

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра комп'ютерних наук

КУРСОВА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Технології віртуальної та доповненої реальності»
на тему:

**«Розробка прототипу гри в AR " LeukoDefenders ".
3D моделі та левел дизайн»**

студента III курсу групи КН-21-1
спеціальності 122 «Комп'ютерні науки»
Марчук Назар Анатолійович
(прізвище, ім'я та по-батькові)

Керівник Граф М.С.

Дата захисту: «13» червня 2024 р.
Національна шкала _____
Кількість балів: _____
Оцінка: ECTS _____

Члени комісії

_____	<u>Лисенко М. С.</u>
(підпис)	(прізвище та ініціали)
_____	<u>Квасніков В. П.</u>
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерних наук
Освітній рівень: бакалавр
Спеціальність 122 «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»
Зав. кафедри
_____ М.С. Граф
“ _____ ” _____ 2024р.

ЗАВДАННЯ
НА КУРСОВУ РОБОТУ СТУДЕНТУ

1. Тема роботи: Розробка прототипу гри в AR «LeukoDefenders»

керівник курсового проекту: Граф М. С.
2. Строк подання студентом: “ 13 ” _____ червня 2024р.
3. Вхідні дані до роботи: Розробка прототипу гри в AR «LeukoDefenders»
4. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
 1. Аналіз проблематики, методів та засобів вирішення задачі; _____
 2. Проектування та розробка програмного забезпечення _____
 3. Опис роботи з ігровим додатком та його тестування _____
5. Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)
 - 1) Презентація
 - 2) Посилання на git hub: <https://github.com/Kn211mna/AR>
6. Консультанти розділів.

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3	Граф М.С., зав. кафедри КН		

7. Дата видачі завдання “ 13 ” червня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсового проекту	Строк виконання етапів проекту	Примітки
1	Постановка задачі	12.02.2024	виконано
2	Пошук, огляд та аналіз аналогічних розробок	03.03.2024	виконано
3	Формулювання технічного завдання	05.03.2024	виконано
4	Опрацювання літературних джерел	10.04.2024	виконано
5	Проектування структури	14.04.2024	виконано
6	Написання програмного коду	17.04.2024	виконано
7	Налагодження	20.05.2024	виконано
8	Написання пояснювальної записки	23.05.2024	виконано
9	Захист	13.06.2024	

Студент

(підпис)

Марчук Н.А.

(прізвище та ініціали)

Керівник проекту

(підпис)

Граф М.С.

(прізвище та ініціали)

РЕФЕРАТ

Завданням на курсову роботу було створення прототипу гри «LeukoDefenders».

Пояснювальна записка до курсової роботи на тему «Розробка прототипу гри в AR «LeukoDefenders»» складається з вступу, трьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 32 сторінках друкованого тексту.

Пояснювальна записка має 8 сторінок додатків.

Список використаних джерел містить 9 найменувань і займає 1 сторінку. В роботі наведено 20 рисунків. Загальний обсяг роботи – 40 сторінок.

Ключові слова: UNITY, ДОДАТОК, ГРА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, АЛГОРИТМ, AR.

					«Житомирська політехніка».24.122.09.000 – ПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Марчук Н.А.			Розробка програми "Графічний редактор"				Літ.		Арк.	Аркуші
Керівник		Граф М. С.									4	71
Реценз.									ФІКТ, гр. КН-21-1			
Н. Контр.												
Затверд.												

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ	7
1.1 Аналіз задачі, існуючого програмного продукту.....	7
1.2 Аналіз задачі, засобів та методів її вирішення.....	7
Висновки до першого розділу.....	9
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	12
2.1 Проектування загального алгоритму роботи програми	12
2.2 Розробка функціональних алгоритмів роботи програми	14
2.3 Збірка проекту	14
Висновки до другого розділу	18
РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ	19
3.1 Опис роботи з програмним додатком	19
3.2 Тестування роботи програмного забезпечення.....	20
Висновки до третього розділу.....	21
ВИСНОВКИ.....	22
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	23
ДОДАТКИ.....	24
Додаток А.....	25

ВСТУП

Актуальність теми:

Комп'ютерні ігри займають важливе місце в сучасній культурі та розвагах, надаючи користувачам можливість занурюватися в нові світи та отримувати унікальні враження. Окрім розважальної функції, вони суттєво впливають на економіку, створюючи робочі місця та сприяючи зростанню галузі через продажі та інновації. Високоякісні ігри стимулюють технологічний прогрес, вимагаючи постійного вдосконалення апаратного та програмного забезпечення, а також сприяють розвитку технологій, які можуть використовуватися в інших галузях промисловості. Індустрія комп'ютерних ігор славиться своєю інноваційністю та творчим підходом. Розробники постійно вдосконалюють геймплей та впроваджують нові ідеї, що сприяє розвитку індустрії. AR у відеоіграх стирає межу між реальністю та віртуальним світом, дозволяючи гравцям взаємодіяти з цифровими об'єктами у фізичному просторі. Це додає новий рівень занурення та інтерактивності, розширюючи горизонти як для розробників, так і для користувачів. Завдяки AR комп'ютерні ігри стають більш захоплюючими та функціональними, відкриваючи нові шляхи для творчості, навчання та соціальної взаємодії.

Мета курсової роботи:

Мета курсової роботи полягає у вивченні та практичній реалізації процесу розробки комп'ютерної гри з використанням технологій AR. Робота фокусується на дослідженні основних аспектів, таких як використання сучасних мов програмування, реалізація головних геймплей-механік та систем взаємодії об'єктів, а також впровадження доповненої реальності.

Об'єкт і предмет дослідження:

Об'єктом дослідження є процес розробки комп'ютерної гри з технологіями AR, зокрема архітектура програмного забезпечення та ключові аспекти, що впливають на успішність і якість розробленої гри. Предметом дослідження є реалізація геймплей-механік та систем взаємодії об'єктів у контексті доповненої реальності.

