

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Острозька академія»
Економічний факультет
Кафедра економіко-математичного моделювання та інформаційних
технологій

КВАЛІФІКАЦІЙНА РОБОТА/ПРОЄКТ
на здобуття освітнього ступеня бакалавра

на тему: **«РОЗРОБКА ІГРОВОГО ЗАСТОСУНКУ В ЖАНРІ SHOOT 'EM UP
НА ІГРОВОМУ РУШІЮ UNITY ДЛЯ ПЛАТФОРМИ ANDROID»**

Виконав: студент 4 курсу, групи КН-41
першого (бакалаврського) рівня вищої освіти
спеціальності 122 Комп'ютерні науки
освітньо-професійної програми «Комп'ютерні науки»
Грищук Микола Володимирович

Керівник:
Гаврильчик Леонід Сергійович

Рецензент:
Місай Володимир Віталійович

РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ

Завідувач кафедри економіко-математичного моделювання та інформаційних
технологій

_____ (проф., д.е.н. Кривицька О.Р.)

Протокол № ____ від « ____ » _____ 2022 р.

Острог, 2022

Міністерство освіти і науки України
Національний університет «Острозька академія»

Факультет: економічний

Кафедра: економіко-математичного моделювання та інформаційних технологій

Спеціальність: 122 Комп'ютерні науки

Освітньо-професійна програма: Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри економіко-математичного моделювання
та інформаційних технологій

_____ Ольга КРИВИЦЬКА

«_____» _____ 20__ р.

ЗАВДАННЯ
на кваліфікаційну роботу/проект студента

Грищука Миколи Володимировича

1. Тема роботи/проекту: Розробка ігрового застосунку в жанрі shoot'em up під мобільну систему android на рушію unity

керівник роботи/проекту: Гаврильчик Леонід Сергійович

Затверджено наказом ректора НаУОА від 29 жовтня 2021 року №110.

2. Термін здачі студентом закінченої роботи/проекту: 03 червня 2022 року.

3. Вихідні дані до роботи/проекту: постановка задачі, аналіз аналогів, вихідні дані.

4. Перелік завдань, які належить виконати: опис предметного середовища; огляд наявних аналогів; постановку задачі; опис архітектури рішення, що розробляється; опис коду та інтерфейсу програми; тестування.

5. Перелік графічного матеріалу: діаграма прецедентів функціональних вимог, діаграма станів ігрового циклу, діаграма станів системи меню.

6. Консультанти розділів роботи/проєкту:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Гаврильчик Л.С.	01.12.2021р.	01.12.2021р.
2	Гаврильчик Л.С.	01.12.2021р.	01.12.2021р.
3	Гаврильчик Л.С.	01.12.2021р.	01.12.2021р.

7. Дата видачі завдання: 01.12.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи/проєкту	Строк виконання етапів	Примітка
1	Затвердження теми роботи/проєкту	до 01.11.2021 р.	
2	Постановка завдання	до 01.12. 2021 р.	
3	Розробка архітектури та загальної структури системи	до 01.02.2022 р.	
4	Розробка структур окремих підсистем	до 01.03. 2022 р.	
5	Програмна реалізація системи	до 01.05.2022 р.	
6	Попередній захист кваліфікаційної роботи/проєкту	до 01.06.2022р.	
7	Здача кваліфікаційної роботи/проєкту на кафедру	03.06.2022 р.	

Студент: _____ Микола ГРИЩУК

Керівник кваліфікаційної роботи: _____ Леонід ГАВРИЛЬЧИК

АНОТАЦІЯ
кваліфікаційної роботи/проєкту
на здобуття освітнього ступеня бакалавра

Тема: Розробка ігрового застосунку в жанрі shoot 'em up на ігровому рушію Unity для платформи Android.

Автор: Грищук М.В.

Науковий керівник: Гаврильчик Леонід Сергійович

Захищена «.....»..... 20__ року.

Пояснювальна записка до кваліфікаційної роботи: 40 стор., 21 рис., 1 табл., 14 джерел.

Ключові слова: ігровий застосунок, 2D гра, Android, мова програмування C#, інтегроване середовище розробки, графічний редактор.

Короткий зміст праці:

Кваліфікаційна робота на тему: «Розробка ігрового застосунку в жанрі shoot 'em up на ігровому рушію Unity для платформи Android» присвячена розробці 2D гри з можливістю користування на платформі Android. Актуальність обраної для кваліфікаційної роботи/проєкту теми пов'язана з великою популярністю мобільних ігрових додатків. У процесі роботи використано такі сучасні засоби як: ігровий рушій Unity, мова програмування C#, інтегроване середовище розробки Microsoft Visual Studio Code, графічний редактор Adobe Photoshop.

Qualification thesis on the topic: "Development of a game application in shoot'em up genre for the Android mobile system on the Unity engine" is dedicated to the development of 3D games with the ability to use on the Android platform. The relevance of the topic chosen for the qualifying work is due to the great popularity of mobile gaming applications. In the process of work such modern means were used as: Unity game engine, C # programming language, Microsoft Visual Studio Code integrated development environment, Adobe Photoshop graphic editor.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ	6
ВСТУП	7
РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ	9
1.1. Опис предметного середовища (функціональної моделі, процесу діяльності)	9
1.2 Огляд існуючих додатків за даною тематикою	15
1.3 Постановка задачі	20
Висновок до розділу 1	20
РОЗДІЛ 2.ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	21
2.1 Аналіз предметної області	21
2.2 Проектування системи	22
2.3 Математичне та алгоритмічне забезпечення	23
Висновок до розділу 2	27
РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	28
3.1. Засоби розробки	28
3.2 Вимоги до технічного та програмного забезпечення	29
3.3. Опис програмної реалізації	30
3.3.1. Робота з графікою	30
3.3.2 Розробка меню гри	30
3.3.3 Інтерфейс та органи керування	32
3.3.4 Розробка Player та PlayerController	33
3.3.5 Розробка CameraController	34
3.3.6 Розробка Enemy	34
3.3.7 Робота над логікою гри	36
3.3.8 Робота з фізикою гри	37
Висновок до розділу 3	37
РОЗДІЛ 4. ТЕСТУВАННЯ	38
4.1 Функціональне тестування	38
4.2 Usability Test	39
Висновок до розділу 4	39
ВИСНОВКИ	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

- **OS (OC)** – операційна система
- **ПЗ** – програмне забезпечення
- **API** - Інтерфейси прикладного програмування
- **APK** – Комплект додатків
- **GUI** - Графічний інтерфейс користувача
- **SDK** - Комплект для розробки програмного забезпечення
- **CLI** - Спільна мовна інфраструктура
- **CLR** - Спільна мова виконання
- **SSAO** - Screen space ambient occlusion (заломлення світла в екранному просторі)
- **Collider** (колайдер) - ігровий об'єкт, до якого прикріплений колайдер та нормальний не кінематичний Rigidbody. Rigidbody колайдери повністю симулюються фізичним двигуном і можуть реагувати на колізії та сили, прикладені зі сценарій.
- **Сценарій** - це невелика програма, яка містить послідовність дій, створених для автоматичного виконання завдання.
- **Спрайт** - двомірне зображення, що застосовується в комп'ютерній графіці. Частіше за все — растрове зображення, що вільно переміщується по екрану.
- **Android** - операційна система для мобільних платформ. Розроблена на базі Linux компанією Android, Inc.
- **Кросплатформеність** - здатність застосунку функціонувати на різних апаратних платформах і операційних системах.
- **Сцена** - сукупність графічних моделей розташованих в певному порядку.

ВСТУП

Людині XXI віку важко уявити своє життя без високотехнологічних пристроїв, як-то смартфони, планшетні ПК, бо саме ці науково-технічні новинки дозволяють не тільки підтримувати зв'язок з людьми із різних куточків світу, а й допомагають у формуванні та розвитку сучасного інформаційного простору.

Сфера ІТ прогресує кожного дня. Розробки у сфері мобільних додатків стають дедалі більш різноманітними, бо навіть побутова техніка(телевізори, холодильники і т.д.) зараз керується під Android OS. Сьогодні за допомогою смартфона можна не тільки робити високоякісні зображення, а й переглянути інформацію стосовно будь-кого або - чого (сторінки у соц.мер., блогах, інфо.порталах і т.д.).

Наявність сучасних технологій дає нам змогу не тільки полегшити життя, а ще й економить наш час. Інформація та дані все частіше розглядаються як життєво важливі ресурси, які повинні бути організовані таким чином, щоб ними можна було легко користуватися.

В наш час, найпоширенішою оперативною системою для мобільних пристроїв є ОС Android. Android підтримує велику кількість засобів розробки та девайсів від різних виробників. Головна причина поширення ОС Android - безкоштовні засоби розробки, в той час як розробка під систему iOS вимагає високих початкових витрат.

