Custom використовує матеріал, що настроюється для вашого ландшафту. Цей матеріал має бути шейдером, орієнтованим для рендерингу ландшафту.

Наступний розділ налаштувань -Tree & Detail Objects. Перша опція – Draw. Тут можна налаштувати промальовування деталей. Detail Distance - це відстань від камери, після якої деталі відсікатимуться. Detail Density представляє кількість об'єктів деталей/трави у цій умовній одиниці площі. Tree Distance є відстанню від камери, за яким дерева будуть відсікатися. Так як у нашій грі не будуть дерева наступні параметри з розділу Tree & Detail Objects, нам не потрібно розбирати.

Першими трьома інструментами панелі інструментів Terrain inspector є інструменти зміни висоти. Ці інструменти використовують для малювання змін висоти на поверхні.

Вони надають набір "пензлів" з налаштуваннями Brush Size та Opacity; Ці інструменти дуже схожі на інструменти малювання будь-якого редактора зображень. Вони дозволяють малювати деталі ландшафту [10].

					КС КРБ 123.328.00.00 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Першим інструментом є Raise / Lower Height. При малюванні за допомогою цього інструменту висота ландшафту збільшуватиметься доки ви ведете мишкою із затиснутою лівою кнопкою по terrain. Якщо утримувати мишу в одній конкретній області, висота сумуватиметься.

Другий інструмент у редакторі ландшафту – Paint Height. Цей інструмент використовується для встановлення цільової висоти області ландшафту.

Він працює аналогічно до інструменту Raise/Lower, але має додаткову властивість для встановлення цільової висоти Height. Ви можете вибрати бажану висоту Height, змінивши значення Height. Кнопка Flatten поруч із властивістю height, яка просто встановлює висоту всього terrain в задане значення Height.

Третій інструмент – Smooth Height. Цей інструмент не різко збільшуватиме або зменшуватиме висоту terrain. Натомість він пом'якшить ландшафт і зменшить появу різких змін.

Unity дозволяє малювати ландшафти використовуючи текстури . Текстури можуть застосовуватись із різною прозорістю, що означає, що можна робити плавні переходи між двома текстурами. Інструмент Paint Texture має кілька пензлів і дозволяє вам змінювати різні атрибути [11].

					<i>КС КРБ 123.328.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		21

РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА

2.1 Особливості представлення моделі дрона

Квадрокоптер містить по дві пари обертових двигунів та гвинтів, котрі розташовані у чотирьох вершинах квадратного каркасу. Пристрій може здійснювати вертикальний зліт та посадку, як і звичайний вертоліт. Проте системи керування у вертольоту та квадрокоптера мають значні відмінності за рахунок динаміки польоту.

Дрон має два двигуни, котрі здійснюють обертання у вершинах квадратної рами: два - за годинниковою стрілкою, та два двигуни – проти неї (рис. 2.1).

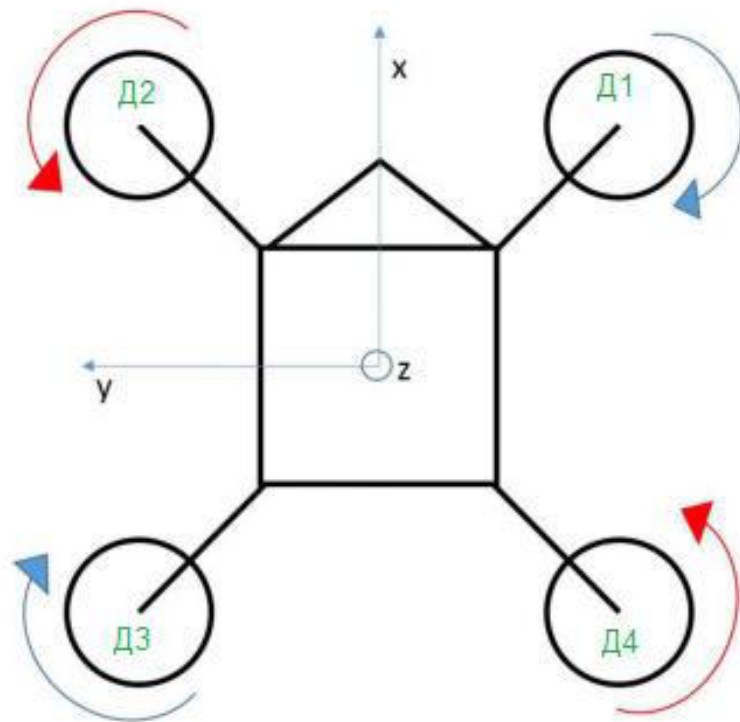


Рисунок 2.1 – Схема обертання двигунів квадрокоптера
(Д1 –двигун 1, Д2 – двигун 2, Д3 – двигун 3, Д4- двигун 4)

					КС КРБ 123.328.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Шелофастов Б.			Літ.	Арк.	Аркушів
Керівник.		Стадник Н.Б.					
Реценз.					ТНТУ, каф. КС, гр. СІс-42		
Н. Контр.							
Затверд.		Осухівська Г.М					

Просторовий рух забезпечується зміною кінцевої сили та крутного моменту на кожній з осей. Для забезпечення правильної зміни самого крутного моменту і сили у схемі, двигуни повинні працювати у парах (табл. 2.1).

Таблиця 2.1 – Попарна робота двигунів

	Газ				Крен				Тангаж				Рискання			
Двигуни	1	2	3	4	1	2	3	3	1	2	3	4	1	2	3	4

Поступальний рух дрона потребує нахилу платформи до необхідної осі. Для задання опорного кута пристрою використовуються крен ϕ , тангаж θ , рискання ψ . Умови переміщення у 3D -просторі наведено на рис. 2.2.

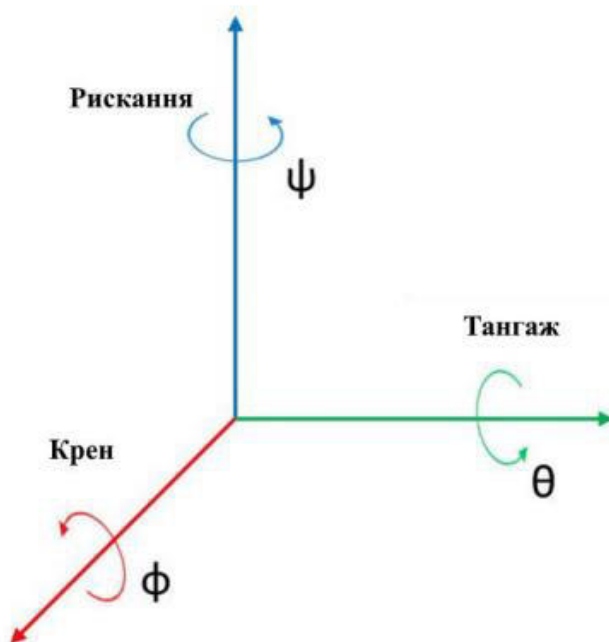


Рисунок 2.2– Необхідні позначення для руху в тривимірному просторі

Виходячи із розташування двигунів з табл. 2.1, проста зміна швидкості руху однієї з пар двигунів спричинює рух у шести ступенях вільності [12].

Модель квадрокоптера буде розроблена із використанням рівнянь Ньютона – Ейлера. Виходячи з цього, дрон буде описуватися як жорстке тіло, котре переміщується через різноманітні системи відліку на базі системи

координат, яка є нерухомою. Це дасть змогу нам описати просторове положення орієнтації польоту конструкції.

Квадрокоптер моделюється сферичним щільним тілом (з масою M і радіусом R) по центру системи. Двигуни представляються як чотири точки мас з масою m , котрі розташовані на відстані l від центру (рис. 2.3).

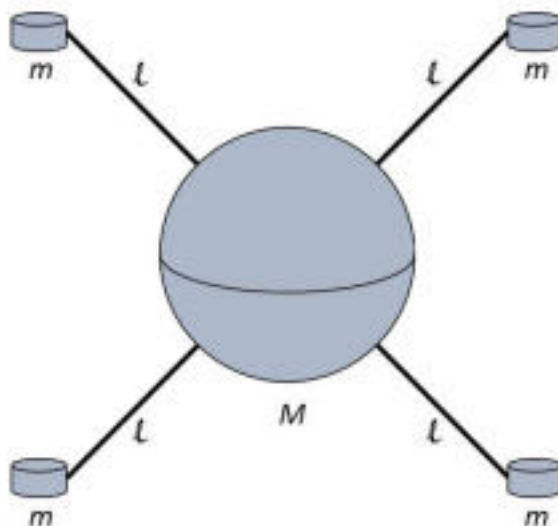


Рисунок 2.3– Модель квадрокоптера

Відповідну інерцію всієї системи можна обчислити за формулою (2.1):

$$J_x = J_y = \frac{2MR^2}{5} + 2ml^2 \quad (2.1)$$
$$J_z = \frac{2MR^2}{5} + 4ml^2$$

2.2 Структура гри

Структура гри буде виглядати наступним чином: у нас є дрон і його завдання набрати якнайбільше очок, подолавши перешкод і ворожих дронів. Саме керування дроном відбуватиметься через клавіатуру, а надалі перевести керування на джойстик, так ми зможемо більше занурити користувача в ігрову

					КС КРБ 123.328.00.00 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

атмосферу і надати реалізму. У нашого дрона буде вбудовані гармати для знищення ворожих дронів, сама суть навчити користуватися дроном.

Побудуємо схему гри (рис. 2.4), вона буде виглядати так: у нас є екран завантаження, який буде відповідати за старт гри, для цього створюється окрема сцена під назвою “SplashScreen”, після завантажувального екрана запускається наступна сцена, яка запускає саму гру або перший рівень ігри. Якщо під час ігрового процесу ми вмираємо, то наш дрон повертається до стартової точки.

Також є екран перемоги у нашому випадку, наш дрон просто припиняє літати, а за допомогою UI вікна ми відобразимо кількість очок, набраних на даному рівні гри. Після цього ми переходимо на новий рівень гри.