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1 Аналіз задачі, існуючого програмного продукту

Проаналізуємо існуюче програмне забезпечення, для аналізу були взяті наступні програмні додатки:

- AR Invaders
- The Walking Dead: Our World

AR Invaders (рис 1.1)

мобільна гра з доповненою реальністю, розроблена невеликою інді-студією. Гра була випущена в грудні 2010 року для пристроїв на базі iOS та Android. У грі використовується камера пристрою для інтеграції віртуальних об'єктів у реальний світ через доповнену реальність.

Геймплей та механіки. Гравець керуючи захисником за допомогою прицілювання та кнопки “Shot” стріляє по ворогам, що стають швидшими з кожним рівнем та рухаються по різним траєкторіях. Гравець може рухатися в реальному просторі, щоб краще прицілюватися. Індикатор здоров'я показує рівень життя, який зменшується при влучаннях ворогів. Меню дозволяє змінити зброю, налаштувати гру і робити паузу.

Економічний вплив та популярність. AR Invaders не отримала широку популярність але демонструє потенціал розвитку AR-шутерів із простим, але захопливим геймплеем. Внутрішньоігрові покупки відсутні, що робить гру привабливою для користувачів, які шукають чистий AR-досвід без додаткових затрат. Вона також сприяла розвитку технологій доповненої реальності та показала її можливості для аудиторії.

Оцінки та відгуки. Гра отримала загалом позитивні відгуки від критиків та гравців, зокрема, за інноваційний геймплей та здатність залучати людей до фізичної активності. В Play Market для iOS та Android отримали високі оцінки.

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		7



Рис 1.1 AR Invaders

The Walking Dead: Our World (рис 1.2)

цифрова мобільна гра з доповненою реальністю, розроблена Next Games для платформ iOS та Android. Гра була випущена в липні 2018 року і базується на популярному серіалі AMC The Walking Dead, пропонуючи гравцям боротися з зомбі у реальному світі через AR-технології.

Геймплей та механіки. The Walking Dead: Our World пропонує гравцям геолокацію, дозволяючи гравцям досліджувати навколишній світ, знаходити ходячих, рятувати виживших та боротися з рейдерами. Гравці можуть збирати легендарних персонажів, зброю та предмети з серіалу, а також брати участь у сюжетних місіях та кооперативних викликах.

Економічний вплив та популярність. The Walking Dead отримала широку популярність серед фанатів The Walking Dead. Гра стабільно підтримується розробниками, які регулярно випускають оновлення, додаючи

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

нові персонажі, події та місії, що підтримувало інтерес гравців та забезпечувало стабільний потік доходів через внутрішньоігрові покупки.

Оцінки та відгуки. Гра отримала загалом позитивні відгуки від критиків та гравців. На Metacritic вона була високо оцінена за інноваційний підхід до AR-геймплею та інтеграцію з реальним світом.



Рис 1.2 The Walking Dead: Our World

1.2 Аналіз задачі, засобів та методів її вирішення

Гра, яку я розробляю, являє собою інтерактивний мобільний застосунок із доповненою реальністю. Основна мета гри — знищувати віруси, а також виживати потрібний час не даючи вірусам поширитися.

Взаємодія через доповнену реальність за допомогою камери пристрою дозволяє краще прицілюватися. Це додає глибини геймплею і дає можливість розробляти різноманітні тактики для перемоги. Гравець отримує роль дослідника, що контролює імунну систему та допомагає знищити шкідливі бактерії, перш ніж вони завдадуть шкоди організму.

Механіка прицілювання та атаки у доповненій реальності дозволяє гравцеві взаємодіяти з реальним світом через камеру свого пристрою. Використання AR Foundation забезпечує розміщення об'єктів на площинах та

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

інтеграцію віртуальних вірусів у реальне середовище. Гравець знищує шкідливі віруси, що поширюються в просторі і мішають пройти рівень та захистити імунну систему. Цей процес включає рівні складності, додаючи елемент прогресу.

Для розробки гри було обрано Unity, потужну та популярну інтегровану середу розробки (IDE) для створення ігор та інших інтерактивних додатків. Unity підтримує 2D і 3D графіку, що дозволяє створювати візуально привабливі ігри з багатим функціоналом. Вбудовані інструменти для роботи з фізикою, анімаціями та звуком спрощують процес розробки.

Середовище для написання програмного коду було обрано Visual Studio. Visual Studio є однією з найпопулярніших інтегрованих середовищ розробки (IDE), особливо для роботи з мовою програмування C#. Visual Studio надає розширені можливості для налагодження коду, інтеграцію з системами контролю версій та безліч інструментів для підвищення продуктивності розробки.

Використання AR-Foundation дозволяє інтегрувати віртуальні об'єкти у реальний світ, що забезпечує новий рівень взаємодії та занурення.

Висновки до першого розділу

В першому розділі було проведено аналіз задачі, а також існуючих програмних продуктів, таких як AR Invaders та The Walking Dead: Our WorldDuel Links. Досліджено особливості їхніх механік, зокрема інтеграцію доповненої реальності, геймплейні механізми та економічний вплив. Аналіз показав, що ці елементи забезпечують глибокий геймплей та взаємодію з реальним світом.

Для розробки моєї гри було обрано Unity та Visual Studio, які забезпечують потужні інструменти для створення ігор з підтримкою 2D і 3D графіки, а також ефективного програмування на мові C#. Використання AR-Foundation для доповненої реальності дозволяє створити інтерактивний та захоплюючий ігровий досвід, інтегруючи віртуальні об'єкти у реальний світ.

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

Таким чином, проведений аналіз підтвердив доцільність вибору технологій та підходів для розробки гри, яка поєднує в собі бої та доповнену реальність, що забезпечує високий рівень занурення та взаємодії гравців.

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Проектування загального алгоритму роботи програми та дизайн з моделюванням.

Маючи достатній об'єм теоретичної інформації відносно потрібної нам галузі, я можу перейти до етапу проектування програмного забезпечення. Потрібно скласти загальну схему роботи програми (Рис. 2.1 - 2.2), що надасть можливість оцінити алгоритм роботи з цією програмою користувачем.