Розробка даного ігрового додатку має на меті створити за допомогою сучасного інтегрованого середовища розробки та ігровою рушію Unity, що містить всі необхідні інструменти та бібліотеки для створення застосунків для платформи Android

Мета дослідження – створення програмного ігрового додатку для мобільних пристроїв на базі Android.

Задачі дослідження:

- Аналіз предметної області;
- Проектування інформаційної системи;
- Вибір інструментарію, методів реалізації та тестування ПП.

Об'єктом дослідження виконання роботи є процес аналізу мобільних ігрових застосунків у жанрі shoot'em up та вивчення їх ігрових механік.

Предметом дослідження є технології: ігровий рушій Unity, двовимірний графічний редактор Adobe Photoshop, мова програмування C#.

Для досягнення мети та вирішення завдання реалізації мобільного ігрового додатку необхідно здійснити:

- Опис предметного середовища;
- Огляд наявних аналогів;
- Постановку задачі;
- Опис архітектури рішення, що розробляється;
- Опис коду та інтерфейсу програми;
- Тестування.

РОЗДІЛ 1

ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1. Опис предметного середовища (функціональної моделі, процесу діяльності)

Для реалізації застосунку «SpaceBlaze» необхідно було вибрати технології, які будуть використовуватись для його створення. Було обрано наступні технології для роботи: мова програмування C#, ігровий рушій Unity, середовище розробки Visual Studio Code, графічний редактор Photoshop.

Платформа Android

Наявність операційної системи (ОС) - головна особливість, яка відрізняє смартфон від звичайного мобільного телефону. При виборі конкретної моделі телефону або пристрою, операційна система часто є визначальним фактором.

Android – операційна система і платформа для мобільних телефонів, планшетів, електронних книг, навігаторів, смарт годинників, фітнес-браслетів, фотоапаратів, ігрових приставок, холодильників, ноутбуків, окулярів Google Glass, телевізорів та інших пристроїв.

Платформа заснована на власній реалізації віртуальної машини Java, а також ядрі Linux для мобільних пристроїв. Операційною системою володіє Google, яка купила її в 2005 році в компанії Android.Inc., яку потім купила Google. Android відкриває можливість створенню Java-додатків, які керують пристроєм через бібліотеки, розроблені Google.

Крім того, можна писати програми на C та іншими мовами програмування за допомогою Android Native Development Kit 1.5 (Cupsake), випущеної 30 квітня 2009 р. Серед основних поліпшень з'явилася підтримка запису і перегляду відео в режимі камери; підтримка 19 Bluetooth A2DP; можливість автоматичного підключення до Bluetooth гарнітури.

Розробка мобільних ігрових додатків має низку послідовних етапів. Але щоб створити мобільний додаток потрібен певний інструмент, особливо, якщо цей додаток ігровий. На практиці використовуються рушії, де вже є реалізована велика кількість потрібних функцій для створення ігрових додатків

Мова програмування C#

C# – об'єктно-орієнтована мова програмування, створена спеціально для роботи у середовищі Microsoft .NET Framework. Мова C# була розроблена з урахуванням сильних і слабких особливостей інших мов, зокрема Java і C++. Специфікація мови C# була написана Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде. Андерс Хейлсберг відомий у світі програмування як автор компілятора Turbo Pascal і лідер команди, яка створила Delphi.

Ключові особливості мови C#:

- Компонентна орієнтованість
- Код зібраний воедино (декларації і реалізації об'єднані разом)
- Уніфікована система типів і їх безпечність
- Автоматична і мануальна робота за пам'яттю
- Використання єдиної бібліотеки класів - CLR

Специфікація C# визначає мінімальний набір бібліотек типів і класів, на який має розраховувати компілятор. На практиці, C# найчастіше використовується з якоюсь реалізацією Common Language Infrastructure.

Задля зручності розробки на проекті був використаний редактор коду Visual Studio Code.

Visual Studio Code - це легкий, але потужний редактор вихідних кодів, який працює на робочому столі та доступний для Windows, macOS та Linux. Вона підтримує JavaScript, TypeScript і Node.js і має багату екосистему розширень для інших мов.

Редактори коду дозволяють зручно та легко розробляти інтерфейси

програм, чи веб-сайтів, мають настроюваний зовнішній вигляд, підсвічування синтаксису, можливість додавання сторонніх плагінів для покращення взаємодії з користувачем, що дає високі переваги над типовими текстовими процесорами.

Переваги Visual Studio Code:

- Безкоштовний.
- Велика кількість плагінів.
- Динамічний набір тексту.
- Підтримка Git.
- Невелика вага – редактор не потребує значних ресурсів операційної системи.

Unity - це повноцінний рушій, багатоплатформний інструмент, спрямований на створення 2D та 3D застосунків в одному редакторі. Безліч популярних мобільних ігрових продуктів створені саме на цьому рушії, такі як: Hearthstone: Heroes of Warcraft, Age of Magic, Royal Blood та інші.

На веб-сайті Unity представлено кілька тарифних планів рушія, відрізняються вартістю і рівнем підтримки (рис. 1). Кожна з версій цього програмного забезпечення відкриває для користувача новий функціонал, який відсутній в базовій версії. Відмінності версій один від одного представлені на рисунку 1.1

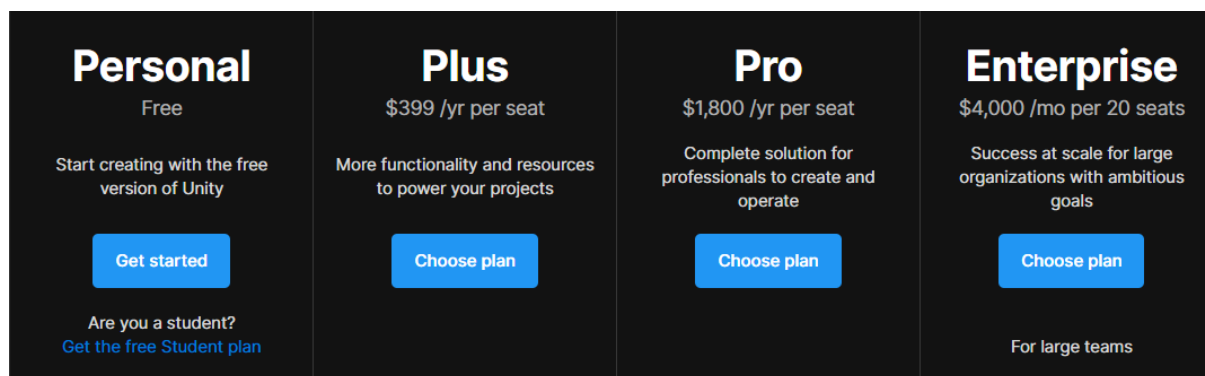


Рис. 1.1. Вартість передплати рушія Unity

Так само можна побачити основні відмінності версій та їх вартість. У нашому випадку буде використана версія Personal, вона є безкоштовною, не має обмежень для проекту, а найголовніше, має практично все необхідне для старту розробки з нуля.

Це потужне, але в той самий час просте в роботі ПЗ, що дозволяє створювати і випускати 2D і 3D-ігри. Розробка мобільних ігор на Unity відкриває перед розробниками безліч можливостей платформи для підтримки та монетизації створених ігор.

У 2008 році Unity почали співпрацювати з iOS, з Android — у 2010, а далі розробники змогли створювати свої проекти і для ігрових консолей Xbox і Playstation.

Понад 50% всіх мобільних ігор розроблено саме на Unity. Спочатку Unity призначався для розробки на комп'ютерах Mac, пізніше з'явилося оновлення, що дозволяє працювати з Windows.

Плюси рушія Unity:

- Зрозумілий редактор та інструментарій: за декілька днів основні речі може освоїти навіть той, хто вперше стикається з розробкою ігрового застосунку. А якщо питання залишаються — відповіді є на одному з багатьох ресурсів, форумів, та у відеоуроках на YouTube. Створення гри на Unity буде під силу навіть школяреві.
- Сучасний рівень графіки, здатний конкурувати з більш дорогими рушіями. Unity, безумовно, програє UnrealEngine за можливостями, але може працювати із освітленням, стандартним набором постпроцессингових ефектів, SSAO.
- Ігровий рушій Unity поширюється умовно безкоштовно. Платити потрібно буде тільки за розширення пакетів підписки. На ліцензії кілька разів на рік бувають знижки, зазвичай -20%.
- Велика спільнота розробників.
- Безліч опублікованих ігор.