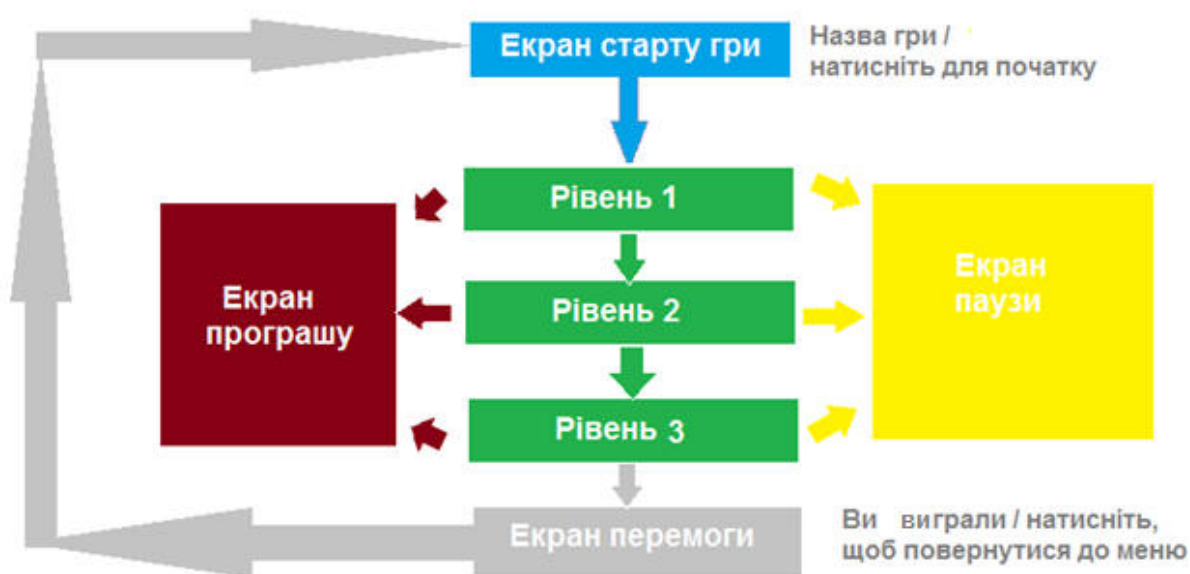


Рисунок 2.4 – Схема гри

Користувач симулятора освоює керування і здійснює поставлені дії за час, який відводиться на це. Щойно усі задачі будуть успішно виконані, буде запропоновано здійснити перехід на черговий рівень. Якщо користувач втратить дрон чи управління ним, рівень можна почати заново. Також якщо всі поставлені завдання не виконані, тоді буде також запропоновано повторити спробу. Порядок дій процесу навчання користувача наведено на рис. 2.5.



Рисунок 2.5 – Порядок дій при навчанні

З метою демонстрації варіантів сценаріїв використання користувачем програмного продукту на рис. 2.6 наведена ULM діаграма.

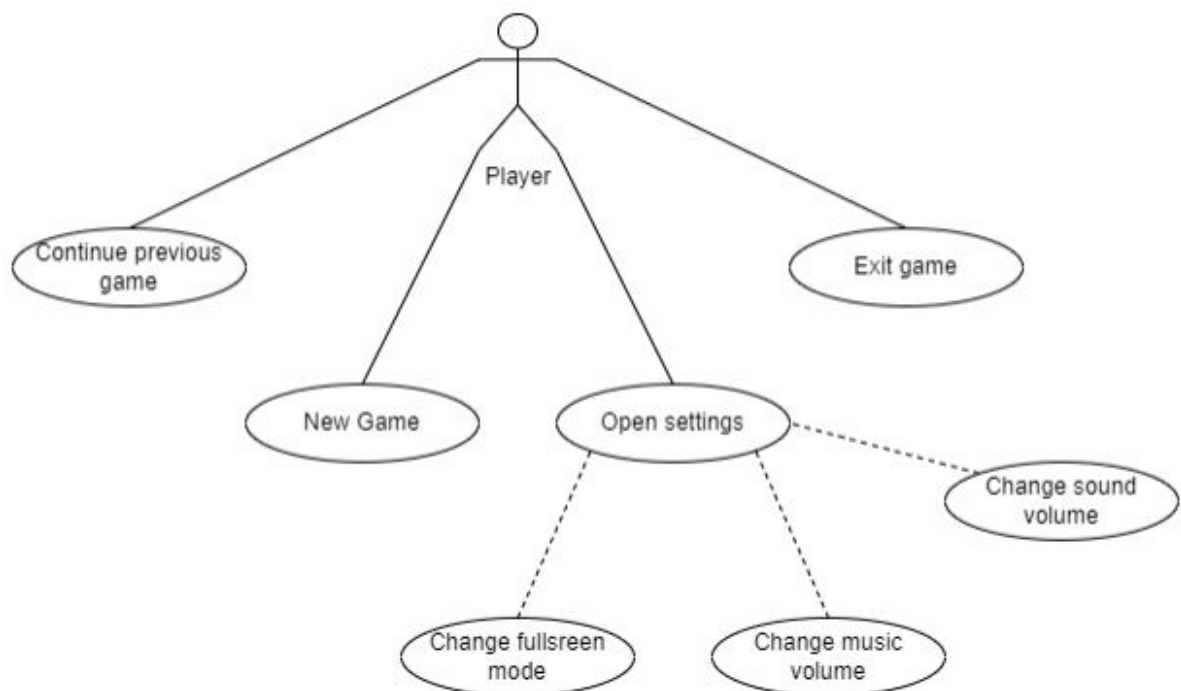


Рисунок 2.6 – Діаграма варіантів використання

2.3 SkyBox у Unity

Для того, щоб передати атмосферу нашої гри скористаємося інструментом Skybox, який є панорамною текстурою, котра міститься за усіма складовими елементами в сцені, з метою моделювання космосу, або міста, або будь-якої перспективи. Skybox - це панорамний вид, розділений на шість текстур, що представляють шість видимих напрямків уздовж осей [13].

Canvas для Unity UI - це область, всередині якої розміщуються всі елементи інтерфейсу користувача. Тобто коли ми створюємо новий ігровий об'єкт, то Canvas для нього створюється автоматично. Починаючи з версії 4.6, для роботи з елементами інтерфейсу користувача використовується нова версія Unity UI. Як і під час роботи із звичайними об'єктами, елементи HUD організовують в ієрархії, кореневим елементом яких є об'єкт із компонентом Canvas. При цьому порядок в ієрархії визначає порядок рендеру: об'єкти, які знаходяться в самому низу, малюються останніми і, відповідно, розташовуються поверх інших об'єктів. Не має значення, який тип програми розробляється - як 3d, так і 2d гри використовують одну і ту ж логіку роботи з UI. Для зручності проектування кожен елемент UI відображається у вигляді прямокутника. Для маніпуляцій із нею використовується два основних компонента, Rect Tool і Rect Transform (рис. 2.7). Останній — це новий компонент, у якого, окрім уже вказаних полів, є особливе поле Anchors (Якоря) [13].

Якорі визначають прив'язку елементів до розмірів батьківського елемента. У компонента є 4 якорі, кожен із яких відповідає за одну з вершин елемента. Позиція та розміри елемента обчислюються на основі відстані між вершиною та якорем та позицією самого якоря. Позиція якоря, своєю чергою, визначається у відсотковому відношенні від розмірів батьківського елемента. Таким чином, якщо всі чотири елементи знаходяться в тому самому місці, розмір елемента буде постійним.

Якщо в одній точці знаходяться якорі однієї площини (верхній і нижній або лівий і правий), елемент не буде розтягуватися щодо цієї осі. Якщо ж якорі

					КС КРБ 123.328.00.00 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

не в одній точці, то позиція кожного якоря буде розрахована у відсотках і до отриманої величини буде додано відстань до вершини (вона не змінюється) [14].

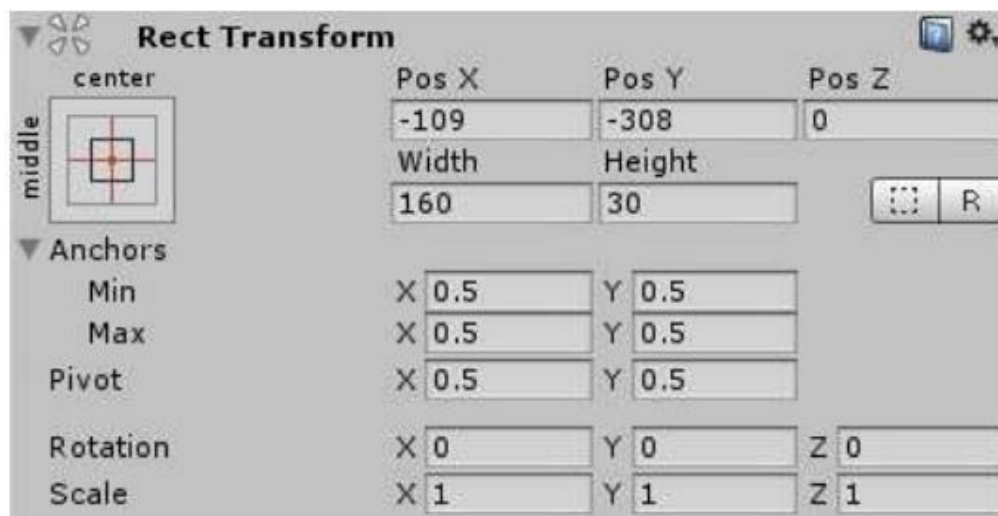


Рисунок 2.7 – Властивості Rect Transform

Варіанти відображення Canvas Unity UI Canvas відповідає параметру Render Mode, який має три режими:

- Screen Space - Overlay - Canvas рендериться поверх всіх елементів сцени. Якщо провести аналогію, то Canvas з елементами - це як вікно з наклейками в поїзді. Світ за вікном може змінюватися (рух камери в Unity), але наклейки залишаються незмінними. Цей режим змінює розмір Canvas, коли змінюється розмір екрана. Він ідеально підходить для статичної інформації на кшталт рахунку або інструментів керування;

- Screen Space - Camera - це те ж вікно, але замість "наклейок" на ньому розташована об'ємна аплікація (не має значення, це 2d або 3d гра). Також розміщується поверх решти елементів сцени, але при цьому на елементи Canvas поширюються закони перспективи (за умови, що тип проєкції камери – Perspective). Цей режим ідеально підходить для меню або інших елементів, які мають виглядати об'ємними;

- World Space - режим, в якому Canvas виступає елементом сцени і може бути дочірнім по відношенню до інших елементів Game Object. Цей режим ідеально підходить для хмаринок з текстом персонажів, індикаторів патронів у

зброї тощо [14].

2.4 Створення ландшафту гри

Перше, що потрібно створити – це ландшафт гри і розробити дизайн, для цього скористаємося спеціальним об'єктом для створення ландшафту. За допомогою інструменту Terrain ми створимо наш ландшафт. Сам Terrain надає нам інструменти, необхідні для редагування та оформлення ландшафту [15].

Перший крок – це зробити новий проєкт Unity. Тепер додамо нову сцену (File – New Scene). Нова сцена була створена з двома ігровими об'єктами, Main Camera та Directional Light. Тепер, щоб додати ігровий об'єкт terrain, перейдемо в меню GameObject і оберемо 3D Object – Terrain. З допомогою цього об'єкта графічний рушій додасть плоску поверхню (звану Terrain) в нашу сцену (рис. 2.8). Ця поверхня є сіткою, котру ми маємо намір зробити для створення ландшафту [11].