Загальний алгоритм роботи програми

Після запуску програми вона відразу готова до прийняття команд користувача. В меню користувач може виконувати наступні дії:

- Вибрати "Continue" для продовження гри.
- Вибрати "New Game" для нової гри.
- Вибрати "Statistic" для перегляду статистики.
- Вибрати "Exit" для виходу з гри.

При натисканні кнопки "Statistic" користувач має можливість:

- Переглянути кількості рівнів
- Переглянути статистику проходження рівня.

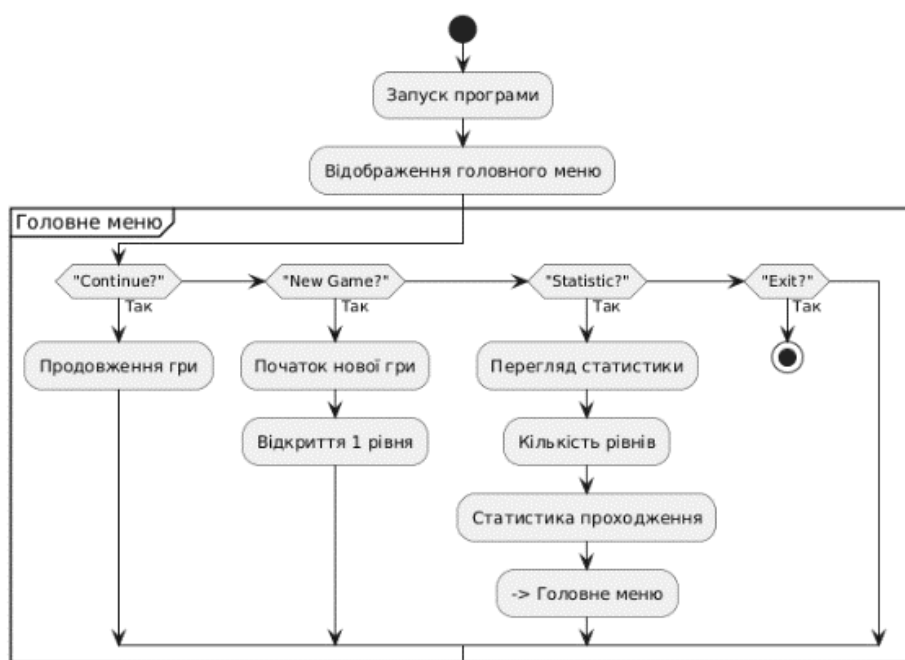


Рис. 2.1 Загальна схема роботи програми

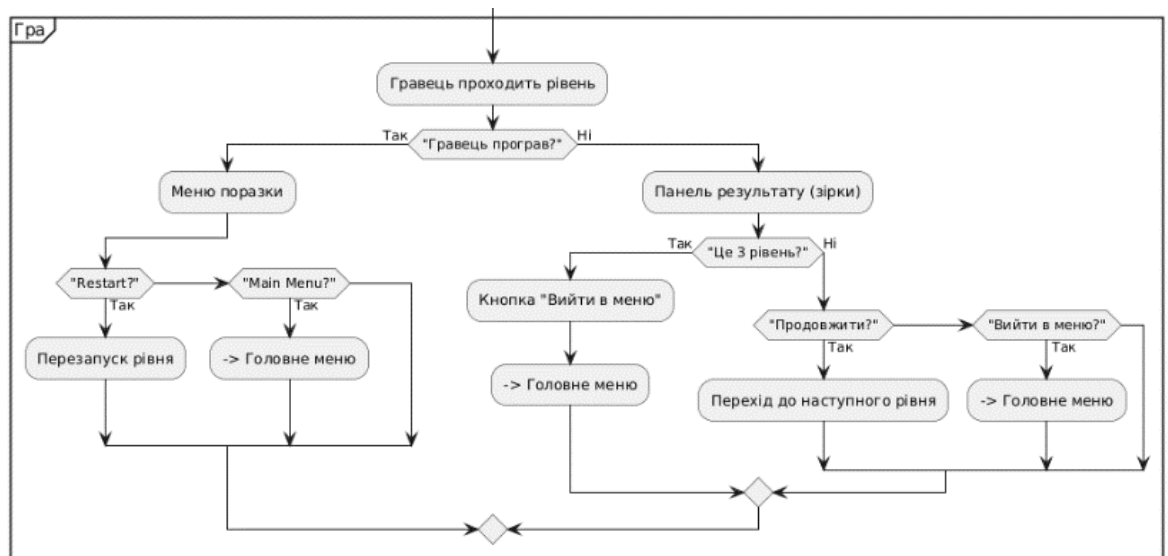


Рис. 2.2 Продовження схеми роботи програми

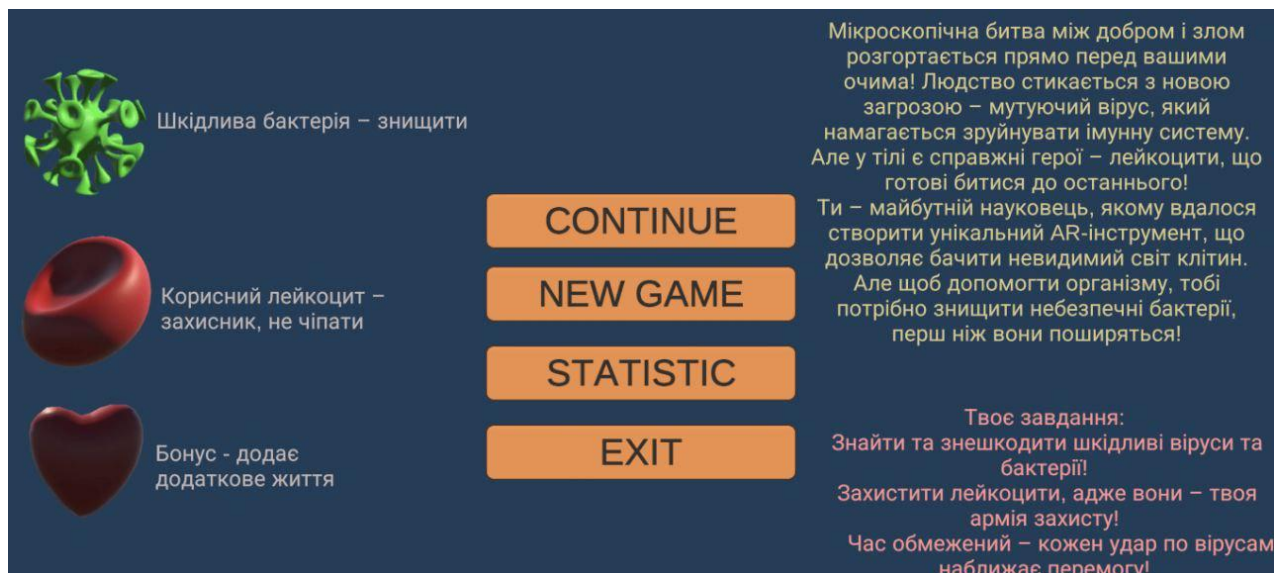


Рис. 2.3 Головне меню програми

Розробка 3D моделей(рис2.4)

Розробка 3D моделей є важливим етапом створення гри, оскільки вони надають візуальну привабливість та забезпечують глибокий ігровий досвід.

Дизайн UI гри (рис 2.5)

Як і у випадку зі створіннями, моделі оточення оптимізовані для забезпечення плавної роботи на різних пристроях, враховуючи обмеження продуктивності мобільних платформ.