- Внутрішній магазин готових рішень, де можна купити готові фрагменти коду, текстури та звуки.
- Можливість створення фотореалістичної графіки.
- Розробка на Unity дозволяє легко імпортувати створені програмні застосунки між ОС Windows, Linux, OS X, Android, iOS, на консолі PlayStation, Xbox, Nintendo, на VR- і AR-пристрої.

Мінуси середовища розробки Unity:

- Розробка гри на Unity вимагає навичок програмування.
- Безліч вбудованих компонентів роблять продукт об'ємним. Це може стати проблемою, адже користувачі не люблять завантажувати великі ігри, а в деяких країнах (наприклад, Індія та Бразилія) люди користуються недорогими, слабкими гаджетами, які не потягнуть вашу гру.
- У розробників немає доступу до вихідного коду власної гри. Вам доведеться чекати, поки інженери Unity самі це зроблять. початкових кодів вам не дадуть навіть по платній ліцензії.
- Немає інтеграції із зовнішніми сервісами та бібліотеками (наприклад, Facebook), розробники змушені налаштовувати це вручну.
- Неможливість додати до рушія сторонню фізику.

Adobe Photoshop - графічний редактор, розроблений і поширюваний фірмою Adobe Systems. Цей продукт є лідером ринку в області комерційних засобів редагування растрових зображень, і найвідомішим продуктом фірми Adobe. Часто цю програму називають просто Photoshop (Фотошоп). У наш час Photoshop доступний на платформах Mac OS X/Mac OS і Microsoft Windows.

Попри те, що спочатку програма була розроблена як редактор зображень для поліграфії, в наш час вона широко використовується і у веб-дизайні. У більш ранній версії була включена спеціальна програма для цих цілей - Adobe ImageReady, яка була виключена з версії CS3 за рахунок інтеграції її функцій в сам Photoshop, а також включення в лінійку програмних продуктів Adobe Fireworks, що перейшло у власність Adobe після придбання компанії Macromedia.

Photoshop тісно пов'язаний з іншими програмами для обробки медіафайлів, анімації та іншої творчості. Спільно з такими програмами, як Adobe ImageReady (програма скасована у версії CS3), Adobe Illustrator, Adobe Premiere, Adobe After Effects и Adobe Encore DVD, він може використовуватися для створення професійних DVD, забезпечує засоби нелінійного монтажу і створення таких спецефектів, як фони, текстури і т. д. для телебачення, кінематографу і всесвітньої павутини. Основний формат Photoshop, PSD, може бути експортований і імпортований всіма програмними продуктами, переліченими вище. Photoshop CS підтримує створення меню для DVD. Спільно з Adobe Encore DVD, Photoshop дозволяє створювати меню або кнопки DVD. Photoshop CS3 у версії Extended підтримує також роботу з тривимірними шарами.

Серед можливостей Adobe Photoshop можна виділити наступні:

- висока якість обробки графічних зображень;
- зручність і простота в експлуатації;
- великі можливості, які дозволяють виконувати будь-які операції створення і обробки зображень;
- широкі можливості автоматизації обробки растрових зображень, які базуються на використанні сценаріїв;
- сучасний механізм роботи з кольоровими профілями, які допускають їх втілення в файли зображень з метою автоматичної корекції кольорових параметрів при виводі на друк для різних пристроїв;
- великий набір команд фільтрації, за допомогою яких можна створювати найрізноманітніші художні ефекти.

Через високу популярність Photoshop підтримка його формату файлів, PSD, була реалізована в його основних конкурентів, таких, як Macromedia Fireworks, Corel PHOTO-PAINT, Pixel image editor, WinImages, GIMP, Jasc Paintshop Pro і т. д.

Photoshop підтримує такі колірні моделі або способи опису кольорів зображення (в нотації самої програми — режим зображення):

- RGB
- LAB
- CMYK
- Grayscale
- Bitmap
- Duotone
- Indexed
- Multichannel

Підтримується обробка зображень, як з традиційною глибиною кольору (8 біт, 256 градацій яскравості на канал), так і з підвищеною (16 біт, 65536 відтінків в кожному каналі). Можливе збереження у файлі додаткових елементів, як то: направляючих (Guide), каналів (наприклад, каналу прозорості — Alpha channel), шляхів обтравки (Clipping path), шарів, що містять векторні і текстові об'єкти. Файл може включати колірні профілі (ICC), функції перетворення кольору (transfer functions).

1.2. Огляд існуючих додатків за даною тематикою

Shoot 'em up («перестріляти їх всіх») — жанр відеоігор, в яких гравець, керуючи персонажем або технічним засобом, бореться з великою кількістю ворогів за допомогою стрілянини. Процес гри часто зображений в дуже стилізованій манері. У короткому варіанті, поширеному серед фанатів жанру, має назву «шмап» (анг. *shmup*). У Японії такі ігри відомі як «shooting games» («ігри зі стріляниною», також використовується аббревіатура «STG»). На піку популярності вони часто називалися просто як «shooter», тобто «стрілялка» і в більшості були двовимірними. Але з появою тривимірних ігор жанр став більш збірним.

Жанр зародився, як прийнято вважати, на аркадних ігрових автоматах з гри Space Invaders (1978). Витоки жанру можна простежити до «Spacewar!», однієї з найбільш ранніх комп'ютерних ігор, розробленої в 1961 році і в кінцевому рахунку випущеної для ігрових залів на початку 1970-х років. Тим не менш, винахід жанру приписують Томохиро Нишикидо, творцеві Space Invaders. Процес цієї гри полягав у протистоянні гравця і великої кількості ворогів, що з'являються з верхнього краю екрана і з кожним наступним рівнем збільшують швидкість руху та темп стрільби. Така концепція існує і сьогодні: в майже кожному вертикальному скролл-шутері гравець розташовується «носом» до верху екрана, звідки з'являються противники. Нововведенням жанру є тактика «dodge and shoot» — маневрування з метою ухилитися від пострілів і відповідна стрілянина.

Гра «Space Invaders» мала і великий комерційний успіх, що навіть викликало нестачу монет для ігрових автоматів в Японії. Свій варіант гри поспішила випустити майже кожна компанія, що займається виробництвом ігрових автоматів. Деякі з цих варіантів, такі як «Space Stranger» і «Super Invader Attack» були майже клонами, які навряд чи що додали до геймплею оригіналу. Інші, наприклад Galaxian від компанії Namco продовжили розвиток жанру, додавши більш складні траєкторії руху ворогів і покращену графіку.

За аналог було взято Android-додаток гри «Thunder Legend X».



Рис. 1.2. Меню Android-додатку «Thunder Legend X»

Даний представник жанру Action Shooter, розроблений Purplebeard, тримає гравців на краю свого місця, ухиляючись від ворожих кораблів і куль.

Мета Thunder Legend X — переміщатися своїм кораблем по екрану, торкаючись і перетягуючи пальцем. Особливостями даного проекту є:

- Різноманітність кораблів, проте гравець починає з одного основного корабля, поки не отримає достатньо зірок, щоб придбати інший. (Рис. 1.3)
- Можливість переміщувати свій корабель в будь-яке місце на екрані, гравець не обмежений невеликою площею.
- Гравці можуть оновлювати такі речі, як здоров'я, зброю, а також скорочувати час перезарядки щита та суперзброї. (Рис. 1.4)
- У Thunder Legend X також є різні рівні складності, але ви можете отримати доступ до вищих рівнів складності лише після того, як покращите один або два кораблі. (Рис. 1.5)



Рис. 1.3. Меню вибору корабля

- Зброя в Thunder Legend X повністю автоматизована, окрім «суперзброї» та щита. «Суперзброя» може бути використана один раз, після чого повинна охолонути, перш ніж її можна буде використовувати знову.

Поряд із безліччю кораблів на вибір, Thunder Legend X дозволяє гравцям налаштовувати свій корабель, витрачаючи зірки на оновлення.

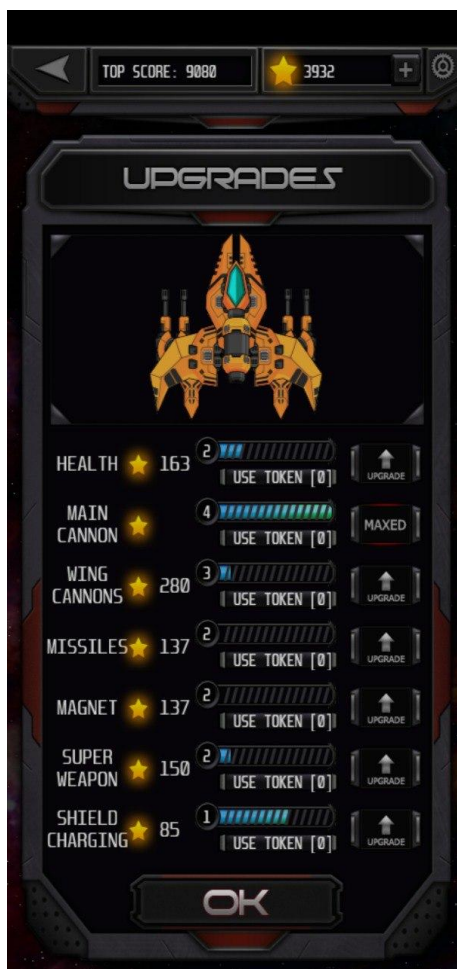


Рис. 1.4. Меню покращення корабля

Коли корабель гравця стає кращим, ви зможете виживати довше і заробляти більше зірок, граючи в рівні.