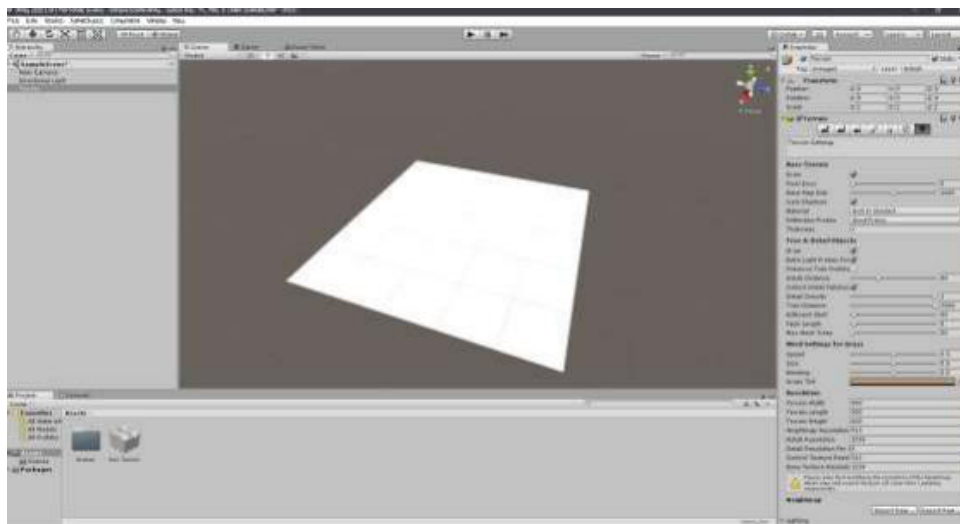


Рисунок 2.8 – Terrain

Задамо розмір terrain використовуючи в Terrain settings в розділі Resolution, поміняємо значення Terrain Width і Terrain Length до 1000, а Terrain Height до 600, розмір трійну змінюється для того, щоб збільшити масштаб нашої локації. Після цього потрібно вирівняти наш об'єкт у центрі нашої сцени. Далі ми

					КС КРБ 123.328.00.00 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

використовуємо інструменти для створення локації:

- Raise / Lower Height;
- Paint Height;
- Smooth Height.

Ці компоненти допоможуть нам задати потрібну нам форму скелі та рельєф локації. Для реалізації ландшафту в першу чергу скористаємося інструментом Paint Height, цей інструмент дозволить нам створити додатковий шар на наш terrain, це дозволить нам у подальшому допоможе створювати ущелини та ями. Для того щоб використовувати додатковий шар, зайдемо в параметр paint height і виберемо пункт height задаємо значення 100 і натиснемо на кнопку Flatten. Як бачимо на рис. 2.9 наш terrain піднявся на сто одиниць, що дозволить нам надалі створювати ущелини за допомогою інструменту Raise / Lower Height, так само за допомогою цього інструменту ми зможемо створити скелі, рівнини чи пагорби. У розділі Raise / Lower Height є такий параметр як Brush size, що дозволяє змінювати розмір нашого пензля, що дозволить нам намалювати різних розмірів гір і рівнин, а з допомогою Opacity контролювати висоту [15].

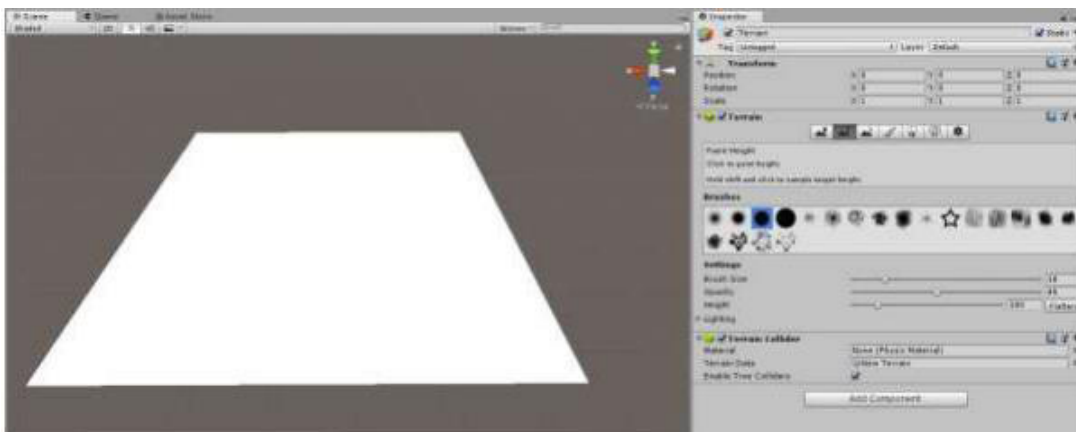


Рисунок 2.9 – Вплив Paint Height на Terrain

Після всіх маніпуляцій з інструментами дозволяє нам змінювати зовнішній вигляд нашого terrain' та врешті-решт ми отримуємо наш фінальний різновид ландшафту (рис. 2.10).

					КС КРБ 123.328.00.00 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

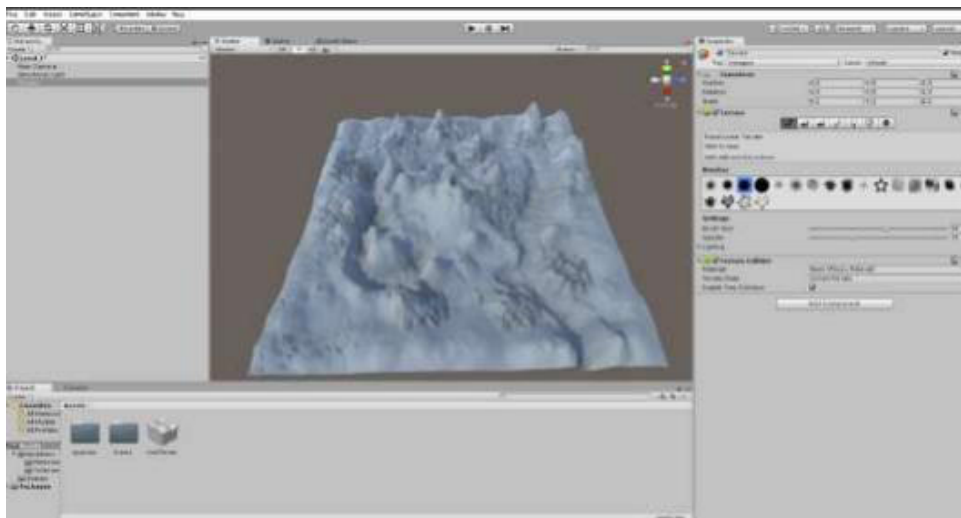


Рисунок 2.10 – Фінальний вид ландшафту

Після того як ми створили необхідний ландшафт потрібно зайнятися забарвленням terrain'a . Для цього скористаємося assets store (рис. 2.11), де можна буде знайти потрібний нам матеріал для нашого ландшафту [15].

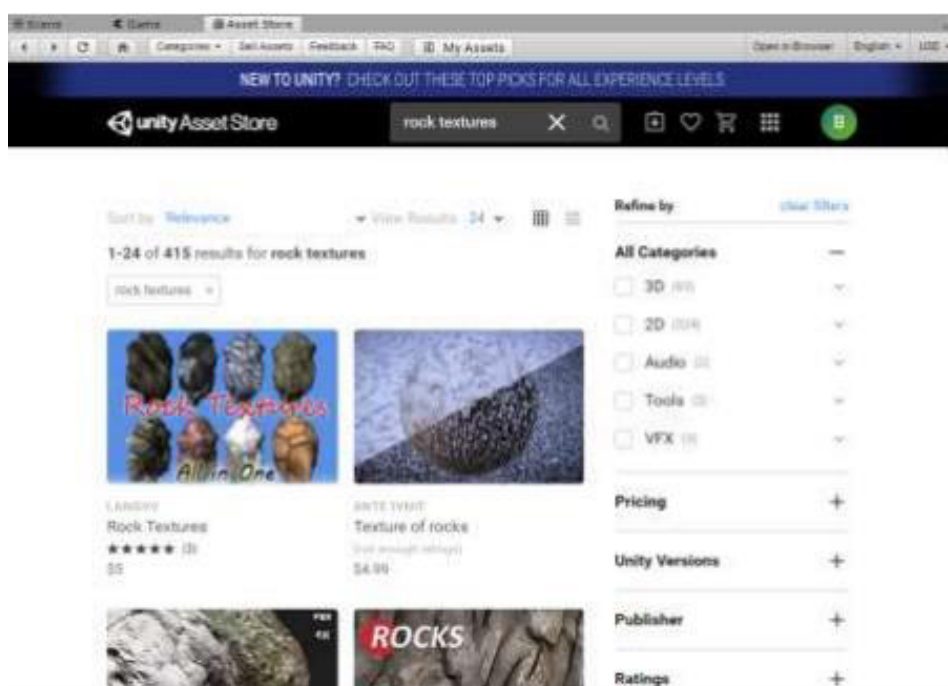


Рисунок 2.11 – Assets - store текстур

Вибравши відповідну нам текстуру для нашого ландшафту, додамо їх до нашого проекту. Далі ми повинні за допомогою інструменту Paint texture пофарбувати ландшафт (рис. 2.12), для цього в пункті Texture додамо текстуру, що підходить для нас, і пофарбуємо наш ландшафт [10].

					КС КРБ 123.328.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

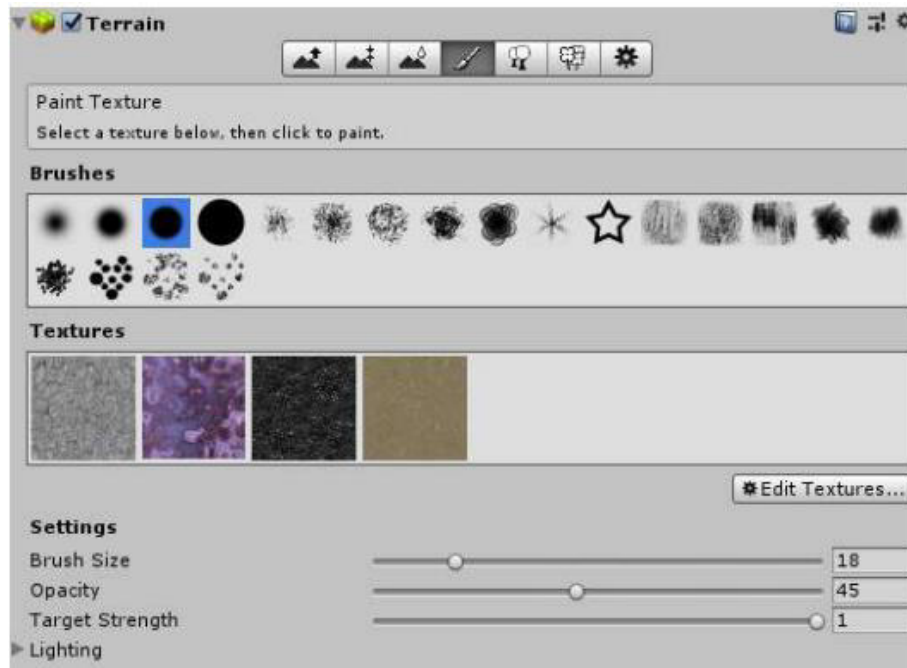


Рисунок 2.12 – Налаштування текстур

2.5 Налаштування Skybox

Фон - найважливіша частина гри, за допомогою фону можна передати атмосферу гри, а також показати гравцю, на якій локації він знаходиться. Для створення фону потрібно створити матеріал, який зберігатиме дані про фон. Підібравши потрібний для нас фон, додаємо його у раніше створений нами матеріал, і прив'язуємо наш Skybox до сцени.