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

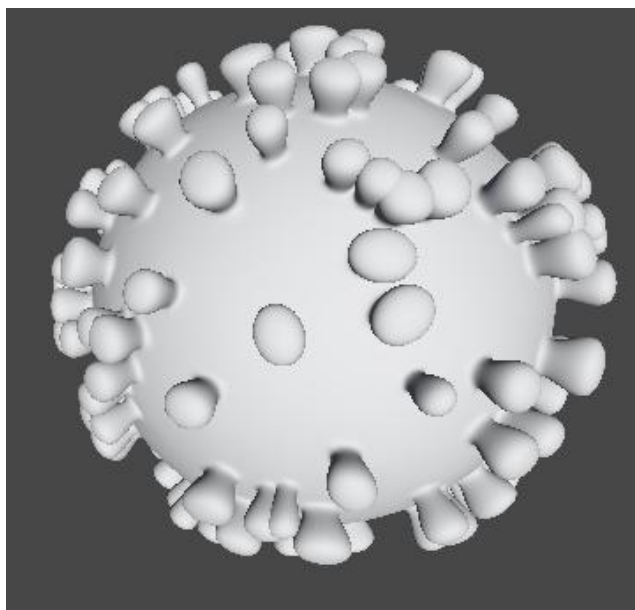


Рис 2.4 Бактерія

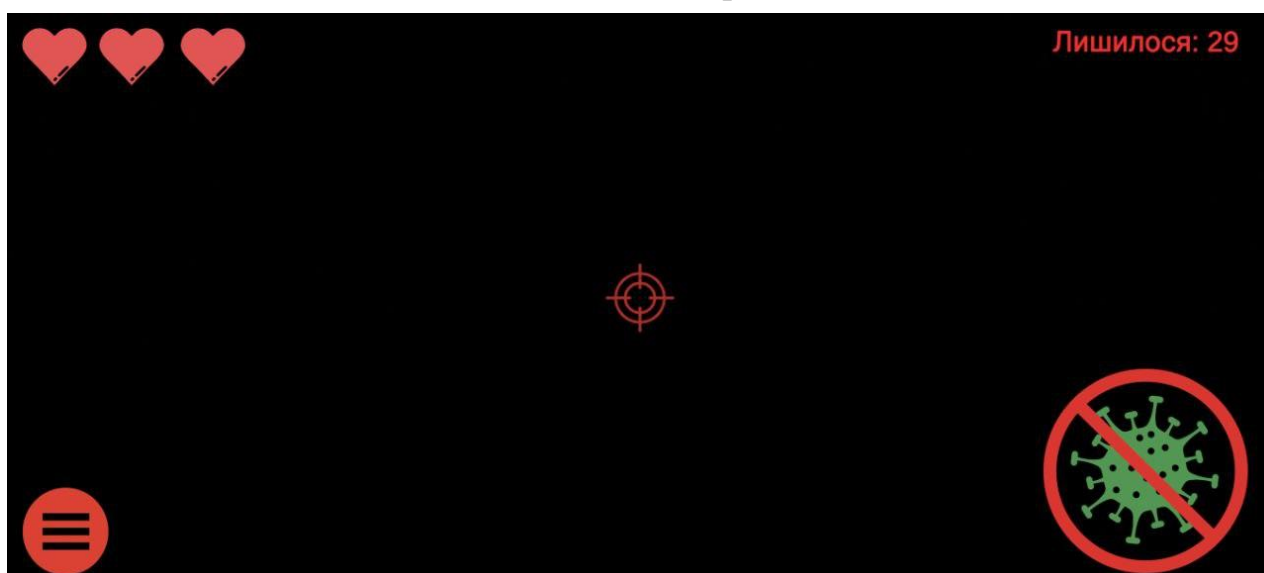


Рис 2.5 Дизайн ігрової UI

2.2 Розробка функціональних алгоритмів роботи програми

Розробка функціональних алгоритмів є ключовим етапом у процесі створення програмного забезпечення. В цьому розділі буде розглянуто алгоритми, які забезпечують основну функціональність гри, зокрема бої між гравцем та вірусами та збереження статистики гравців. Алгоритми побудовані на основі коду, наданого в попередньому розділі, і описують процеси роботи з вірусами (Fighter) та спаунер ігрових об'єктів.

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2.1 Алгоритм запуску та ініціалізації гри

1. Ініціалізація даних прогресу:

- На старті гри створюється або ініціалізується об'єкт LevelProgressData, який містить масив зірок для кожного рівня.
- Якщо дані вже існують у PlayerPrefs, вони можуть бути завантажені (опціонально, якщо додати таку логіку).