Рис. 1.5. Меню рівнів складності

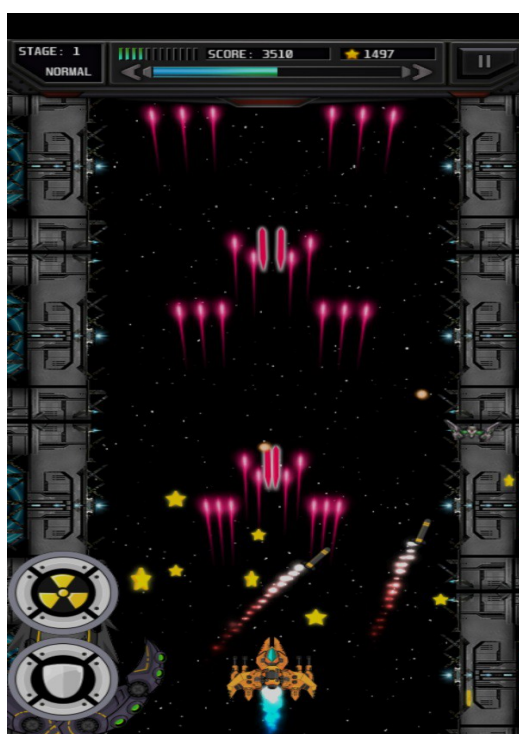


Рис. 1.6. Екран гри

Загалом, Thunder Legend X — це насичений космічний шутер, який весело грати і досить легко керувати.

1.3. Постановка задачі

Метою проекту є створення двовимірного мобільного ігрового застосунку для платформи Android версії 5.0 і вище із використанням мови програмування С#. Додаток має назву «SpaceBlaze» та виконаний у жанрі shoot 'em up.

Ігровий додаток повинен містити наступні механіки:

- Shooting,
- Switching weapons,
- Score points.

У додатку також повинні бути присутні елементи меню із кнопками з наступним функціоналом:

- Пауза
- Відновлення гри після паузи
- Вихід з гри

Керування грою буде проводитися екранними контролерами - один для руху літака гравця в просторі, інший - для пострілів з зброї.

На екрані гри також буде присутнім елемент кількості очок гравця, які будуть здобуватися за знищення об'єктів та ворожих літаків, а також полоса, що буде графічно висвітлювати об'єм здоров'я літака гравця.

Висновок до розділу 1

В першому розділі описана постановка задачі та вимоги до кінцевого продукту, а також наведено приклад мобільного додатку та середовища розробки, які було обрано для виконання проекту. Були окреслені їх плюси та мінуси. Наступний розділ буде присвячений вибору інструментів для розробки.

РОЗДІЛ 2

ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Аналіз предметної області

В даний час комп'ютерні технології стали активно застосовуватися в різних сферах нашого життя. Створення різних простих і складних комп'ютерних програм для різних областей знань постійно розвивається.

Залежно від віку дитини комп'ютерна програма може виступати в якості опонента по грі, бути оповідачем, репетитором, екзаменатором. Вже існують комп'ютерні програми, спрямовані на розвиток різних психічних функцій дітей, таких, як візуальне і слухове сприйняття, увагу, пам'ять, словесні і логічні міркування і т.д, які можуть бути успішно використані дітьми старшого дошкільного і молодшого шкільного віку. Тепер на зміну настільним комп'ютерам приходять мобільні - планшетні ПК, смартфони, що дозволяє більш зручно працювати з усією інформацією. Люди грають, щоб скоротати час, або просто отримати дозу позитиву

Гра - це пізнавальна діяльність, вона є свого роду практичною формою мислення про природу навколишнього дитини соціальної реальності. Через особливості гри, дитина в грі вперше долучається до абстрактного мислення.

Комп'ютерні ігри - це новий вид розвиваючого навчання. Комп'ютерні технології відкривають нові можливості для використання педагогічних методів:

- використання матеріалу різної складності. Індивідуально дитині завжди можна запропонувати саме те, що в даний момент відповідає його можливостям і цілям навчання;
- для того щоб "бачити" проблеми в розвитку дитини, які важко виявити в

традиційному навчанні;

- для того щоб сформувати розуміння дитиною процесу отримання власних навичок;

2.2. Проектування системи

Створення ігрового застосунку є дуже трудомістким і складним процесом. В ході якого величезна кількість елементів об'єднуються в одну комплексну програму, призначену для того, щоб розважати того, хто буде нею користуватись.

Система керування додатком представлена у вигляді коду мовою програмування C#. Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML.

Для початку з допомогою рушія Unity ми виділили основні пакети із сценаріями гри:

- `GamePlay` - відповідальний за хід самої гри, керування гравця, ворожі об'єкти та зброю.
- `GameOver` - закінчення гри за програшу гравця, або пауза
- `MainMenu` - головне меню та його функціональні клавіші

Також у додатку присутні сценарії, що відповідають за джойстики керування гравця та пострілів зі зброї.

У наборі з назвою `GamePlay` присутні сценарії керування гравцем, об'єм його здоров'я, а також кількість балів, отриманих за знищення ворожих об'єктів. Прописані вороги та система їх поведінки, за якою вони завжди слідує за гравцем та намагаються його знищити. Присутня система підвищення складності гри за досягнення певної кількості балів, а також покращення зброї та збільшення обсягу здоров'я гравця.

Сценарії з набору GameOver виконуються у випадку, якщо здоров'я гравця падає до нуля, або ж гравець сам ставить гру на паузу, чи взагалі виходить з неї.

Контролер головного меню MainMenu містить в собі сценарії із кнопками, що відповідають за меню паузи, відновлення гри та вихід з неї.

На стадії розробки гри вкрай необхідно приділити увагу графічній, візуальній складовій гри, від її якості залежить популярність гри, а значить і кількість її продажів.

Але й не варто сильно захоплюватися цим процесом, так як це може позначитися на підсумковій продуктивності ігрової програми і відповідно погіршити такі показники як фреймрейт або fps (frames per second), відгук управління і плавність ігрового процесу, обсяг займаного іграми платформи простору і часом завантажити програму з мережі.

2.3. Математичне та алгоритмічне забезпечення

Сценарії є невід'ємним компонентом усіх програм, які створюються з допомогою Unity. Більшість програм потребує сценарії, щоб реагувати на вказівки гравця та виконувати події в ігровому процесі тоді, коли вони повинні відбуватися. Крім того, сценарії можна використовувати для створення графічних ефектів, керування фізичною поведінкою об'єктів або навіть для реалізації спеціальної системи штучного інтелекту для персонажів та об'єктів у грі.

Редактор сценаріїв є дуже потужною функцією. Це дозволяє розробникам створювати інструменти та розширювати редактор Unity для виконання спеціальних завдань, які можуть допомогти у розробці вашого проекту.

Ви можете визначити, як властивості відображаються в переглядачі за допомогою спеціальних ящиків властивостей. Ви можете створити будь-яку кількість користувацьких вікон у вашому проекті, вони поведуться так само, як і переглядач, сцена або будь-які інші вбудовані вікна Unity. Це чудовий спосіб додати інтерфейс користувача до підсистеми для вашої гри.

Щоб розширити редактор, потрібно виконати кілька простих кроків: Створити сценарій, що походить від EditorWindow. За допомогою коду ініціюємо відображення самого вікна. Реалізуємо код GUI для свого інструменту. Створимо просте вікно редактора, яке матиме кілька елементів для початку.

1. Створюємо у проекті нову папку під назвою «Редактор»

2. У папці «Editor» створюємо новий сценарій із назвою «EditorScripting»

(Рис 2.1).