Skybox - панорамний вид, розділений на шість текстур, що представляють шість видимих напрямків вздовж основних осей (вгору, вниз, ліворуч, праворуч, назад і вперед). Якщо Skybox створено правильно, текстурні зображення будуть як єдине ціле з обох боків, щоб дати безперервний навколишній образ, який можна розглядати «зсередини» у будь-якому напрямку (рис. 2.13). Панорама знаходиться позаду всіх інших об'єктів у сцені і обертається відповідно до поточної орієнтації камери (не плутайте з положенням камери, котре завжди береться таким, щоб бути в центрі панорами). Skybox таким чином, простий спосіб додати реалізму до сцени з мінімальним навантаженням на відеокарту.

					<i>КС КРБ 123.328.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

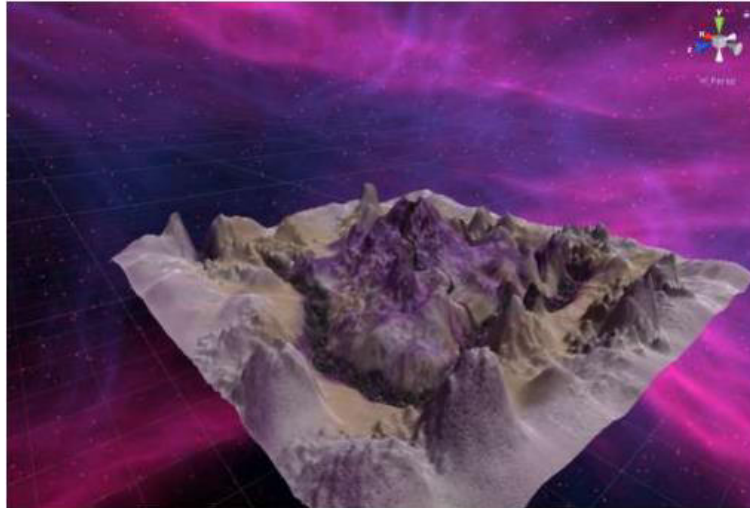


Рисунок 2.13 – Skybox

Unity постачається з набором високоякісних Skybox у наборі Standard Assets, але можна також підібрати панорамні малюнки з інтернету чи спробувати створити його особисто, застосовуючи ПЗ для 3D -моделювання. Після цього потрібно перейти до нашого skybox, потім до панелі Inspector. І визначаємо його атрибути, для Wrap Mode встановимо значення Clamp (якщо ви встановите значення Repeat, краї зображень не утворюватимуть «безшовну» текстуру) [11].

Сам Skybox – це спеціальний матеріал, який застосовує один із шейдерів у підменю RenderFX. Якщо вибрати Skybox Shader, ми побачимо в панелі інспектора щось на зразок наступного, з шістьма прикладами текстур.

Skybox Cubed шейдер працює в загальному аналогічно, проте потребує текстури, котру додадуться до cubemap asset [11].

					КС КРБ 123.328.00.00 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Створення дрону

Перейдемо до створення самого дрону, він буде сформований з різних об'єктів, які будуть з'єднані разом, сам дрон нагадуватиме космічний корабель. Для створення дрону завантажуюємо в assets store відповідні деталі для його створення. Як видно на рис. 3.1, ми збираємо корабель за допомогою деталей, котрі завантажені з assets store. Підбираємо відповідні деталі та збираємо космічний дрон [14].

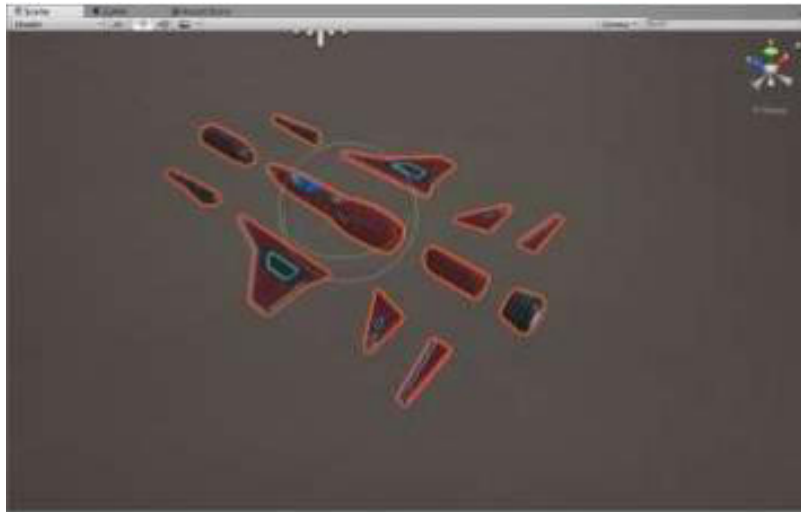


Рисунок 3.1 – Складання корабля

На рис. 3.1 показано фінальний вид дрону. Надалі до дрону потрібно під'єднати анімацію горіння двигуна та пострілів з лазерних гармат, сам космічний дрон був пофарбований у жовтий колір, а основні деталі корабля були пофарбовані блакитним та темно-синім кольорами.

Змн.	Арк.	№ докум.	Підпис	Дата	КС КРБ 123.328.00.00 ПЗ		
Розроб.		Шелофастов Б.			Літ.	Арк.	Аркушів
Керівник.		Стадник Н.Б.					
Реценз.					ТНТУ, каф. КС, гр. СІс-42		
Н. Контр.							
Затверд.		Осухівська Г.М					



Рисунок 3.2 – Зібраний корабель

3.2 Реалізація Splash Screen

Приступимо до створення початкового екрану та музики у грі, при цьому музика має зациклюватися. Splash Screen потрібен для того, щоб показати будь-яку інформацію про гру. Для реалізації splash screen потрібно створити нову сцену і до цієї сцени додаємо музику. Напишемо скрип SplashScreen, його завдання - це запускати початковий екран і перемикати на нову сцену (рис. 3.3).



Рисунок 3.3 – Налаштування Audio source

3.3 Рух камери і Standard assets

Ассет Unity – це елемент, який ви можете використовувати у своїй грі чи проєкті. Ассет може відбуватися з файлу, створеного поза Unity, наприклад з 3D-моделі, аудіо-файлу, файлу зображення або з іншого типу, підтримуваного Unity. Деякі типи ассетів можуть створюватися в рушію Unity, наприклад контролер аніматорів, аудіомікшер або текстура рендерингу [6].

Використовуємо інструмент way paint, він відповідає за рух камери. Важлива частина геймплею нашої гри полягає в тому, щоб камера рухалася заданою траєкторією і для її реалізації скористаємося way point, де ми задаватимемо траєкторію руху нашої камери. Для цього ми скористаємось стандартними ассетами в Unity. Заходимо на asset store та знаходимо Standard Assets пакети (рис. 3.4) [6].

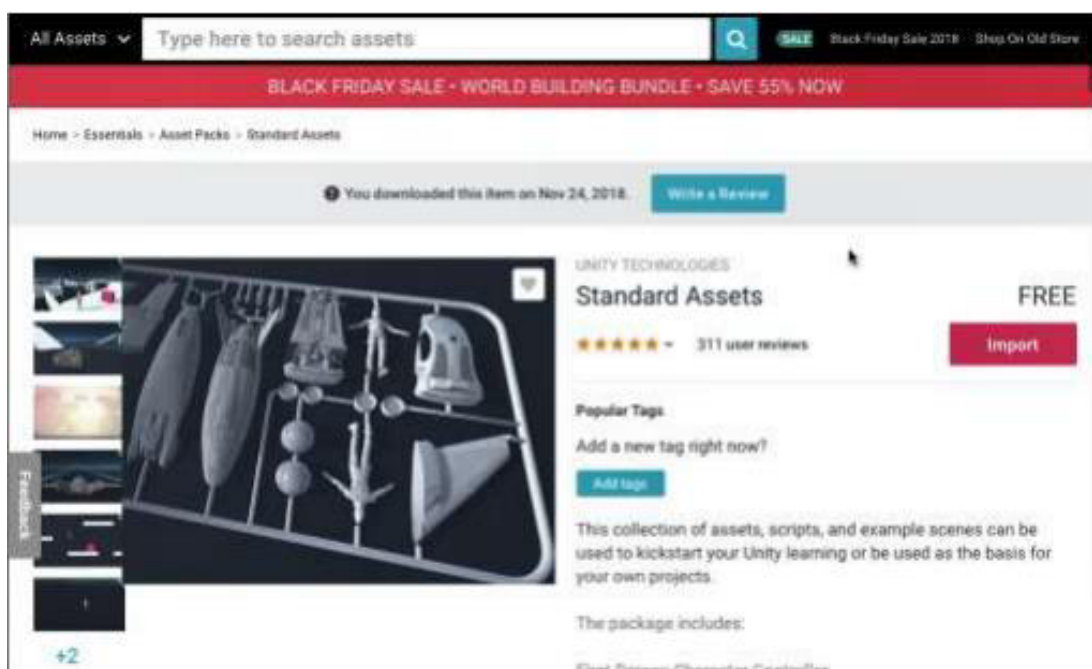


Рисунок 3.4 – Standard assets

Далі створюємо порожній проєкт і назвемо його way, потім створюємо дочірній об'єкт Point. Створимо за допомогою дочірніх об'єктів point шлях, при цьому шлях має бути замкнутим, як видно на рис. 3.5.

					КС КРБ 123.328.00.00 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.5 – Маршрут камери

Після того, як ми проклали шлях за допомогою об'єкта point, потрібно прив'язати весь наш шлях до скрипта findPath, а скрипт WaypointProgressTracker підлаштовуємо до main Camera, вона у свою чергу рухатиме нашу камеру заданим шляхом. Управління дроном відбуватиметься за допомогою скрипта CrossPlatformInput, цей скрипт дозволяє керувати дроном не лише за допомогою клавіатури. Обмежимо пересування нашого дрона, адже ми маємо такий об'єкт main camera, і дрон повинен рухатися щодо нашої камери і не виходити за її межі. Для цього введемо у скрипті PlayerControl обмеження та налаштуємо плавне керування.