2. Відображення головного меню:

- Відображається головне меню (MainMenuManager), де гравець може обрати "Нова гра", "Продовжити" або переглянути статистику.

2.2.2 Алгоритм початку нової гри та продовження

Створення та ініціалізація персонажів (Fighter) відбувається через компонент FighterSpawner.

1. Нова гра:

- При натисканні "Нова гра" викликається метод OnStartGame у MainMenuManager.
- Всі дані прогресу (зірки за рівні) скидаються через levelProgressData.Init(totalLevels) та видаляються з PlayerPrefs (ResetProgressInPlayerPrefs).
- Завантажується перший рівень.

2. Продовження гри:

- При натисканні "Продовжити" визначається, який рівень ще не пройдено (перша позиція з 0 зірок).
- Завантажується відповідний рівень.

2.2.3 Алгоритм проходження рівня

1. Початок рівня:

- На старті рівня ініціалізується інтерфейс (GameUIManager), встановлюється кількість життів, таймер, скидаються бонуси.

2. Ігровий процес:

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

- Об'єкти (віруси, здорові кулі, бонуси) спавняються на знайдених AR-площинах (SpawnScript).
- Гравець стріляє по об'єктах через ShootScript.
- При попаданні у вірус або здорову кулю викликаються відповідні методи (OnPopped у BallonScript).

3. Завершення рівня:

- Якщо гравець втрачає всі життя — показується панель поразки.
- Якщо час вичерпано або всі цілі знищено — підраховуються зірки (CalculateStars), викликається CompleteLevel, зберігається прогрес у LevelProgressData та PlayerPrefs.

2.2.4 Алгоритм збереження та завантаження прогресу

1. Збереження прогресу:

- Після завершення рівня у методі SetStars класу LevelProgressData оновлюється масив зірок.
- Для кожного рівня зірки зберігаються у PlayerPrefs (PlayerPrefs.SetInt("LevelStars_{i}", stars)).

2. Завантаження даних:

- При старті гри (опціонально, якщо додати) можна зчитувати зірки з PlayerPrefs і ініціалізувати масив starsPerLevel у LevelProgressData.
- Обробка отриманих даних та їх використання у грі.

3. Скидання прогресу:

- При виборі "Нова гра" всі ключі зірок видаляються з PlayerPrefs (PlayerPrefs.DeleteKey("LevelStars_{i}")).

2.3 Збірка проекту

Після завершення проектування можна перейти до розробки програмного забезпечення. Етап проектування спростив розробку додатку та дав поштовх для розбиття програми на відповідні класи для розробки програми (Рис. 2.6).

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

Спочатку було створено 3D проект та додано все потрібне для створення гри. Структура матиме наступний вигляд.

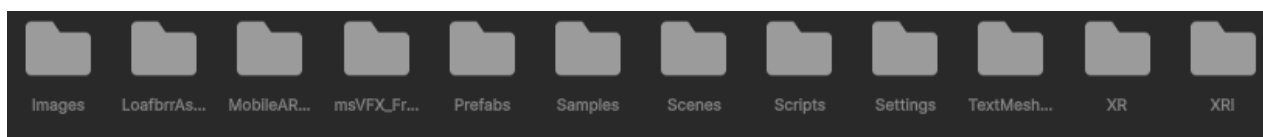


Рис. 2.6. Структура проекту

Наступним кроком було додано всі сцени та розставлено у правильному порядку (Рис. 2.7)