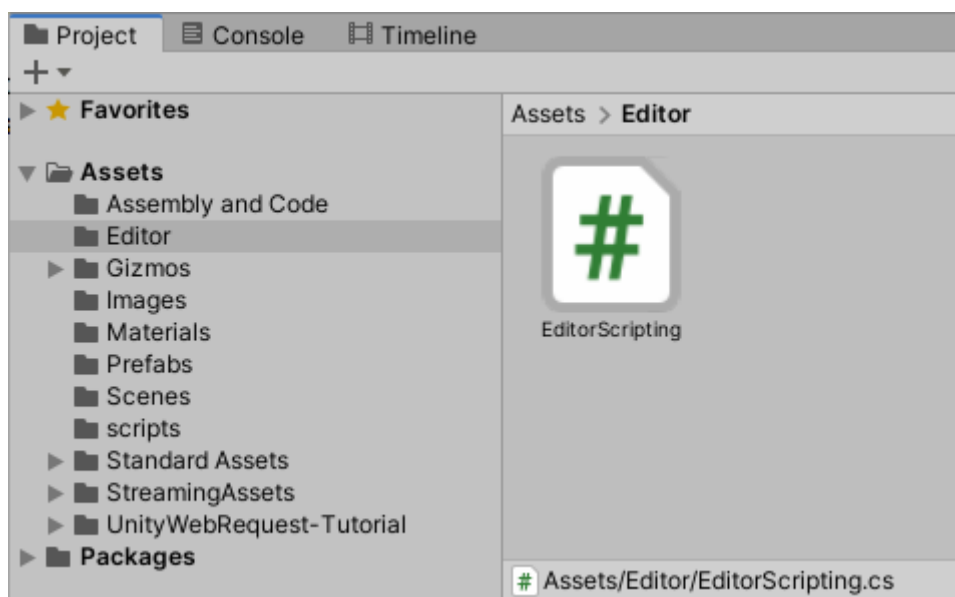


Рис. 2.1. Сценарій, поміщений у папку Editor

3. Відкриваємо сценарій. Щоб отримати сценарій, успадкований від модуля EditorWindow, додаємо у код «using Unity Editor;» у рядку 4, а потім замінюємо «MonoBehaviour» на «EditorWindow» у рядку 6. Це дозволить успадкувати багато елементів керування і функціональні можливості Windows Editor Unity, наприклад, можливість закріпити і зберегти їх положення (Рис 2.2).


```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEditor;
5
6  public class EditorScripting : EditorWindow
7  {
8
9      private void OnGUI()
10     {
11         // OnGUI Method Contents
12     }
13 }
14

```

Рис. 2.2. Вигляд EditorWindow

Наступний крок, який потрібно виконати, це зробити відображення вікна на екрані. Це можна зробити за допомогою функції, яка активується властивістю MenuItem (Рис. 2.3).

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEditor;
5
6  public class EditorScripting : EditorWindow
7  {
8
9      [MenuItem("Window / Custom Controls")]
10
11     public static void ShowWindow()
12     {
13         EditorWindow.GetWindow(typeof(EditorScripting));
14     }
15
16     private void OnGUI()
17     {
18         // OnGUI Method Contents
19     }
20 }
21

```

Рис. 2.3. Вигляд коду для показу вікна

4. Додаємо `[MenuItem("Window/Custom Controls")]` — цей рядок додає вікно до меню Window, таким чином, отримуємо «Window / Custom Controls».

5. Замінюємо «Custom Controls» на будь-яку назву, яку ми бажаємо дати вікну.

6. Додаємо «EditorWindow.GetWindow(typeof(EditorScripting));» - цей рядок створить вікно. При його виборі, буде відображено меню в Unity Engine, яке успадкує всі основні функції від всіх вікон редактора.

Код створить вікно редактора, яке можна закріпити. Воно діятиме як будь-яке інше вікно Unity і зберігатиме свою позицію між використанням. Вікно можна закріпити так само, як і будь-яке інше вікно, а також можна використовувати з користувацькими макетами.

Фактичний вміст вікна відображається шляхом реалізації функції OnGUI. Використання класів EditorGUI та EditorGUILayout лише для редактора дозволить вам змішувати і поєднувати елементи керування.

```

public class EditorScripting : EditorWindow
{
    string custString = "String Here";
    bool groupEnabled;
    bool optionalSettings = true;
    float jumpMod = 1.0f;
    float impactMod = 0.5f;

    // Show the Window.
    [MenuItem("Window / Custom Controls")]

    public static void ShowWindow()
    {
        EditorWindow.GetWindow(typeof(EditorScripting));
    }

    // Add the Controls
    void OnGUI()
    {
        GUILayout.Label("Base Settings", EditorStyles.boldLabel);
        custString = EditorGUILayout.TextField("Text Field", custString);

        groupEnabled = EditorGUILayout.BeginToggleGroup("Optional Settings", groupEnabled);
        optionalSettings = EditorGUILayout.Toggle("Double Jumping Enabled", optionalSettings);
        jumpMod = EditorGUILayout.Slider("Jump Modifier", jumpMod, -5, 5);
        impactMod = EditorGUILayout.Slider("Impact Modifier", impactMod, -5, 5);
        EditorGUILayout.EndToggleGroup();

        GUI.backgroundColor = Color.red;