Алгоритм роботи методу findPath() показано на рис. 3.6.

					КС КРБ 123.328.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

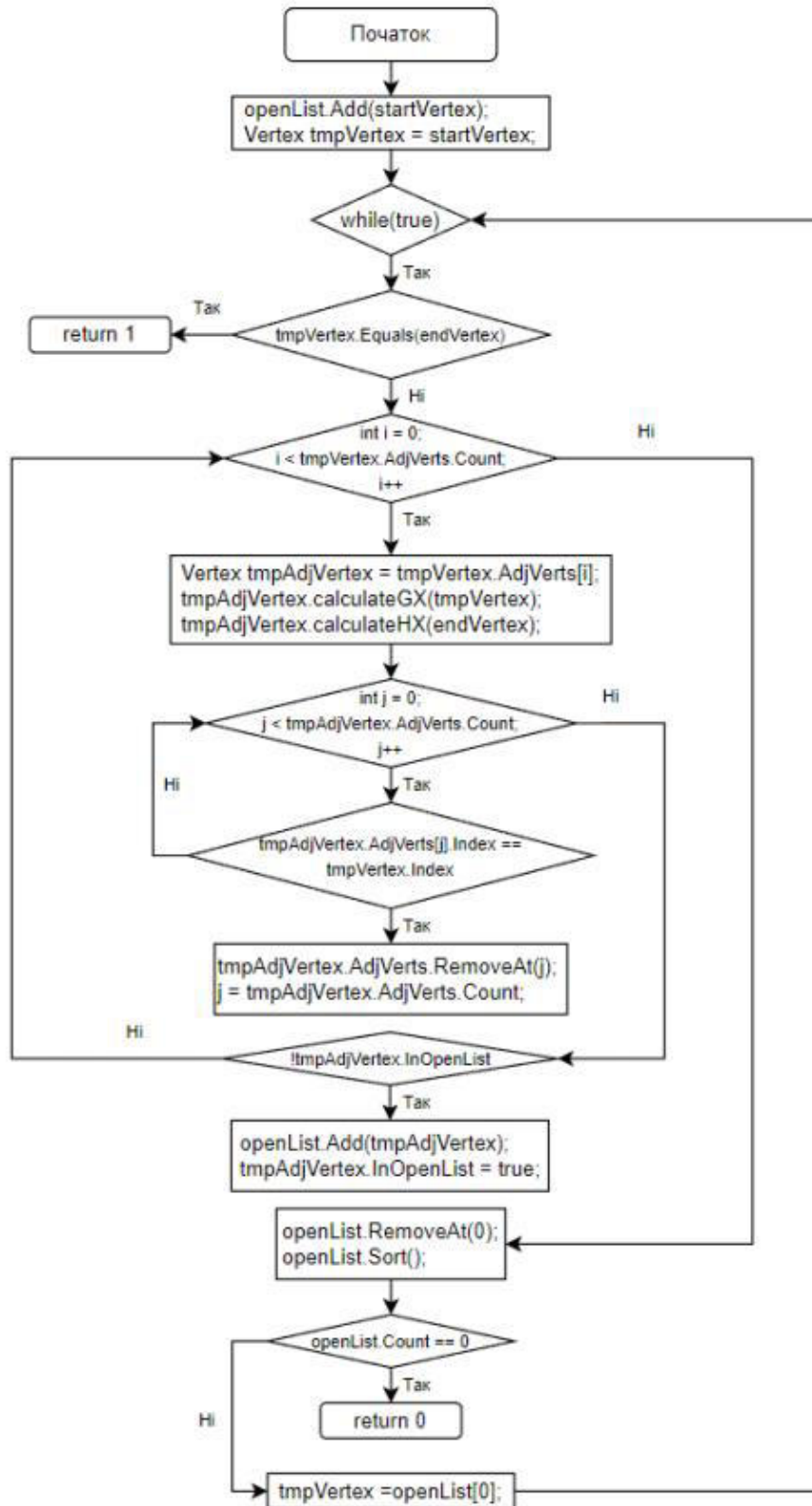


Рисунок 3.6 – Блок - схема методу findPath()

Відповідальною за обертання нашого корабля пропишемо ще одну функцію RotateShip(), щоб політ дрона був плавним, основна проблема гри – це грамотне налаштування управління і для цього були враховані такі проблеми як

рух по осях та обертання [14]. Програмний код цієї функції наведено на рис. 3.7.

```
void RotateShip()  
{  
    float xRot = transform.localPosition.y * xRotFactor + yMove  
ove * xMoveRot;  
    float yRot = transform.localPosition.x * yRotFactor + xMove  
ove * yMoveRot;  
    float zRot = xMove * zMoveRot;  
  
    transform.localRotation = Quaternion.Euler(xRot, yRot, zRo  
t);  
}
```

Рисунок 3.7 – Код функції RotateShip()

3.4 Реалізація ефектів

Ефекти - це невід'ємна частина ігор і різних симуляцій, зробимо для нашої гри ефекти пострілу, додамо ефект Particle system і, як видно на рис. 3.8, текстура наших частинок не схожа на постріли з лазерної гармати, сам ефект відображається в локальних координатах. Для цього додамо нові ефекти і підберемо відповідні нам. Надамо нашому ефекту відповідний вигляд. У Particle System є властивість Shape виберемо пункт Cone, а в розділі Angle значення поставимо значення 0, Radius , що приблизно соромиться до нуля, таким чином у нас вийде ряд частинок [7].

У розділі Emission - надамо значення Rate over Time рівної трьом в Durattion 0.20 і підключимо looping. Поміняємо розмір лазера та його колір, після цього наш ефект Particle system присвоюємо до gameobject, щоб надати фізичного змісту нашій грі. Створимо ефект вибуху, щоб під час зіткнення чи пострілу з лазерної гармати гравець розумів, що власне сталося. У параметрі Shape задаємо значення Sphere, налаштуємо час випромінювання частинок вибуху та швидкість руху та радіус. Параметрі lifetime введемо два значення один і два, тим самим покажемо в анімації вибуху різні розміри частинок.

					КС КРБ 123.328.00.00 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.8 – Анімація двигуна

Для реалізму додамо відповідний для вибуху матеріал, а саме підберемо текстуру схожу на частинки, що горять, і налаштуємо градієнт. Підключимо до нашої анімації вибуху музику та пропишемо скрип, який запускатиме цю анімацію. Придумаємо перешкоди для нашої гри, розробимо схему перешкод та змінимо ландшафт. Додамо нові game object і розподілимо їх по всій локації, також потрібно враховувати, що перешкоди повинні взаємодіяти з іншими об'єктами такими як дрон та його постріли з гармати (рис. 3.9) [14].



Рисунок 3.9 – Анімація кораблів

Як бачимо на рис. 3.10, що у цій сцені як перешкод взято кілька дронів. Дрони були створені за допомогою конструктора в Unity та assets store.

					КС КРБ 123.328.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40



Рисунок 3.10 – Дрони як перешкода

Скористайтеся компонентами колайдера для взаємодії з довкіллям, компоненти колайдера визначають форму об'єкта для фізичних зіткнень. Колайдер не обов'язково повинен мати ту ж форму, що й сітка об'єкта, і фактично грубе наближення часто більш ефективно і невиразне в ігровому процесі. Для створення підлоги, стін та інших нерухомих елементів сцени до об'єктів без компонента Rigidbody можна додавати колайдери. Вони називаються static (статичні) колайдерами. Зазвичай не варто переміщати статичні колайдери, змінюючи положення їх трансформації, так як це матиме значний вплив на продуктивність фізичного двигуна (рис. 3.11).



Рисунок 3.11 – Ворожі кораблі

					КС КРБ 123.328.00.00 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

У разі зіткнення система скриптингу може це виявити та виконати дії, зазначені у функції `OnCollisionEnter`. Однак ми також можемо використовувати фізичний двигун просто для виявлення того, що один колайдер входить у простір іншого без створення колізії. Колайдер, налаштований як тригер (за допомогою властивості `Is Trigger`), не веде себе як твердий об'єкт і просто пропускатиме інші колайдери крізь себе. Коли інший колайдер увійде на територію колайдера, тригер викличе функцію `OnTriggerEnter` у скриптах об'єкта, до якого приєднаний тригер [11].

У нашому випадку, коли дрон стикається з іншими об'єктами або знищує їх, то найкраще використовувати функцію `OnTriggerEnter`. Скрипт `CollisionHandler`, відповідає за все що відбувається з нашим дроном, тобто якщо він вибухне, то ми почнемо гру заново.

3.5 Реалізація вікна UI

`Canvas` для `Unity UI` - це область, всередині якої розміщуються всі елементи інтерфейсу користувача. Тобто, коли ми створюємо новий ігровий об'єкт, то `Canvas` для нього створюється автоматично. Починаючи з версії 4.6, для роботи з елементами інтерфейсу користувача використовується нова версія `Unity UI`. Як і під час роботи із звичайними об'єктами, елементи `HUD` організовують в ієрархії, кореневим елементом яких є об'єкт із компонентом `Canvas`. При цьому порядок в ієрархії визначає порядок рендеру: об'єкти, що знаходяться в самому низу, малюються останніми і, відповідно, розташовуються поверх інших об'єктів. Не має значення, який тип програми ви розробляєте - як 3d, так і 2d ігри використовують одну і ту ж логіку роботи з `UI`. Те саме стосується платформи. Не має значення, це `Web` з `HTML5` або гра для `Android` - достатньо буде описати одне полотно `Canvas` [6].

Створимо `UI` вікно і налаштуємо його так, щоб воно відображало окуляри для нашого дрона, а для приємного візуального ефекту підберемо шрифт, що підходить нам (рис. 3.12).