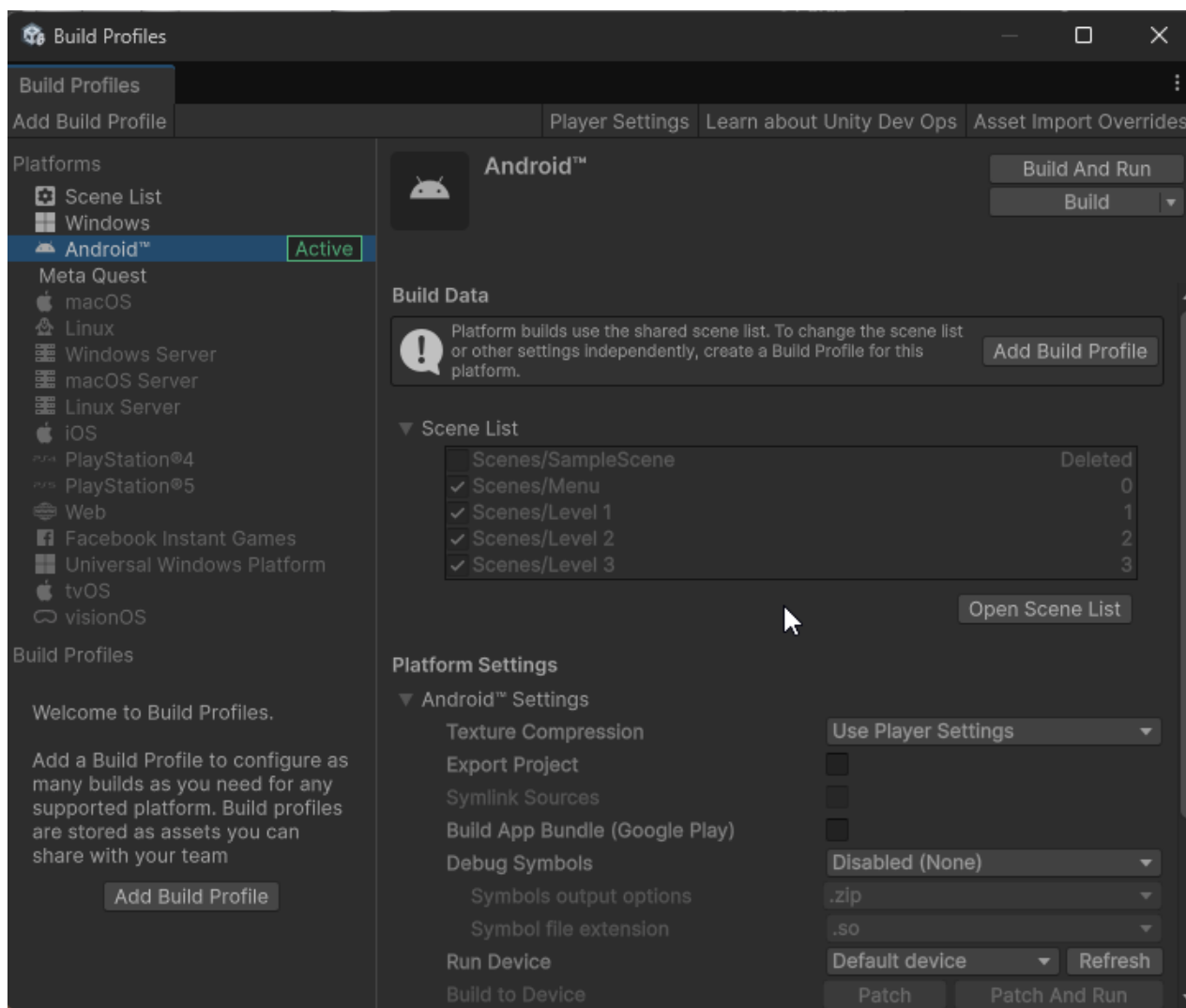


Рис. 2.7. Порядок переходу по сценам

Збереження прогресу: Використання ScriptableObject для збереження прогресу користувачів локально.

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

Висновки до другого розділу

Побудова функціональних алгоритмів забезпечує ясність і логічність у виконанні ігрових процесів. Вони описують, як повинна працювати програма та взаємодіяти з користувачем, а також як обробляються та зберігаються дані. Використовуючи ці алгоритми, можна покращувати та оптимізувати гру, щоб досягти максимальної продуктивності та відповідати очікуванням гравців.

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

3.1 Опис роботи з програмним додатком

1 Опис роботи з програмним додатком

Запускаючи гру, з'являється перша сцена, що складається з меню (Рис. 3.1).



Рис. 3.1. Головний екран

Натиснувши кнопку реєстрація ми потрапляємо до під меню цієї кнопки, де є можливість ввести електронну пошту, нікнейм та пароль (Рис. 3.2).

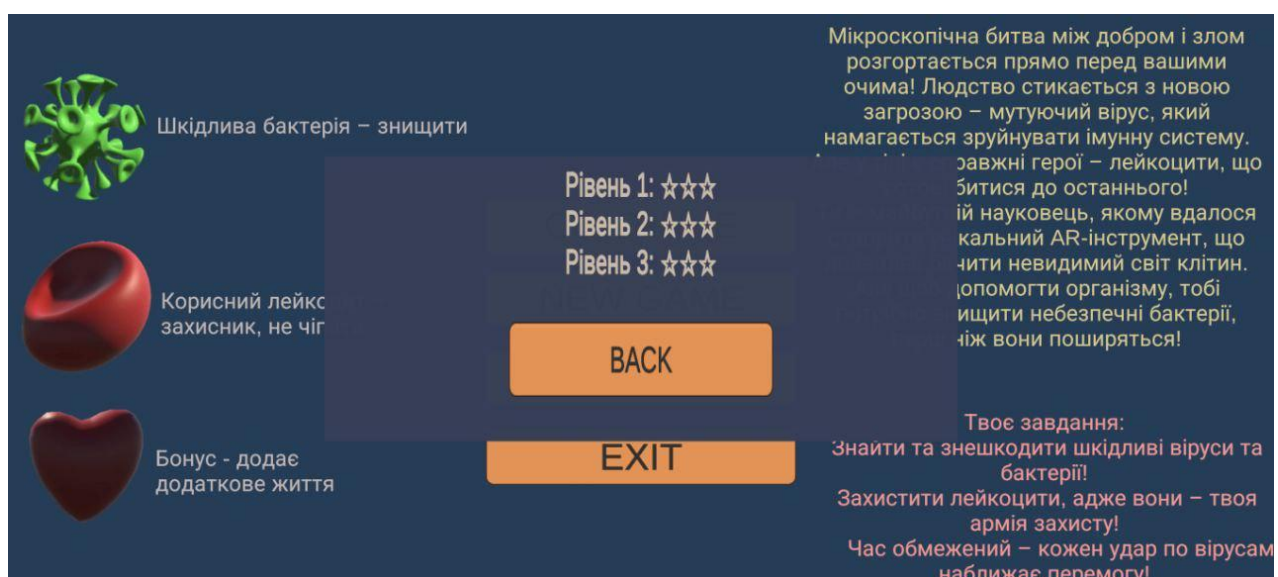


Рис. 3.2. Перехід по кнопці статистика

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ док.ум.	Підпис	Дата		19

Також в головному меню доступні кнопки нова гра та продовжити. При натисканні кнопки нова гра стартує, в якій відбуваються бої (Рис. 3.3).