        GUILayout.FlexibleSpace();
        EditorGUILayout.BeginHorizontal();
        GUILayout.FlexibleSpace();
        if (GUILayout.Button("Reset", GUILayout.Width(100), GUILayout.Height(30)))
        {
            custString = "String Here";
            optionalSettings = false;
            jumpMod = 1.0f;
            impactMod = 0.5f;
        }
        EditorGUILayout.EndHorizontal();
    }
}

```

Рис. 2.4. Код вікна з елементами керування та дизайну

Це загальні інструменти та знання, щоб розширити просте вікно редактора, створене за цим посібником. Його можна перетворити на «шахрайське» меню, або додати в конвеєр виробництва для дизайнів ігор. Як би там не було, це дуже корисний навик, щоб мати можливість розширити движок Unity для конкретних потреб.

Висновок до розділу 2

У другому розділі було проведено аналіз предметної області та підготовлено постановку задачі, виконано розробку архітектури ігрового додатку, визначено основні пакети сценаріїв гри.

РОЗДІЛ 3

ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Засоби розробки

Засоби розробки включають технічну документацію Unity для роботи з інтерфейсом і бібліотеками, які використовуються при створенні ігрового продукту, публікації на відеохостингу YouTube, а також статті в мережі Інтернеті.

Візуалізатор Unity. В інструменті розробки Unity реалізована велика кількість інструментів для створення додатків і додавання спеціальних під'єднаних модулів.

Також в Unity влаштований простий редактор Drag & Drop інтерфейс, який легко налаштовується, його функціонал включає велику кількість різних комбінацій екранів, за допомогою яких можна виконувати тонкі налаштування всього проекту відразу всередині самого редактора. В Unity є підтримка таких мов програмування як C#. Розробка кожного проекту в Unity починається з розробки незалежних сцен та окремих файлів. Також в самому русії є свій магазин, в якому можна знайти багато різних об'єктів, анімацій, ефектів для цих анімацій та ще велику кількість налаштувань для проекту (рис 3.1).

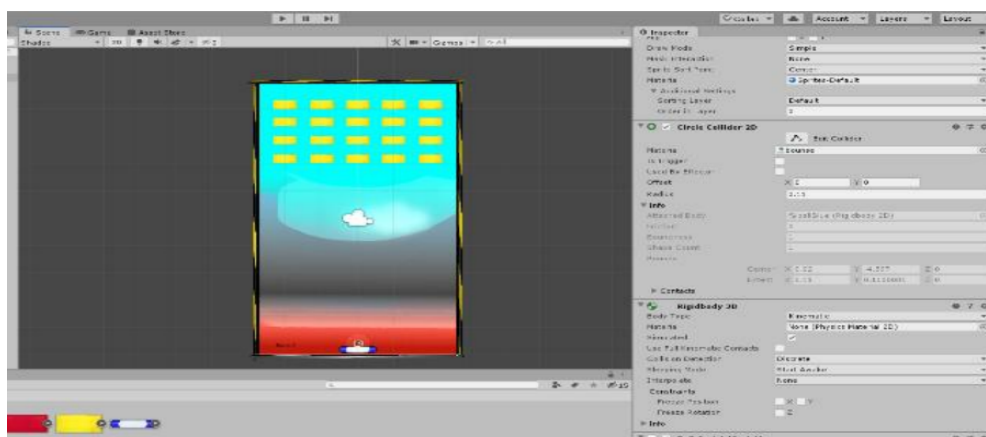


Рис. 3.1. Вікно створення фізичних можливостей

В кожній сцені міститься багато об'єктів (моделі), а пусті об'єкти – це ті об'єкти, що це не мають жодної ігрової моделі. Кожен об'єкт має в собі набори деяких компонентів, з якими будуть взаємодіяти сценарії. Кожен такий елемент має свою назву (Unity надає можливість називати різні об'єкти схожими назвами), вони мають свої теги та шари, на яких вони будуть відображатися. Логіка гри задається за допомогою написання певних сценаріїв до кожного елемента, та додаванню цих сценаріїв до потрібного об'єкта. Кожен сценарій має два початкових методи start та update. В них описуються основні дії та явища, які будуть відбуватися з періодичним викликом.

3.2. Вимоги до технічного та програмного забезпечення

Додаток «SpaceBlaze» має бути написаним на мові C# з використанням ігрової рушію Unity та має працювати на платформі Android версії 5.0 і вище.

Для створення ігрового застосунку впроваджено наступні вимоги до програмного забезпечення:

- Ігровий рушій – Unity;
- Мова програмування – C#;
- Середовище програмування – Visual Studio Code;
- Бібліотека моделей та плагінів – Unity Asset Store;
- Створення двовимірної графіки – Adobe Photoshop;

Вимоги до технічного забезпечення пристрою для розробки:

- Процесор: Intel Core i5-3470;
- Відеокарта: NVIDIA® GeForce® GTX 660 2 ГБ чи подібні;
- Оперативна пам'ять: 8 ГБ;
- Операційна система: Windows 10/11 64-розрядна.

Вимоги до технічного забезпечення мобільного пристрою:

- Процесор: 4-ядерний;

- Оперативна пам'ять: 2 ГБ;
- Місце на диску: 30 МБ;
- Операційна система: Android 5.0.

3.3. Опис програмної реалізації

3.3.1. Робота з графікою

Графіка створюється в редакторі Adobe Photoshop. Для розробки можна використовувати різні фільтри зображення. Для початку потрібно додати на екран спрайт і перейти до редагування зображення. Додамо невелике зображення та 4 точки до нього, цього буде достатньо для обробки колізій. Надалі будемо додавати інші властивості які допоможуть налаштувати коректну поведінку кульки. Після додавання всіх об'єктів, можна перейти до налаштування їх властивостей. Треба розташувати всі елементи по своїх позиціях.

Для того, щоб створити графіку, ми використовуємо графічний редактор Adobe Photoshop. Створюємо всі необхідні графічні об'єкти і додаємо їх в проект. Де вже сам рушій перетворить їх в зразок текстури - спрайт. Також в конструкторі є вбудований редактор спрайтів, в якому можна створювати та редагувати спрайти.

3.3.2. Розробка меню гри

Для того, щоб створити меню в Unity, не знадобиться багато часу, для цього потрібно створити окрему сцену під назвою MainMenu.

Сцена MainMenu містить в собі налаштування камери, головне меню та взаємодію його елементами, такими як відновлення та вихід з гри.

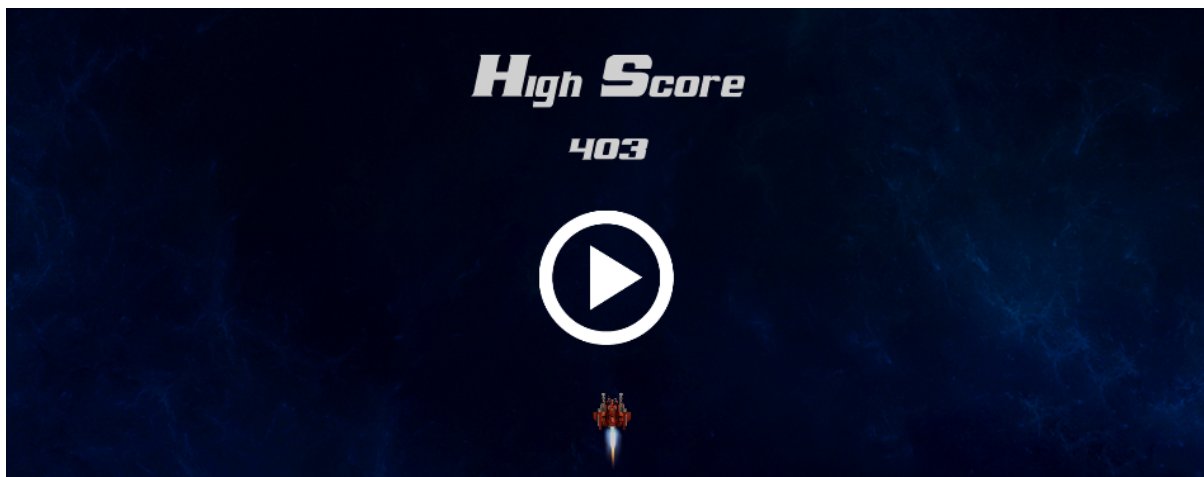


Рис. 3.2. Сцена головного меню

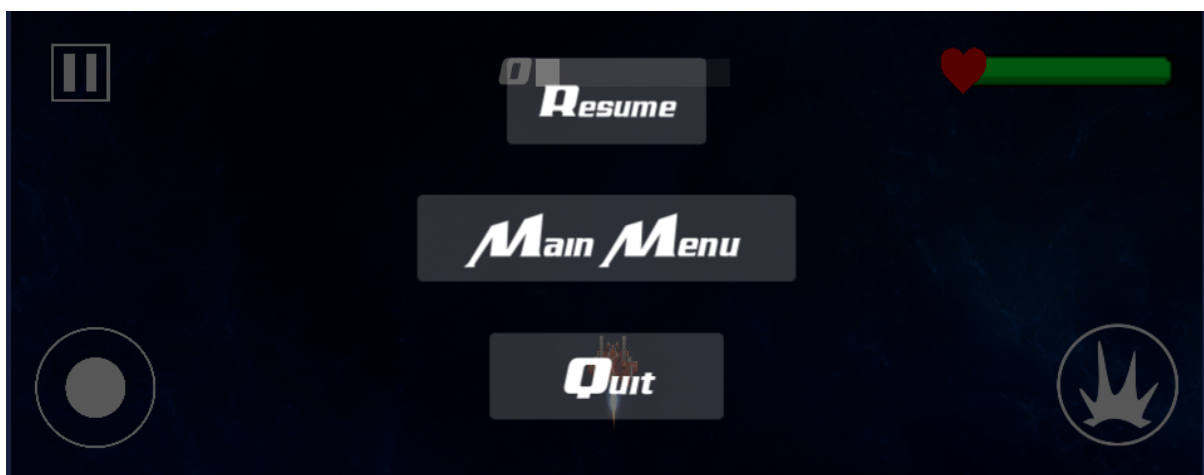


Рис. 3.3. Меню паузи

Сцена MainMenu містить в собі два сценарії: Play Button та Start Game, які відповідають за:

- Початок гри;
- Перехід від сцени гри до паузи;
- Вихід з застосунку;
- Графічні елементи(кнопки, лічильник балів та об'єм здоров'я).

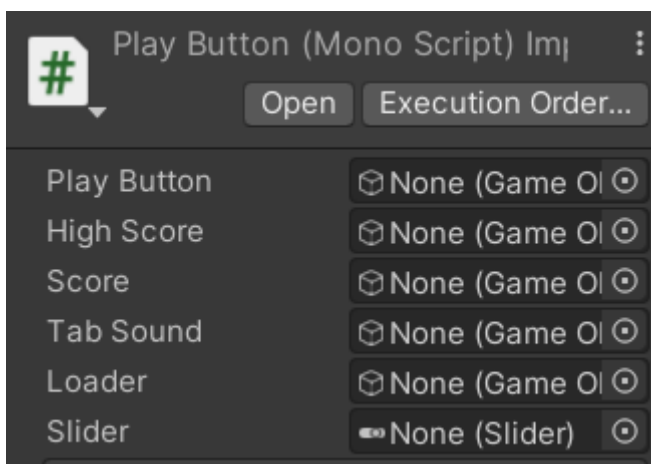


Рис. 3.4. Компоненти Play Button

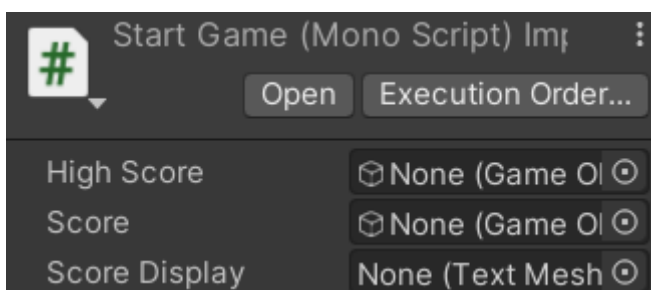


Рис. 3.5. Компоненти Start Game

3.3.3. Інтерфейс та органи керування

На рисунку 3.6 продемонстровано інтерфейс органів керування, а також ігрової статистики (кількість зароблених балів та залишок очок здоров'я гравця).

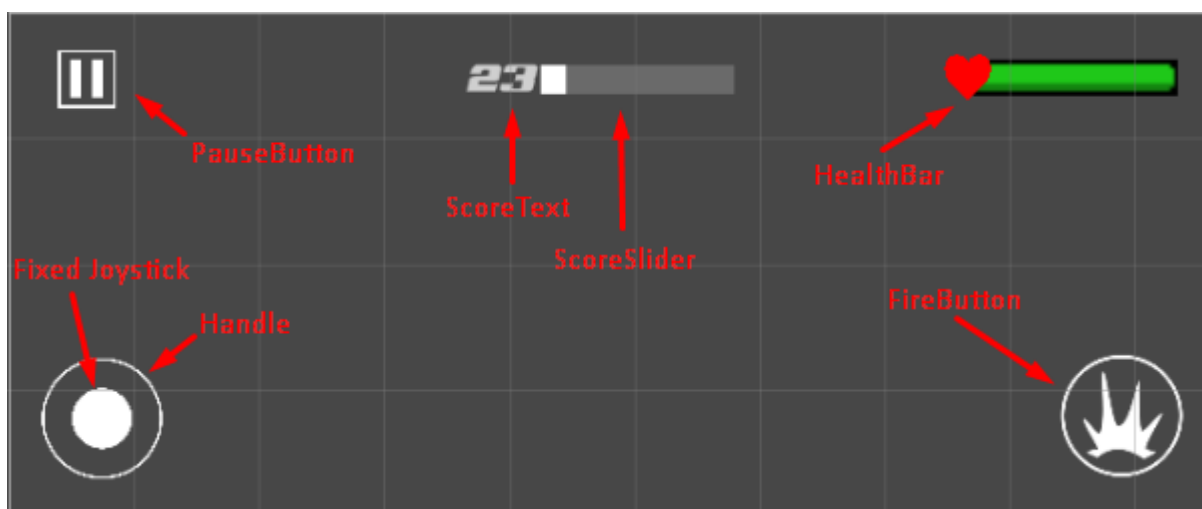


Рис. 3.6. Інтерфейс ігрової сцени

У інтерфейсі ігрової сцени присутні наступні елементи:

- Pause Button - відповідальний за вихід в меню паузи;
- Fixed Joystick та Handle - відповідальні за переміщення гравця сценою;
- FireButton - відповідальний за постріли зі зброї корабля гравця;