					КС КРБ 123.328.00.00 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

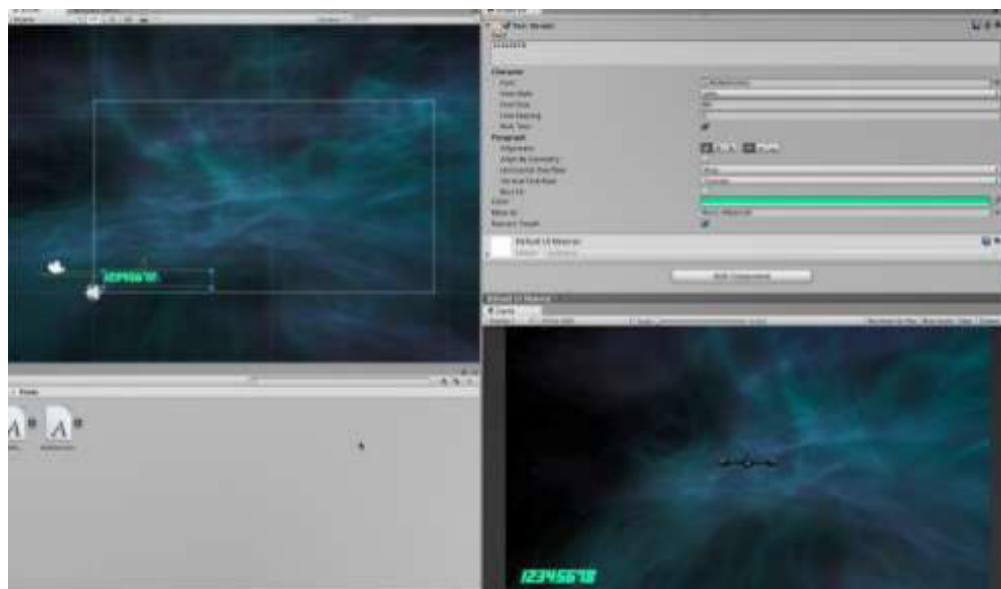


Рисунок 3.12 – UI вікно

Тепер наше завдання записувати окуляри у вікно UI-text, для цього потрібно написати скрипт, завдання скрипту підраховувати всі окуляри, які ми будемо отримувати під час ігрового процесу та виводити їх на екрані. У нас є скрипт Score.cs який і буде відповідати за це, перш за все потрібно створити змінну, яка буде зберігати кількість очок у нашому випадку - це буде score значення якої дорівнюватиме нулю, так як нам потрібно змінювати параметр text в полі UI то для роботи з цим параметром підключимо у скрипті Score.cs бібліотеку UnityEngine.UI. Тепер ми можемо змінювати значення в параметрі text, напишемо метод void Start(), де буде підраховуватись окуляри і виводити їх на екран (рис. 3.13).

```

// setup script properties
private void Start()
{
    if (target == null)
    {
        target = new GameObject(name + " Waypoint Target
").transform;
    }

    Reset();
}

```

Рисунок 3.13 – Код методу Start()

За допомогою скрипту Enemy ми передаємо кількість знищених об'єктів, а в методі void Death() все описується (рис. 3.14).

```

void Death()
{
    GameObject fx = Instantiate(deathFX, transform.position, Q
uaternion.identity);
    fx.transform.parent = parent;
    Destroy(gameObject);
}
}

```

Рисунок 3.14 – Код методу Death()

3.6 Створення анімації за допомогою інструмента TimeLine

Розробники часто використовують катсцени. У деяких іграх створюють спеціально відрендеровані анімовані сцени з моделями високої деталізації, в інших застосовують справжні внутрішньоігрові моделі. Завдяки використанню всередині ігрових моделей можна заощадити багато засобів і часу, тому що нам не доведеться створювати нові моделі, клуні та анімації виключно для катсцен.

					КС КРБ 123.328.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

Але навіть якщо ви використовуєте вже готові моделі, це не означає, що потрібно жертвувати спецефектами або драматизмом! Unity має потужний інструмент, що дозволяє створювати захоплюючі катсцени - Unity Timeline.

Timeline - це GameObject з компонентом timeline, який можна редагувати у вікні Unity Timeline, що управляє ключовими кадрами анімації та циклами життя об'єктів. При створенні катсцени за допомогою Unity Timeline, ми задаємо ключі анімації та визначаємо, коли вони мають спрацювати. Для створення Timeline потрібно зайти на вкладку Window – Sequencing-Timeline (рис. 3.15).

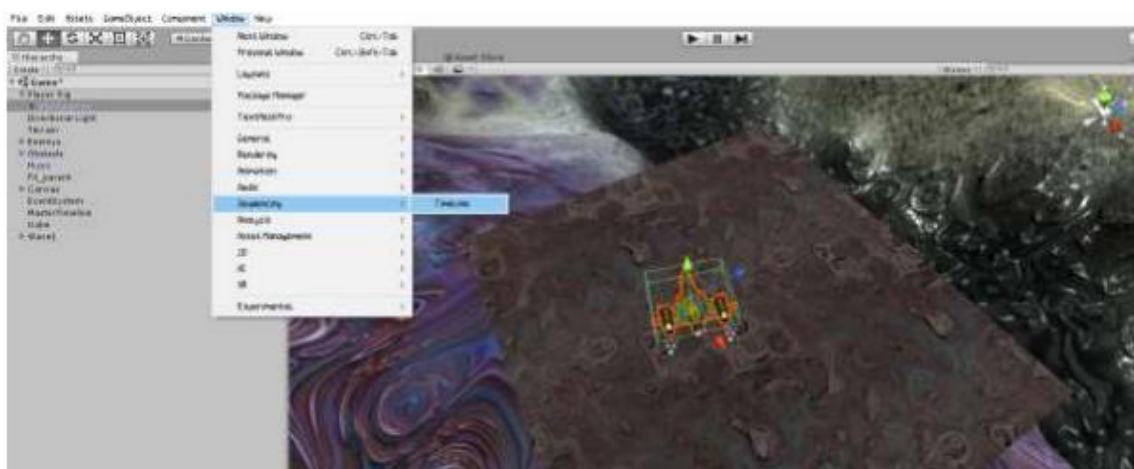


Рисунок 3.15 – Створення анімації

Для того щоб почати працювати з Timeline потрібно вибрати об'єкт, який ми анімуватимемо, у нашому випадку це ворожий корабель, після цього в полі Timeline натиснути на кнопку create, щоб створити файл анімації (рис. 3.16).

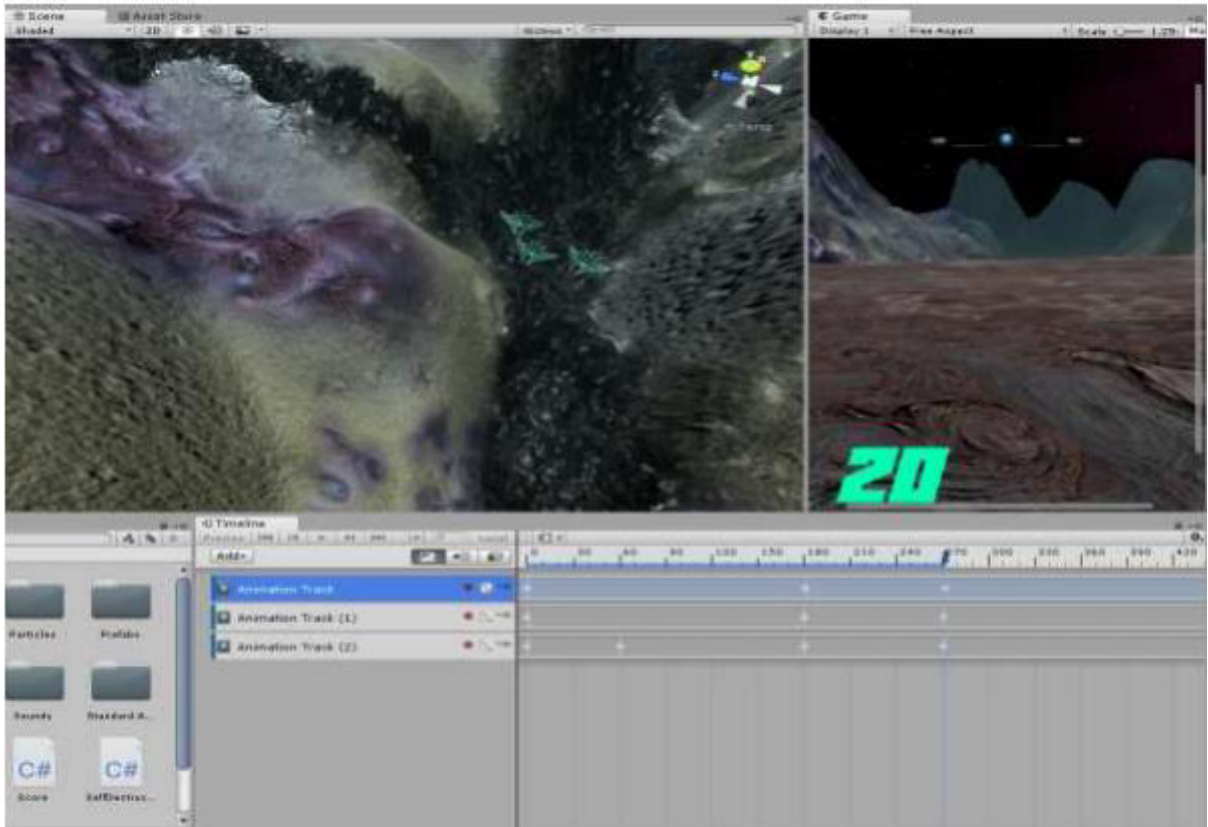


Рисунок 3.16 – Анімація дрону

Задамо стартову позицію корабля і після цього запишемо анімацію корабля за ключовими кадрами, після чого отримуємо запис анімації для даного об'єкта. Виконуємо всю роботу і з іншими кораблями.

Наприкінці ми повинні отримати повноцінну гру, де буде присутня ігрова зона, ворожі об'єкти, меню гри.

Після всієї виконаної роботи, потрібно зібрати саму гру, так як Unity є кросплатформовим графічним рушієм, то збірка (монтування) гри буде виконуватись під ОС Windows [14].

3.7 Монтування гри

Для того, щоб змонтувати наш проект у повноцінний продукт, потрібно зайти у параметр File - Build Settings (рис. 3.17). Потім необхідно розподілити за порядком всі наші сцени та вибрати у вікні Platform пункт PC, Mac&Linux та натискаємо кнопку Build.

					<i>КС КРБ 123.328.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

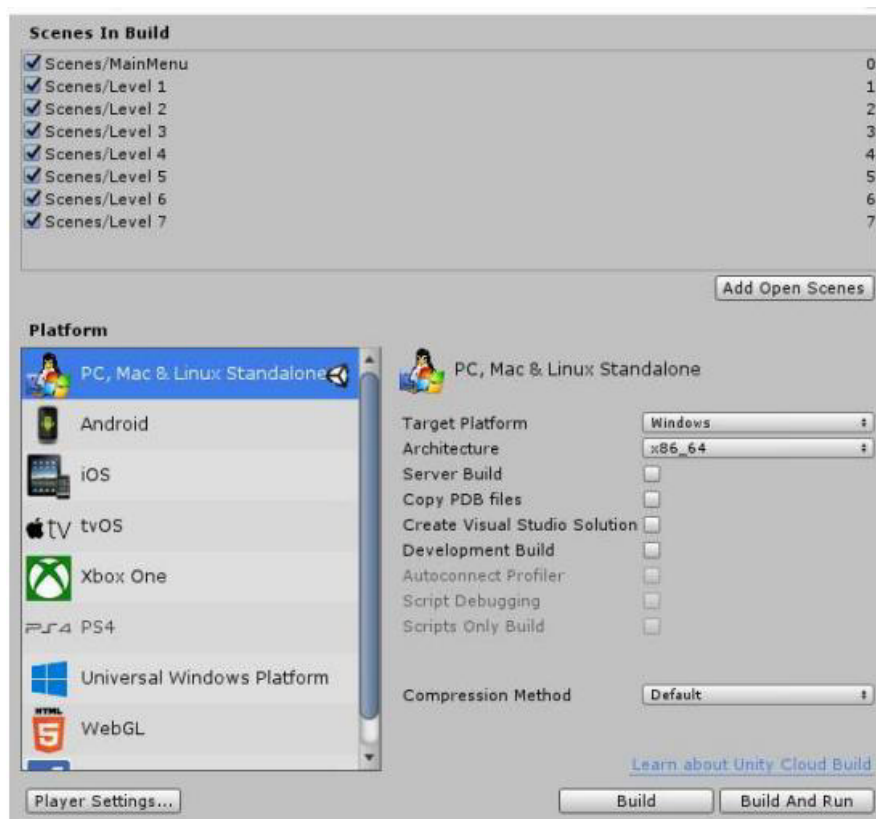


Рисунок 3.17 – Монтування проекту

Результатом виконання наведених вище операцій буде готовий проект, який можна успішно використовувати для наглядного навчання операторів дронів.

