

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук (або центр післядипломної освіти, або навчально-науковий центр заочної форми навчання)
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Ігровий програмний застосунок у жанрі Tower Defence
(тема)

Виконав:
студент 4 курсу, групи ПЗПІ-20-2

Гребінченко М.В.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова

Керівник асистент кафедри ПІ Матвєєв Д.І
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри

(підпис)

З.В.Дудар
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
 Кафедра _____ програмної інженерії _____
 Рівень вищої освіти _____ перший (бакалаврський) _____
 Спеціальність _____ 121 – Інженерія програмного забезпечення _____
 Тип програми _____ Освітньо-професійна _____
 Освітня програма _____ Програма Інженерія _____
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Гребінченку Максиму Володимировичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи _____ Ігровий програмний застосунок у жанрі Tower Defence

Затверджена _____ наказом _____ по _____ університету _____ від _____

20.05. 2024р. № 471 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії _____ 10.06.2024

3. Вихідні дані до роботи Розробити ігровий застосунок в жанрі tower defence, котрий міститиме на рівень з чотирма хвилями ворогів, систему веж, головне меню, який буде розроблено за допомогою мови візуального програмування Blueprint на ігровому двигуну Unreal Engine 5.

4. Перелік питань, що потрібно опрацювати в роботі

Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі та постановка задачі	08.04.2024	<i>виконано</i>
2	Аналіз та вибір ігрового двигуна	10.04.2024	<i>виконано</i>
3	Вивчення робочого інструментарію двигуна	18.04.2024	<i>виконано</i>
4	Проектування ігрового застосунку	24.04.2024	<i>виконано</i>
5	Розробка прототипу	30.04.2024	<i>виконано</i>
6	Проектування інтерфейсу гри	15.05.2024	<i>виконано</i>
7	Реалізація ігрових механік та UI	18.05.2024	<i>виконано</i>
8	Тестування ігрового застосунку	28.05.2024	<i>виконано</i>
9	Підготовка пояснювальної записки	31.05.2024	<i>виконано</i>
10	Підготовка презентації та доповіді	03.06.2024	<i>виконано</i>
11	Нормоконтроль	04.06.2024	<i>виконано</i>
12	Рецензування	05.06.2024	<i>виконано</i>
13	Попередній захист	06.06.2024	<i>виконано</i>
14	Занесення диплома в електронний архів	07.06.2024	<i>виконано</i>
15	Допуск до захисту у зав. кафедри	09.06.2024	<i>виконано</i>

Дата видачі завдання 8 квітня 2024р.

Студент _____ Гребінченко М.В.
(підпис)

Керівник роботи _____ асистент кафедри ІІІ Матвеев Д.І.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра, 69 стор., 45 рис., 7 табл., 11 джерел.

ГЕЙМПЛЕЙ, ГРА, ЛЕВЕЛ ДИЗАЙН, ОПТИМІЗАЦІЯ, ПРОГРАМНИЙ ЗАСТОСУНОК, СТРАТЕГІЯ, ASSETS, BLUEPRINTS, C++, TOWER DEFENCE, UNREAL ENGINE 5, VFX.

Об'єкт розробки – ігровий програмний застосунок в жанрі Tower Defence.

Мета розробки – створення ігрового програмного застосунку, який матиме у собі готовий рівень з декількома хвилями різних ворогів, та вежами які їх атакують.

Метод рішення – середовище розробки Unreal Engine 5, мова візуального програмування Blueprints.

У результаті розробки створено ігровий програмний застосунок у жанрі Tower Defence, котрий має в собі рівень з чотирма хвилями ворогів та різними вежами які їх атакують.

GAMEPLAY, GAME, LEVEL DESIGN, OPTIMIZATION, SOFTWARE APPLICATION, STRATEGY, ASSETS, BLUEPRINTS, C++, TOWER DEFENCE, UNREAL ENGINE 5, VFX.

The object of development is a game software application in the genre of Tower Defence.

The purpose of the work is to create gaming software application that will have a complete level with a few waves of enemies and tower that attacks them.

Solution method – Unreal engine 5 framework using built-it Blueprint system.

As a result of the development, a game software application was created, which have a complete level with a few waves of enemies and tower that attacks them

Заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу EIArKhNURE.