- ScoreText та ScoreSlider - відображення кількісного та графічного значення кількості здобутих очок;
- HealthBar - відображення кількості здоров'я гравця.

3.3.4. Розробка Player та PlayerController

Керування гравцем проводиться з допомогою описаних вище елементів керування, які є частиною інтерфейсу. З їх використанням відбувається переміщення літака гравця системою координат по осям X та Y, тобто горизонтально та вертикально.

Сценарій PlayerController також містить в собі систему прогресування, з допомогою якої підвищується складність гри за досягнення певної кількості очок. Рівень складності гри впливає не тільки на хід самої гри, а і на гравця також. Збільшуючи об'єм його здоров'я та покращенням зброї його літака.



Рис. 3.7. Спрайт літака гравця

Сценарій Player містить в собі присвоєння гравцю змінної здоров'я та його максимального значення - 100 одиниць. Він також відповідає за кінець гри у випадку, якщо значення здоров'я гравця буде дорівнювати нулю.

3.3.5. Розробка CameraController

У зв'язку з тим, що наш ігровий проект виконаний у стилі TopDown (вигляд зверху), потрібно налаштувати вид камери під час гри.

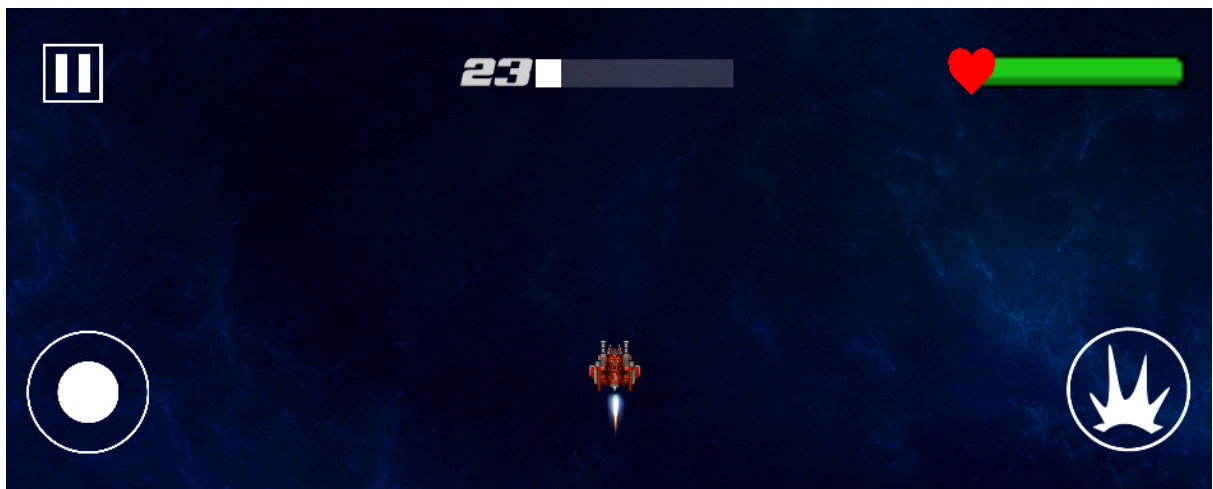


Рис. 3.8. Вигляд ігрової сцени з боку гравця

Під час гри камера буде слідкувати за гравцем, це можливо виконати з допомогою додавання сцени ToBeFollowedByCam.

3.3.6. Розробка Enemy

Enemy - ворог, знищення якого є основною метою гри.

До типу Enemy в даному проекті відносяться пасивні сутності в вигляді астероїдів, які несуть небезпеку гравцю, слідкуючи за ним, та намагаючись врізатись в нього, від чого гравець отримує шкоду.

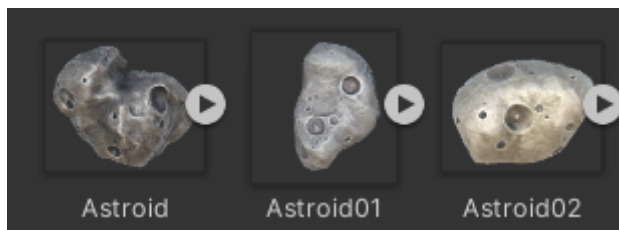


Рис. 3.9. Спрайти астероїдів

Присутні також сутності з активною метою знищення гравця - ворожі літаки, які споряджені зброєю і будуть використовувати її для пострілів по гравцю, в той самий час переслідуючи їх.

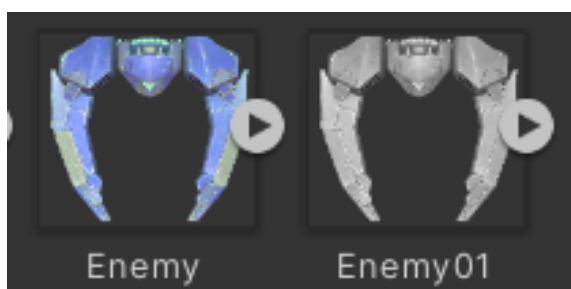


Рис. 3.10. Спрайти ворожих літаків

Для сутностей типу Enemy реалізовано сценарій EnemySpawner, за допомогою якого астероїди та ворожі літаки з'являються в грі випадковим чином. Задля меншого навантаження на фізичний носій та коригування складності гри діють обмеження на загальну кількість одночасного існування ворогів на ігровій сцені та обумовлено часову затримку між їх появою.



Рис. 3.11. Ворожі об'єкти в сцені гри

Ворожі сутності в грі теж, як і сам гравець, залежать від об'єму здоров'я, яке збільшується разом із підвищенням складності гри.

3.3.7. Робота над логікою гри

Логіка гри задається за допомогою написання певних сценаріїв до кожного елемента та додаванню цих сценаріїв до потрібного об'єкта. Кожен сценарій має два стартових методи - старт та апдейт. В них ми будемо записувати основні дії, концепції та певні явища, які будуть відбуватися з періодичним викликом.

В результаті створений сценарій можна додати до багатьох об'єктів. Наприклад, сценарій, який під час кожного виклику методу Update перевіряє скільки об'єктів знищив гравець та змінює кількість очок.

У даному проєкті для створення фізики гри використовуються види поведінки Rigidbody 2D та Capsule Collider 2D. Вони дозволяють переміщатися об'єктам із використанням явищ фізики. Для розуміння того, що собою представляє кожна поведінка, можна скористатися прикладами, що є доступними в конструкторі. Всі приклади, пов'язані з фізикою починаються зі слова Physics. Аби надати об'єктам фізичних властивостей, треба натиснути на спрайт, до якого хочемо додати фізику. Та у вікні Properties натиснути на Add / Edit under Behaviors, і натиснути на плюс, та обрати Physics.

Для того, щоб гравець міг керувати своїм кораблем, додаємо набір сценаріїв із назвою Joystick з обробником, який буде реагувати на натискання.

За допомогою них буде відбуватися поєднання подій з спрайтами. Робота з тим чи іншим об'єктом полягає побудові певної логіки їх використання на сцену та настроюванні його властивостей. Використовуючи ці об'єкти, було додано до сцени бекграунд, контролер для натискання і обробки подій, гравця та платформу. Таким чином ми можемо ділі задавати нашу логіку гри.

3.3.8. Робота з фізикою гри

Всі спрайти, що були створені, мають полігони, які будуть відповідати границям зображення. Треба розуміти, що їх завжди можна змінити за потреби у цьому. Основним елементом буде обробка колізій. Вона буде відбуватися, коли гравець чи снаряд буде зіштовхуватися з іншим об'єктом. Тобто їх координати будуть збігатися.

Колайдери можуть бути додані до об'єкта без компонента жорсткого тіла для створення підлоги, стін та інших нерухомих елементів сцени. Вони відносяться до статичних колайдерів. Взагалі, нам не слід переставляти статичні колізії, змінюючи положення Transform, оскільки це сильно вплине на працездатність фізичного двигуна. Колайдери на об'єкті, що мають жорстке тіло, відомі як динамічні колайдери. Статичні колайдери можуть взаємодіяти з динамічними колайдерами, але оскільки у них немає жорсткого тіла, вони не рухатимуться у відповідь на зіткнення. Для початку треба додати компонент Collider до об'єкту нашої гри, Collider буде означати тип обраного рушія, ми будемо використовувати Capsule Collider 2D. Даний колайдер допоможе у генерації зіткнень та наданні ефекту влучання наших набоїв по ворогу.

Однією з найважливіших деталей в нашому проекті є налаштування колізії є (Layers) в них можна назначити для кожного потрібного об'єкта свій рівень доступу. Якщо об'єкти будуть знаходитись на різних рівнях доступу, вони не можуть взаємодіяти між собою. Тому ми не будемо залишати увімкненою взаємодію літака гравця з ворогами та ворожими снарядами.

Висновок до розділу 3

Під час розробки кваліфікаційного проекту було затверджено засоби розробки, які будуть використовуватись під час розробки проекту, вимоги до технічного та програмного забезпечення, а також описано процес реалізації ігрового застосунку за допомогою засобів розробки.

РОЗДІЛ 4

ТЕСТУВАННЯ

4.1. Функціональне тестування