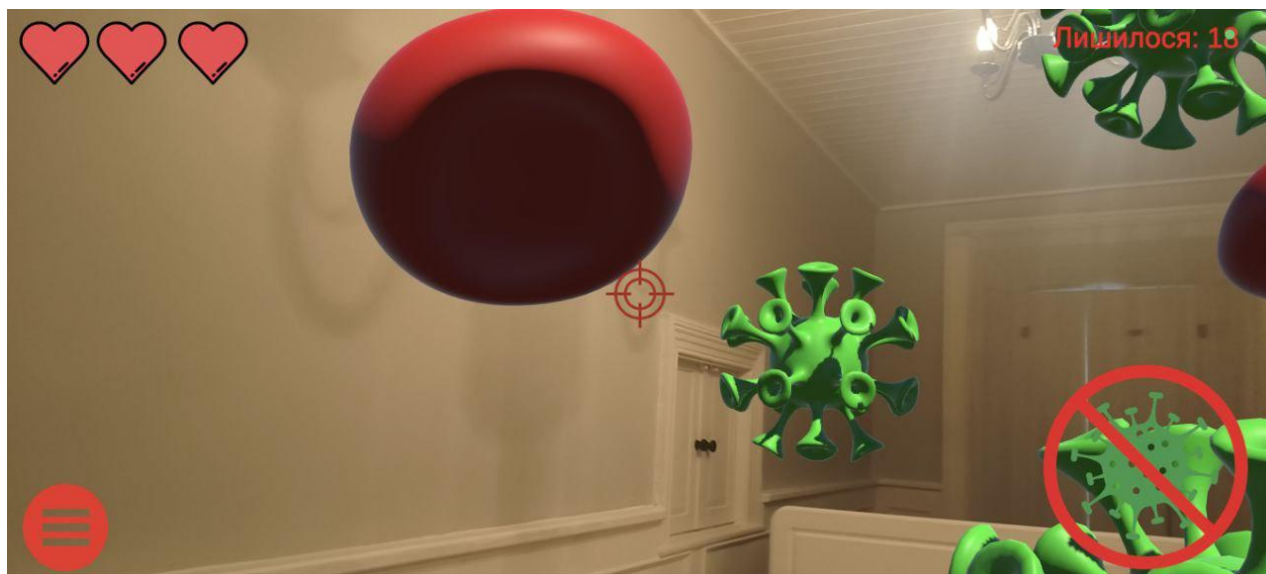


Рис. 3.3. Бої

Головна мета гри – це знищити віруси, та проходження lvls.

3.2 Тестування роботи програмного забезпечення

Тестування гри є критично важливим етапом у процесі розробки, забезпечуючи валідацію як апаратної, так і програмної складових. Воно включає виявлення та виправлення помилок, а також оптимізацію ігрових механік для створення найкращих вражень для гравців. Тестування не лише допомагає забезпечити якість кінцевого продукту, але й дозволяє реалізувати ідеї розробників у реальних умовах.

Під час тестування інтерфейсу користувача було виявлено високу функціональність і зручність взаємодії для гравців. UI виявився інтуїтивно зрозумілим, з чіткою логікою розташування елементів. Серйозні помилки чи конфлікти у роботі інтерфейсу не були виявлені.

Загалом, тестування UI показало, що інтерфейс готовий до використання в реальних умовах і відповідає основним вимогам щодо зручності та ефективності.

Під час тестування процесу прицілювання та атаки було підтверджено ефективність цієї механіки. Гравці могли успішно виконувати ці дії без значних

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

затримок або помилок. Взаємодія вірусів та лейкоцитів з гравцями та іншими елементами гри відбувалася без непередбачених ситуацій, що свідчить про гармонійне інтегрування.

Комплексне тестування інших механік гри

Було проведено всебічне оцінювання інших ігрових механік згідно з розробленими алгоритмами. Практично всі механіки виявилися працездатними та відповідали правильному алгоритму, забезпечуючи високий рівень занурення та комфорту для гравців. Результати тестування вказують на ефективність впроваджених механік та їх відповідність задуманому гейм-дизайну.

Висновки до третього розділу

У третьому розділі було детально розглянуто процес тестування гри, що включав валідацію, апаратне та програмне тестування, а також виявлення та виправлення помилок. Проведені тести підтвердили високу якість реалізації багатьох аспектів гри, а також вказали на деякі області, які потребують додаткової уваги та вдосконалення.

У підсумку, проведене тестування підтвердило готовність гри до використання, хоча деякі аспекти потребують вдосконалення. Результати тестування дозволяють зробити висновок про високу якість основних механік та інтерфейсу, забезпечуючи гравцям позитивні враження та комфортну взаємодію з грою. Подальше покращення виявлених недоліків сприятиме ще більшій стабільності та привабливості кінцевого продукту.

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

ВИСНОВКИ

У першому розділі було проведено детальний аналіз задачі та існуючих програмних продуктів, таких як The Walking Dead: Our World та AR Invaders, з акцентом на їх геймплейні механіки та економічний вплив. Це дозволило визначити ключові елементи, які можуть бути корисними при розробці власної гри, а також обґрунтувало вибір технологій та інструментів для розробки, зокрема Unity, Visual Studio та AR Foundation.

Другий розділ описує процес проектування та розробки гри, включаючи загальний алгоритм роботи програми, створення 3D моделей та функціональних алгоритмів для основних ігрових процесів. Було розроблено чітку структуру проекту, що спростило подальшу реалізацію гри, забезпечуючи логічну послідовність дій та взаємодію гравців із грою.

Третій розділ присвячений опису роботи з програмним додатком та його тестуванню. Проведене тестування підтвердило високу функціональність та зручність інтерфейсу а також ефективність ігрових механік. Виявлені незначні недоліки були проаналізовані та враховані для подальшого вдосконалення гри.

Таким чином, проведений аналіз, проектування та тестування підтвердили доцільність обраних підходів та технологій для розробки інтерактивної гри з високим рівнем занурення та взаємодії для користувачів.

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Unity Store [Електронний ресурс] – посилання на ресурс: <https://store.unity3d.com/>.
2. Unity3D - Saving/Loading Dynamic Scriptable Objects [Електронний ресурс] – посилання на ресурс: <https://discussions.unity.com/t/saving-loading-dynamic-scriptable-objects/912068>
3. Game Design and Development Tutorials [електронний ресурс] -посилання на ресурс: <https://www.gamedev.net/>
4. Навчальна платформа Unity про оптимізацію ігрових проектів [електронний ресурс] - посилання на ресурс: <https://learn.unity.com>
5. Лекції та матеріали з розробки ігор від професіоналів галузі [електронний ресурс] - посилання на ресурс: <https://www.gdcvault.com/free>
6. Ресурс із безкоштовними ігровими активами та статтями про розробку ігор [електронний ресурс] - посилання на ресурс: <https://opengameart.org>
7. Unity AR Foundation [Електронний ресурс] – посилання на ресурс: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html>
8. Unity XR Simulation [Електронний ресурс] – посилання на ресурс: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.1/manual/xr-simulation/simulation-overview.html>
9. Medium – статті про AR-розробку в Unity [Електронний ресурс] – посилання на ресурс: <https://medium.com/tag/unity-ar>
10. Посібник: Створення Контролера Аніматора [Електронний ресурс] – посилання на ресурс: <https://docs.unity3d.com/2019.4/Documentation/Manual/AnimatorControllerCreation.html>