Я, Гребінченко Максим Володимировч, студент гр. ПЗП-20-2, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Ігровий програмний застосунок в жанрі Tower Defence», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(на) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі	10
1.1 Аналіз предметної галузі	10
1.2 Виявлення та вирішення проблем	13
1.3 Постановка задачі.....	17
1.4 Виявлення проблем	19
2 Формування вимог програмної системи	22
2.1 Постановка мети.....	22
2.2 Загальний опис системи.....	22
2.3 Основний функціонал системи.....	22
2.4 Загальні обмеження.....	23
2.5 Припущення та залежності	23
3 Архітектура та проектування програмної системи	25
3.1 Прийняті архітектурні рішення	25
3.2 Прийняті програмні концепції та архітектурні підходи.	26
4 Опис прийнятих програмних рішень	30
4.1 UI елементи гри.....	30
4.2 Реалізація веж.....	37
4.3 Вороги.....	43
5 Тестування розробленого програмного забезпечення	47
5.1 Обґрунтування вибору тестування.....	47
5.2 Вимоги до тестування інтерфейсу.....	47
5.3 Вимоги до тестування геймплею.....	48
5.4 Результати тестування	49
6 Впровадження програмного забезпечення	53
6.1 Вибір сервісу для публікації гри	53
Висновки.....	56
Перелік джерел посилання	57

	7
ДОДАТОК А Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ....	59
ДОДАТОК Б Лістинг програмного коду	60
ДОДАТОК В Слайди презентації.....	63

ВСТУП

У світі ігрової індустрії за останні роки міцно закріпилися популярні жанри, такі як RPG, action, shooter, adventure, strategy. Вони привертають увагу гравців своїми захоплюючими сюжетами, складними механіками та вражаючими візуальними ефектами. Але серед тіні цих мастодонтів можна знайти жанр, який хоч і не настільки помітний, проте залишається стабільним і привабливим для певної категорії гравців. Це жанр Tower Defence. Він не привертає таку ж увагу, як інші жанри, однак за скромним фасадом простої ігрової механіки ховається величезний потенціал для творчості та інновацій.

Ігри в жанрі Tower Defence, або захист вежі, зазвичай являють собою захоплюючу суміш стратегії та аркади. Основна мета гравця - обороняти свою базу або інший об'єкт від хвиль ворогів, що наступають, будуючи захисні споруди - вежі, пастки, перешкоди та інше, залежно від гри. Можливості для креативу в цьому жанрі здаються обмеженими, адже основна механіка залишається незмінною: будівництво веж, оборона, виживання, покращення та оптимізація будівництва веж і знову по колу. Але незважаючи на ці обмеженості, тут приховано багато можливостей для творчості.

Одним із ключових аспектів розробки ігор у жанрі Tower Defence є робота над атмосферою та стилістикою. Адже хоч основна механіка залишається незмінною, кожна гра у цьому жанрі має свій неповторний візуальний стиль, який допомагає виділитися на тлі інших подібних проектів або навіть стати культовою грою. Це може бути футуристичний світ, середньовічні землі або навіть космічний простір чи фантастика. Візуальні елементи, звуковий супровід та загальна атмосфера гри можуть значно вплинути на сприйняття гравцями і зробити гру унікальною та запам'ятовуючою.

Крім візуальної складової, значну роль відіграє ігровий дизайн та інноваційність механік. Розробники мають можливість експериментувати з різними підходами до структури рівнів, типів ворогів, видів захисних споруд та їхніх можливостей. Наприклад, можна впровадити систему розвитку веж, де кожна вежа може мати кілька гілок розвитку з унікальними здібностями та

характеристиками. Це дозволяє гравцям обирати стратегію, яка найбільше відповідає їхньому стилю гри, і робить кожну партію унікальною.

Окрім цього, важливою складовою є сюжет та історія, що супроводжує гравця протягом гри. Хоча у багатьох іграх жанру Tower Defence сюжет відіграє другорядну роль, існують проекти, де добре прописана історія додає глибини та інтриги. Наприклад, гравець може бути частиною великої війни між королівствами, захищаючи свою землю від навали ворогів, або ж виконувати місії у далекому космосі, рятуючи людство від інопланетної загрози.

Також важливим аспектом є мультиплеєрний режим, який дозволяє гравцям змагатися один з одним або співпрацювати у захисті спільної бази. Така взаємодія додає грі динаміки та робить її більш захоплюючою. Гравці можуть випробовувати різні стратегії, обмінюватися досвідом та досягненнями, що сприяє довготривалому інтересу до гри.

Таким чином, незважаючи на те, що жанр Tower Defence може здаватися непомітним серед гучніших ігрових трендів, він надає унікальний майданчик для творчості та інновацій. Розробники ігор у цьому жанрі мають можливість привнести свою унікальну візію до вже знайомої механіки, створюючи захоплюючі та запам'ятовуються ігрові досліди для мільйонів гравців по всьому світу. Кожна нова гра в жанрі Tower Defence може стати свіжим подихом у світі ігор, демонструючи, що навіть у найпростішій на перший погляд механіці можна знайти нескінченні можливості для інновацій та творчості.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Жанр Tower Defence (TD) у світі відеоігор виник наприкінці 20-го століття і став одним із найпопулярніших жанрів стратегій. Історія його виникнення пов'язана з грою "Rampart", що вийшла у 1990 році. "Rampart" поєднувала елементи стратегії, головоломки та аркадного геймплею, в якій гравець мав захищати свій замок від ворожих військ, будуючи оборонні структури. Саме цю гру можна вважати засновником жанру (див. рис. 1.1).