Функціональне тестування – один із видів тестування, спрямованого на перевірку відповідностей функціональних вимог ПЗ його реальним характеристикам. Основним завданням функціонального тестування є підтвердження того, що програмний продукт, який розробляється, володіє усім необхідним функціоналом.

По результатам функціонального тестування було створено таблицю 3.1 з результатами.

Таблиця 3.1

Результати функціонального тестування

№	Назва тесту	Очікуваний результат	Отриманий результат	Тест пройдений?
1	Відображення головного меню гри	На екрані відображається головне меню гри з кнопкою «Play»	На екрані відобразилось головне меню гри з кнопкою «Play»	+
2	Робота кнопки «Play»	При натиску на кнопку «Play» очікуємо перехід до сцени гри	При натиску на кнопку «Play» перейшло до наступної сцени	+
3	Робота кнопки «Quit Game»	При натисканні кнопки «Quit Game» очікуємо вихід з гри	При натиску на кнопку «Quit Game» вийшли з гри	+
4	Відображення меню паузи	На екрані відображається головне меню гри з вікнами вибору «Resume» та «Quit Game»	На екрані відобразилось головне меню гри з вікнами вибору «Resume» та «Quit Game»	+

5	Робота кнопки «START»	При натиску на кнопку «Play» очікуємо завантаження рівня гри	При натиску на кнопку «Play» завантажується рівень гри	+
6	Робота SimpleInput	При натиску на стрілки керування виконується переміщення гравця	При натиску на стрілки керування виконується переміщення гравця	+
7	Робота інтерфейсу сцени гри	Відображається кількість зароблених очок, об'єм здоров'я гравця.	Відображається кількість зароблених очок, об'єм здоров'я гравця.	+
8	Робота GameManager	При смерті гравця гра закінчується	При смерті гравця гра закінчується	+

4.2. Usability Test

Usability Test – це оцінка продукту чи послуги шляхом тестування їх із представниками користувачів. Як правило, під час тесту учасники намагаються виконати типові завдання, а спостерігачі дивляться, слухають і роблять нотатки. Мета полягає в тому, щоб виявити будь-які проблеми з зручністю використання, зібрати якісні та кількісні дані та визначити задоволеність учасника продуктом.

Шляхом такого тестування визначається зручність, зрозумілість та привабливість продукту для користувача. Виконується за допомогою залучення користувачів з метою проходження тестів, з метою отримання надалі інформації та висновків від випробувачів.

Учасникам тестування необхідно виконати такі задачі: Почати гру; Зупинити гру; Продовжити гру; Заробити бали; Закінчити гру.

Висновок до розділу 4

Всі задачі були виконані тестувальниками успішно та без проблем. Респонденти відмітили простоту керування, хороше оформлення та складність проходження. Usability test пройдено успішно.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було вивчено сучасні інформаційні технології навчання, їх вплив на ігрову індустрію. Показані можливості ігрового рушія Unity у розробці ігор.

Додаток Unity, що є професійним ігровим рушієм, використовується у створенні відеоігор для різних платформ. Це інструмент, яким щодня користуються досвідчені розробники, а також один із найдоступніших інструментів для новачків.

При написанні коду використано мову C#. Ця мова дозволяє легко увійти до розробки гри. Мова C# актуальна на сьогоднішній день, тому що дозволяє сконцентруватися на проектуванні програм, позбавляючи розробника безлічі рутинних операцій.

Для створення 2D графіки застосовано програмне забезпечення Adobe Photoshop, яке головним чином призначене для створення растрової графіки. Особливості Adobe Photoshop полягають у багатому обсязі інструментів для операції створення і обробки зображень, високій якості обробки графічних зображень, зручності й простоті в експлуатації, широких можливостях до автоматизації обробки растрових зображень

Таким чином, можна зробити висновок, що використовуючи можливості вище названих програм, можна створювати ігрові додатки розважального характеру для різних платформ.

У ході виконання кваліфікаційної роботи бакалавра було реалізовано ігровий застосунок в жанрі shoot'em up під мобільну систему Android на рушію Unity.

Для вирішення поставленої мети було виконано такі завдання:

- Проведено аналіз актуальності теми;
- Виконано аналіз існуючих аналогів;

- Виконано огляд актуальних програмних середовищ для реалізації додатки;
- Спроектовано ігровий додаток;
- Реалізовано ігровий додаток;
- Проведено тестування ігрового застосунку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація по Android [Електронний ресурс]: Collider. – Режим доступу: <https://developer.android.com/docs>.
2. Документація по Unity [Електронний ресурс]: Unity 3D. – Режим доступу: <https://unity.com/ru>.
3. Колізії та їх застосування [Електронний ресурс]: Операційні системи. – Режим доступу: https://howlingpixel.com/iuk/Ambient_occlusion.
4. Операційні Системи [Електронний ресурс]: Операційні системи. – Режим доступу: https://wiki.cuspu.edu.ua/index.php/Операційні_системи.
5. Android operation information [Електронний ресурс]: Android O S. – Режим доступу: <https://www.gadgetgyani.com/wp-content/uploads/2016/03/android-features-pdf.pdf>.
6. DevGam сравнение игровых движков URL: <http://devgam.com/sravnienie-igrovyyh-dvizhkov-kakoj-vybrat>
7. List of game engines and frameworks URL: https://en.wikipedia.org/wiki/List_of_game_engines
8. List of game engines URL: <https://80.lv/articles/the-most-comprehensive-listof-game-engines/>
9. Unity store URL: https://store.unity.com/ru?_ga=2.200322542.1765686190.1560285797-130111505.1547668569
10. Руководство Unity URL: <https://docs.unity3d.com/ru/current/Manual/index.html>
11. Форум Unity3d URL: <http://unity3d.ru>
12. Відео уроки по Unity3D URL: <https://www.youtube.com/watch?v=m0sKo81DZME>
13. Хокинг Д. Unity in Action: Multiplatform Game Development in C# with Unity 5 : Manning Publications Company. 2015 – 352с
14. Руководство по программированию на C# URL: <https://docs.microsoft.com/ru-ru/dotnet/articles/csharp/programmingguide/index>