					КС КРБ 123.328.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Стихійні лиха та їх класифікація

Стихійні дії сил природи, поки що не повною мірою підвладні людині та щорічно завдають державі і населенню величезних збитків. Стихійні лиха - це такі явища природи, що викликають екстремальні ситуації, порушують нормальну життєдіяльність населення, роботу безлічі об'єктів. Стихійні лиха є трагедією для будь-якої держави. Через стихійні лиха страждає економіка країни, бо при цьому руйнуються виробничі підприємства, знищуються матеріальні цінності, гинуть люди.

Стихійні лиха – небезпечні природні явища, як правило раптового походження, хоча іноді і прогнозовані за допомогою метеорології, але на інтенсивність яких люди впливати не можуть. Їх можна класифікувати: за швидкістю переміщення - землетруси, зсуви, цунамі, снігопади, ожеледі - швидкі; підвищення рівня води в ріках через інтенсивні опади або танення снігу, льоду (повіні), звільнення внутрішньої енергії Землі, виверження вулканів - повільні. Часто виникають потужні, високошвидкісні потоки повітря через швидкий перепад значень атмосферного тиску (урагани, смерчі і т.п.). Стихійні лиха речовинного характеру можуть ініціювати виникнення різноманітних полів, які негативно впливають на здоров'я, самопочуття людини [18].

Стихійні явища часто виникають в комплексі, що значно посилює їх негативний вплив. Небезпечні природні явища визначаються трьома основними групами процесів - ендогенні, екзогенні та гідрометеорологічні.

Стихійні лиха, які характерні для України, за структурою можна поділити на прості, що включають один елемент - наприклад, сильний вітер, зсув або землетрус та складні. Вони складаються з декількох процесів однієї групи або

					КС КРБ 123.328.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Шелофастов Б.				<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник.</i>		Стадник Н.Б.						
<i>Реценз.</i>								
<i>Н. Контр.</i>								
<i>Затверд.</i>		Осухівська Г.М						
						ТНТУ, каф. КС, гр. СІс-42		

Деякі стихійні лиха (пожежі, обвали, зсуви і навіть землетруси) можуть виникати в результаті дій самих людей, тобто мають антропогенне походження, але наслідки їх завжди є діями сил природи. Для кожного стихійного лиха характерна наявність властивих йому вражаючих чинників, що несприятливо впливають на стан здоров'я, життя людини [19].

Причинами стихійних лих можуть бути:

- швидке переміщення речовини (землетрусу, зсуви);
- вивільнення внутріземної енергії (вулканічна діяльність, землетруси);
- підвищення рівня вод річок, ставків і морів (повені, цунамі);
- вплив надзвичайно сильного вітру (урагани, торнадо, циклони).

Важливо своєчасно провести роботи, спрямовані на локалізацію природного лиха, щоб зменшити зони руйнувань, звести до мінімуму кількість загиблих та постраждалих.

В Україні найчастіше спостерігаються такі надзвичайні ситуації природного характеру:

- небезпечні геологічні явища (зсуви, обвали, осипки, просадки земної поверхні);
- небезпечні метеорологічні явища (зливи, урагани, сильні снігопади, сильний град, ожеледь);
- небезпечні гідрологічні явища (повені, паводки);
- природні пожежі лісових та торф'яних масивів;
- масові інфекції та хвороби людей, тварин, рослин.

В останні роки кількість стихійних лих в Україні та в світі в цілому значно збільшилася. Найчастіше в Україні виникають такі природні катастрофи як землетруси, повені, посухи (на Півдні України), лісові пожежі в літню пору року, снігові замети, зсуви поверхні.

Є серйозні підстави вважати, що масштабність впливу лиха й катастроф на соціальні, економічні, політичні та інших процесів сучасного нашого суспільства та їх драматизм вже перевищили такий рівень, який дозволяв ставитися до них

					КС КРБ 123.328.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

як до локальних збоїв у розміреному функціонуванні державних та громадських структур [18].

Отже, перед людиною та громадськістю в XXI в. вимальовується нова мета - глобальна безпека. Досягти цього можна, в першу чергу, за допомогою зміни світогляду людини, а також покращення системи профілактичних заходів у боротьбі зі стихійними лихами, а саме: вдосконалення рятувальних служб та рятувальної техніки, проведення попереджувальних заходів та пропагандистської роботи з громадянами щодо правил поведінки та дій під час стихійних лих. Це допоможе в майбутньому зменшити кількість загиблих та постраждалих від природних катастроф, а також зменшить матеріальні збитки, що були завдані стихійним лихом.

Природні лиха з часом нікуди не зникнуть. Будуть виникати землетруси в геологічно активних районах, будуть виникати повені, а штормові припливи стануть, раз у раз затопляти морські узбережжя, не обійдеться і пожеж. Людина безсила запобігти природним процесам, але тільки в наших силах зменшити кількість жертв і матеріальних втрат.

4.2 Соціальне значення охорони праці

Соціальне значення охорони праці полягає в сприянні росту ефективності суспільного виробництва шляхом безперервного вдосконалення і поліпшення умов праці, підвищення їх безпеки, зниження виробничого травматизму і профзахворювань [20]. Соціальне значення охорони праці проявляється в зростанні продуктивності праці, збереженні трудових ресурсів і збільшенні сукупного національного продукту.

Охорона праці полягає в сприянні росту ефективності виробництва, яке досягається шляхом безперервного вдосконалення і поліпшення умов праці, підвищення їх безпеки, зниження виробничого травматизму і профзахворювань.

Зростання продуктивності праці відбувається в результаті збільшення фонду робочого часу завдяки скороченню внутрішньо-змінних простоїв шляхом

					КС КРБ 123.328.00.00 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

ліквідації мікротравм або зниження їх кількості, а також завдяки запобіганню передчасного стомлення шляхом раціоналізації і покращення умов праці та введенню оптимальних режимів праці і відпочинку та інших заходів, які сприяють підвищенню ефективності використання робочого часу.

Важливим питанням є зростання продуктивності праці, яка відбувається в результаті збільшення фонду робочого часу завдяки скороченню внутрішньозмінних простоїв шляхом ліквідації мікротравм або зниження їх кількості, а також завдяки запобіганню передчасного стомлення шляхом раціоналізації і покращення умов праці та введенню оптимальних режимів праці і відпочинку та інших заходів, які сприяють підвищенню ефективності використання робочого часу [20].

Особливої уваги заслуговує те, що збереження трудових ресурсів і підвищення професійної активності працюючих відбувається завдяки покращенню стану здоров'я і подовженню середньої тривалості життя шляхом покращення умов праці, що супроводжується високою трудовою активністю і підвищенням виробничого стажу. Підвищується професійний рівень також завдяки зростанню кваліфікації і майстерності. Відповідно і збільшення сукупного національного продукту відбувається завдяки покращенню вищеперелічених показників та їх складових компонентів. Збереження трудових ресурсів і підвищення професійної активності працюючих відбувається завдяки покращенню стану здоров'я і подовженню середньої тривалості життя шляхом покращення умов праці, що супроводжується високою трудовою активністю і підвищенням виробничого стажу. Підвищується професійний рівень також завдяки зростанню кваліфікації і майстерності. Збільшення сукупного національного продукту відбувається завдяки покращенню вищеперелічених показників та їх складових компонентів. Крім того, соціальне значення охорони праці проявляється в зростанні продуктивності праці, збереженні трудових ресурсів.

Комплекс заходів з поліпшення умов праці може забезпечити приріст продуктивності праці на 15-20%. Так, нормалізація освітлення робочих місць

					<i>КС КРБ 123.328.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		51

збільшує продуктивність на 6-13% та скорочує брак на 25%. Раціональна організація робочого місця підвищує продуктивність праці на 21%, раціональне фарбування робочих приміщень – на 25% [21]. Збільшення ефективного фонду робочого часу може бути досягнуто за рахунок скорочення тимчасової непрацездатності працівників внаслідок хвороб та виробничого травматизму.

					<i>КС КРБ 123.328.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		52

ВИСНОВКИ

У ході виконання роботи була розроблена гра, призначена для навчання управлінню дроном для ОС Windows за допомогою графічного рушія Unity, що дозволяє гравцям навчитися керувати дроном. Для реалізації цього проекту було використано сучасні технології.

Результат роботи – розробка має практичну цінність, дає можливість навчитися керувати дроном в ігровій формі. Надалі в нашій грі планується створити досконало новий ландшафт, локація буде максимально наближена до реального середовища для польотів, так само враховані такі фактори як:

- швидкість вітру;
- тиск;
- погода.

Розробка на Unity набагато спрощує реалізацію ігор за допомогою вбудованих інструментів. Також Unity підтримує кросплатформенну технологію, дозволяючи розробнику створювати ігри для різних ОС: Windows, Linux, Android тощо. Більшість початківців розробників ігор використовують рушій Unity, тому що він дає змогу розробляти ігри, не потребуючи для цього якихось спеціальних знань. У Unity застосовується компонентно-орієнтований підхід, у якому розробник будує об'єкти і до них додає різноманітні складові компоненти.

Під час реалізації роботи було використано велику кількість інструментів та готових вбудованих бібліотек. На сьогоднішній день Unity відмінний ігровий рушій, як для новачка, так і для досвідченого розробника ігор.