		Марчук Н.А.			«Житомирська політехніка».24.122.09.000 – ПЗ	Арк.
		Граф М. С.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

ДОДАТКИ

Скрипт арени

Скрипт вірусів та лейкоцитів

```
using UnityEngine;

public class BalloonScript : MonoBehaviour
{
    public enum BalloonType { Virus, Healthy }
    public BalloonType balloonType;
    public float maxHeight = 3.0f;

    private int virusHealth = 1;

    void Start()
    {
        if (balloonType == BalloonType.Virus &&
            UnityEngine.SceneManagement.SceneManager.GetActiveScene().buildIndex == 3)
        {
            virusHealth = 2;
        }
    }

    private void Update()
    {
        transform.Translate(Vector3.up * Time.deltaTime * 0.2f);

        if (transform.position.y > maxHeight)
        {
            if (balloonType == BalloonType.Virus)
            {
                var uiManager = Object.FindFirstObjectByType<GameUIManager>();
                if (uiManager != null)
                {
                    uiManager.DecreaseLife();
                }
            }
            Destroy(gameObject);
        }
    }

    public void OnPopped()
    {
        if (balloonType == BalloonType.Virus && virusHealth > 1)
        {
            virusHealth--;
            return;
        }

        var uiManager = Object.FindFirstObjectByType<GameUIManager>();
        if (balloonType == BalloonType.Healthy && uiManager != null)
        {
            uiManager.ShowLosePanel();
        }
        Destroy(gameObject);
    }
}
```

Скрипт спавну

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

```

using UnityEngine;
using UnityEngine.XR.ARFoundation;

public class SpawnScript : MonoBehaviour
{
    public int virusesPerLevel = 20;
    public int healthyPerLevel = 10;
    public float levelTime = 30f;
    public GameObject virusPrefab;
    public GameObject healthyPrefab;
    public float spawnRadius = 2.0f;
    public ARPlaneManager planeManager;
    public GameObject arCamera;
    public GameObject bonusHealPrefab;
    public float bonusSpawnRadius = 2.5f;
    public float bonusChance = 0.1f;

    private int virusesSpawned = 0;
    private int healthySpawned = 0;
    private float spawnInterval;

    private GameUIManager uiManager;

    void Start()
    {
        spawnInterval = levelTime / (virusesPerLevel + healthyPerLevel);
        uiManager = FindFirstObjectByType<GameUIManager>();
        if (arCamera == null)
        {
            Camera cam = Camera.main;
            if (cam != null)
                arCamera = cam.gameObject;
        }
        StartCoroutine(SpawnRoutine());
    }

    IEnumerator SpawnRoutine()

```

```

{
    while (virusesSpawned < virusesPerLevel || healthySpawned < healthyPerLevel)
    {
        yield return new WaitForSeconds(spawnInterval);
        List<ARPlane> planes = new List<ARPlane>();
        foreach (var arPlane in planeManager.trackables)
        {
            planes.Add(arPlane);
        }
        if (planes.Count == 0)
            continue;

        ARPlane plane = planes[Random.Range(0, planes.Count)];
        Vector3 planeCenter = plane.center;
        Vector3 spawnPos;
        int maxTries = 10;
        int tryCount = 0;
        do
        {
            Vector2 randomInPlane = Random.insideUnitCircle * spawnRadius;
            spawnPos = planeCenter + new Vector3(randomInPlane.x, 0,
randomInPlane.y);
            tryCount++;
        }
        while (arCamera != null && Vector3.Distance(spawnPos,
arCamera.transform.position) < 1.0f && tryCount < maxTries);
        bool spawnVirus = false;
        if (virusesSpawned < virusesPerLevel && healthySpawned < healthyPerLevel)
            spawnVirus = Random.value > 0.5f;
        else if (virusesSpawned < virusesPerLevel)
            spawnVirus = true;

        if (spawnVirus)
        {
            Instantiate(virusPrefab, spawnPos, Quaternion.identity);
            virusesSpawned++;
        }
    }
}

```

```

        else
        {
            Instantiate(healthyPrefab, spawnPos, Quaternion.identity);
            healthySpawned++;
        }

        if (Random.value < bonusChance && uiManager != null &&
            uiManager.CanSpawnHealBonus())
        {
            Vector2 bonusOffset2D = Random.insideUnitCircle * bonusSpawnRadius;
            Vector3 bonusSpawnPos = planeCenter + new Vector3(bonusOffset2D.x, 0,
            bonusOffset2D.y);

            Instantiate(bonusHealPrefab, bonusSpawnPos, Quaternion.identity);
            uiManager.MarkHealBonusSpawned();
        }
    }
}

```

Скрипт бонусів

```
using UnityEngine;
```

```
public class BonusScript : MonoBehaviour
{
```

```
    public float speed = 0.2f;
    public float maxHeight = 3.0f;
    public GameObject smoke;
```

```
    private bool used = false;
```

```
    void Update()
```

```
    {
        transform.Translate(Vector3.up * Time.deltaTime * speed);
        if (transform.position.y > maxHeight)
        {
            Destroy(gameObject);
        }
    }

```

```
    public void ApplyBonus(Vector3? effectPosition = null, Vector3? effectNormal =
    null)
    {
```

```
        if (used) return;
        used = true;
```

```
        var uiManager = Object.FindFirstObjectByType<GameUIManager>();
        if (uiManager != null)
        {
            uiManager.AddLife();
        }

```

```
        if (smoke != null && effectPosition.HasValue && effectNormal.HasValue)

```

```
        Instantiate(smoke, effectPosition.Value,
Quaternion.LookRotation(effectNormal.Value));

        Destroy(gameObject);
    }

    private void OnTriggerEnter(Collider other)
    {
        if (used) return;
        if (other.CompareTag("PlayerBullet") || other.CompareTag("Player"))
        {
            ApplyBonus();
        }
    }
}
```