					КС КРБ 123.328.00.00 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Drone technology uses and applications for commercial, industrial and military drones. URL: <https://www.businessinsider.com/drone-technology-uses-applications> (дата звернення: 12.04.2024).
2. Anatomy of A Drone. URL: <https://www.dronefly.com/the-anatomy-of-a-drone> (дата звернення: 12.03.2024)/
3. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Unity_ігровий_рушій (дата звернення: 21.03.2024).
4. C# documentation. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 23.04.2024)
5. Сучасний підручник з JavaScript. URL: <https://uk.javascript.info/> (дата звернення: 20.04.2024).
6. Unity documentation. URL: <https://docs.unity3d.com/> (дата звернення: 20.03.2024).
7. Розробка комп'ютерних ігор за допомогою Unity 3D: електронний навчальний посібник для підготовки студентів спеціальності 121 «Інженерія програмного забезпечення» / Укладач: О.М. Ляшенко. – Херсон: видавництво ФОП Вишемирський В.С., 2018. 220 с.
8. Розробка ігор. URL: <https://metanit.com/sharp/> (дата звернення: 19.04.2024).
9. Що таке Unity? URL: <https://lemon.school/blog/shho-take-unity> (дата звернення: 24.03.2024).
10. Основи Unity. URL: <https://www.raywenderlich.com/unity/paths/learn> (дата звернення: 27.03.2024).
11. Створення ландшафту у програмі Unity. URL: <https://www.youtube.com/watch?v=y6k17fK8i50> (дата звернення: 27.03.2024).
12. Padfield Gareth. Helicopter flight dynamics: the theory and application of flying qualities and simulation modeling. Blackwell Pub. Oxford, 2008 ISBN: 978–14051–1817–0.

					КС КРБ 123.328.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

13. Розробка ігор. URL: <https://metanit.com/sharp/> (дата звернення: 19.04.2024).

14. Розробка ігор на Unity 5. URL: <https://itproger.com/course/unity> (дата звернення: 27.03.2024).

15. Повний огляд Unity. URL: <http://devgam.com/polnyj-obzor-unity-5> (дата звернення: 05.05.2024).

16. Knowable magazine. URL: <https://knowablemagazine.org/article/mind/2019/video-games-educational-benefits>. (дата звернення: 03.05.2024).

17. Електронний підручник C#. URL: <https://www.tutorialspoint.com/csharp/index.htm> (дата звернення: 24.03.2024).

18. Стеблюк М.І. Цивільна оборона: Підручник. Знання, 2006. 487 с.

19. Толок А.О. Крюковська О.А. Безпека життєдіяльності: Навч. посібник, 2011. 215 с.

20. Агеєв Є .Я. Основи охорони праці: Навчально-методичний посібник для самостійної роботи по вивченню дисципліни. Львів: «Новий Світ – 2000», 2009. 404 с.

21. Основи охорони праці: Підручник.; 3-те видання / За ред. К. Н Ткачука. К.: Основа, 2011. 480 с.

22. Осухівська Г. М., Тиш Є. В., Луцик Н. С., Паламар А. М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с.

23. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.

24. Velychko D., Osukhivska H., Palaniza Y., Lutsyk N., Sobaszek L. Artificial Intelligence Based Emergency Identification Computer System. *Advances in Science and Technology Research Journal*, 18 no. 2, 2024, P. 296-304.

25. Palamar A., Karpinski M., Palamar M., Osukhivska H., Mytnyk M. Remote Air Pollution Monitoring System Based on Internet of Things. CEUR Workshop

					КС КРБ 123.328.00.00 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

Proceedings, *2nd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2022)*, Ternopil, Ukraine, November 22–24, 2022. Vol. 3309. P. 194-204.

					<i>КС КРБ 123.328.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		56

Додаток А.
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“ ____ ” _____ 2024 р

**КОМП'ЮТЕРНА СИСТЕМА НАВЧАННЯ КЕРУВАННЯ ДРОНОМ В
ІГРОВІЙ ФОРМІ**

ТЕХНІЧНЕ ЗАВДАННЯ

на _8_ аркушах

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІс-42

_____ к.т.н. Стадник Н.Б.

_____ Шелофастов Б.О.

« ____ » _____ 2024 р.

« ____ » _____ 2024 р.

Тернопіль 2024

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютерна система навчання керування дроном в ігровій формі».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.328.00.00

1.2 Виконавець

Студент групи СІс-42, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерної інженерії, Тернопільського національного технічного університету імені Івана Пулюя, Шелофастов Богдан Олександрович.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№ 4/7-468 від 26.04.2024 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 27.04.2024 р.

Плановий термін завершення виконання кваліфікаційної роботи – 24.06.2024 р.

1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ГОСТ, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи.

Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Комп'ютерна система навчання керування дроном в ігровій формі призначена для надання навчання управлінню дроном для ОС Windows за допомогою графічного рушія Unity, що дозволяє гравцям навчитися керувати дроном.

До складу системи повинні входити як апаратна складова, так і програмна.

Доцільність створення зумовлена тим фактом, що для навчання оператора реального дрону найкраще відбувається в ігровій форму із використанням симулятора.

2.2 Мета створення системи

Основна мета створення комп'ютерної системи навчання керування дроном в ігровій формі полягає розробці моделі польоту дрону, за допомогою графічного рушія, де користувач міг би навчитися керувати дроном в ігровій формі.

Для того, щоб досягти поставленої мети роботи, необхідно розв'язати наступні задачі:

- розглянути конструкції дронів;
- ознайомитися з роботою програмного рушія Unity;
- вивчити можливості його інструментів та їх властивостей;
- розробити структуру гри;
- спроектувати та впровадити всі її складові елементи;
- протестувати розробку на предмет можливості навчання оператора дрона.

2.3 Характеристика об'єкту

2.3.1 Основні задачі та функції об'єкту

Комп'ютерна система навчання керування дроном в ігровій формі може використовуватись для навчального процесу операторів, котрі дистанційно керують безпілотниками, оскільки пілотування дрона дуже схоже на звичні ігрові симулятори. Розробка повинна допомогти виробити швидкість реакції, навички навігації в повітрі та навички керування літальним апаратом при різних погодних умовах і позаштатних ситуаціях. Розробка повинна розвинути професійні навички оператора, включно із терплячістю та посидючістю

При проектуванні складових системи, зокрема апаратного і програмного забезпечення, необхідно проаналізувати предметну область та розробити концептуальні схеми взаємодії даних.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

Комп'ютерна система навчання керування дроном в ігровій формі навчатиме керувати безпілотником через симулятор. В ході роботи програми повинні підтримуватися правила гри, а також зрозумілий для гравців ігровий процес. По завершенню роботи програми повинні бути збережені усі налаштування користувача.

Розробка повинна забезпечувати використання на платформах Winows. В цілому, у проектованій системі повинні бути забезпечені:

- надійність роботи апаратної частини;
- точність визначення перешкод, в т.ч. і ворожих дронів;
- продуктивність роботи програмного забезпечення;
- часова ефективність та ефективність використання ресурсів системи.

3.1.1 Вимоги до структури та функціонування системи

До структури та функціонування комп'ютерної системи навчання керування дроном в ігровій формі входить:

- персональний комп'ютер із ОС Windows;
- системне та прикладне програмне забезпечення;

В цілому, концептуальна модель комп'ютерної системи навчання керування дроном в ігровій формі повинна описувати предметну область, а саме визначення позиціювання об'єктів та можливість візуалізації об'єктів без використання діапазону передачі даних.

Функціональні вимоги, що висуваються до системи, виглядають наступним чином:

- можливість польоту дрона;
- можливість огинання перешкод;
- можливість знищення ворожих безпілотників;
- можливість керування правами доступу до інформаційних ресурсів;
- часова ефективність на рівні 1 с;
- масштабованість програмної та апаратної складових системи.

3.1.2 Вимоги по діагностуванню системи

Діагностика комп'ютерної системи навчання керування дроном в ігровій формі відбувається у відповідності до затвердженого розкладу профілактичних заходів.

3.1.3 Перспективи розвитку, модернізація системи

Перспективами розвитку та модернізації комп'ютерної системи навчання керування дроном в ігровій формі є можливий перехід на нові версії апаратного та програмного забезпечення, адаптація та інтеграція з додатковими пристроями користувачів. Існуюча апаратна складова системи при цьому не повинна зазнавати значних змін, а програмне забезпечення системи повинно передбачати гнучкість та здатність до масштабування.

3.1.4 Вимоги до надійності системи

Комп'ютерна система навчання керування дроном в ігровій формі повинна бути захищена на кількох рівнях: фізичному та операційній Фізичний рівень захисту повинен забезпечувати надійність щодо доступу до апаратного забезпечення.

На рівні операційної системи повинен бути організований доступ на основі визначених прав доступу до використання програмного забезпечення, що функціонує.

3.1.5 Вимоги до функцій та задач, які виконує система

Функціональні вимоги та задачі, які повинна реалізовувати комп'ютерна система навчання керування дроном в ігровій формі полягають в наступному:

- інформування оператора про успішний прохід перешкод;
- формування зворотного зв'язку при успішній чи невдалій спробі проходження рівня гри;
- надання точних та адекватних результатів;
- забезпечення часової ефективності роботи системи;
- забезпечення зручності використання програмного продукту.

3.1.6 Вимоги до апаратного забезпечення

- процесор – 2,2 ГГц або більш потужний з кількістю логічних ядер >8;
- відеоадаптер з підтримкою DX ;
- RAM – 4 ГБ або більше;
- об'єм дискового простору – 250 МБт +.

3.1.7 Вимоги до програмного забезпечення

- ігровий рушій Unity 2019.3.7f1;
- середовище розробки Visual Studio 2019;
- мова програмування C#;
- операційна система Windows.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу;

- 1 Будова квадрокоптера.
- 2 Схема роботи квадрокоптера.
- 3 Схема гри та порядок дій при навчанні оператора.
- 4 Скріншот роботи симулятора.

*Примітка: У комплект документації можуть вноситися міни та доповнення в процесі розробки.

5 Техніко-економічні показники

Планована собівартість розробки повинна становити не більше 20 000 грн.

*Примітка: собівартість системи може змінюватись під час розрахунку в процесі розробки.

6 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1.	Ознайомлення з завданням до кваліфікаційної роботи	27.04 – 29.04
2.	Розробка технічного завдання	30.04 – 02.05
3.	Підбір джерел про ігрові системи керування дронами	03.05 – 08.05
4.	Опрацювання літературних джерел	09.05 – 12.05
5.	Виконання дослідження щодо розробки системи керування дронами в ігровій формі	13.05 – 16.05
6.	Розроблення програмного коду	17.05 – 20.05
7.	Оформлення розділу «Аналіз технічного завдання»	21.05 – 27.05
8.	Оформлення розділу «Проектна частина»	28.05 – 31.05
9.	Оформлення розділу «Практична частина»	01.06 – 05.06
10.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	06.06 – 08.06
11.	Оформлення кваліфікаційної роботи	08.06 – 11.06
12.	Нормоконтроль	09.06 – 12.06
13.	Перевірка на плагіат	11.06 – 14.06
14.	Попередній захист кваліфікаційної роботи	14.06 – 17.06
15.	Захист кваліфікаційної роботи	25.06

7 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.