Рисунок 1.1 – Фрагмент гри «Rampart» проти іншого гравця.

Також з виходом відомої гри Warcraft 3 та її доповнення Frozen throne, вона отримала потужний редактор карт і як наслідок на неї з часом також почали з'являтися модифікації які перетворюють гру на Tower Defence (див. рис. 1.2). Спільнота фанатів цієї модифікації була невеликою.



Рисунок 1.2 – Модифікація Tower Defense для гри Warcraft 3.

Справжній прорив у цьому жанрі стався з виходом гри "Desktop Tower Defense" у 2007 році (див. рис. 1.3). DTD - це популярна браузерна гра у жанрі Tower Defense, розроблена в 2007 році Дейвом Скалтоном. Гра набула широкої популярності завдяки своїй простоті, веселому стилю та високій реіграбельності.



Рисунок 1.3 – Гра Desktop Tower Defense.

Ця гра була створена незалежним розробником і була доступна онлайн. "Desktop Tower Defense" визначила основні концепції жанру: гравець мав будувати вежі (турелі) на мапі, які автоматично атакували ворожі одиниці, що рухалися по маршруту. Ця проста, але дуже захоплююча механіка знайшла велику популярність серед геймерів і надихнула на створення безлічі подібних ігор.

Від "Desktop Tower Defense" жанр Tower Defense почав активно розвиватися. Розробники вносили різноманітні ідеї та нововведення у геймплей, додавали різноманітні типи веж, ворожих одиниць, особливі ефекти та умови перемоги. Іноді навіть розробникам вдалося переосмислювати цей жанр, що призводило до випуску таких культових ігор як Plants VS Zombies (див. рис. 1.4).



Рисунок 1.4 – Гра Plants VS Zombies.

Поступово, TD ігри зайняли свою нішу в індустрії відеоігор, і сьогодні можна знайти сотні ігор у цьому жанрі для різних платформ, включаючи ПК, консолі та мобільні пристрої.

1.2 Виявлення та вирішення проблем

Жанр Tower Defence завжди був популярним серед гравців, але, як у будь-якому іншому жанрі, він також стикається зі своїми проблемами. На жаль навіть деякі проблеми лежать в самій суті Tower Defence[1].

Перш за все варто відзначити те, що гра ніколи не зможе полюбитися широкій аудиторії на рівні популярних AAA-ігор в жанрі шутеру або RPG-Adventure. А також те що гра просто не здатна генерувати контент такого об'єму, щоб затримати гравця у грі довше ніж цього б хотілося. Це пов'язане з тим, що вона не вимагає від гравця продвинутих ігрових навичок, а також часто не реалізовує у собі мультиплеєрну складову, що виключає змагальницький інтерес у гравців, і єдине на що вони будуть розраховувати це лише контент який був створений розробниками.

Іншою фундаментальною проблемою можна виділити невисоке захоплення ігрової уваги у гравця. Очевидно, якщо гравця змусити виконувати одні і ті ж дії на протязі кількох годин, то це невблаганно призведе до падіння інтересу та нудьги. Проблема з обмеженим геймплеєм та невеликим розмахом для творчості призводить до того, що розробникам доволі важко захопити увагу гравця.

До іншої категорії проблем можна виділити типові помилки, які можуть трапитися з будь-якими іграми враховуючи сучасні тренди розробки а також останні тенденції на ринку комп'ютерних та мобільних відеоігор, де останні майже закріпили за собою назву «Донатне сміття», що має на увазі головна ціль застосунку це змусити гравця витратити гроші.

В результаті серед цих «сучасних» хвороб можна виділити наступні проблеми:

- монотонність;
- брак інновацій;
- поганий баланс;
- монетизація.

Розберемо почерзі ці проблеми та способи їх рішення.

1.2.1 Монотонність

Ризик монотонності геймплею в Tower Defence іграх становить серйозну проблему, яка може значно підірвати інтерес гравців і знизити загальну оцінку гри. Ця проблема виникає через недостатню різноманітність в геймплеї, що може призвести до втоми та втрати мотивації вдосконалюватися.

Одним з основних факторів, який спричиняє монотонність, є обмежена кількість типів монстрів і об'єктів, з якими гравець зіштовхується. У багатьох іграх Tower Defence можна помітити однотипних монстрів та обмежені варіанти розміщення веж. Наприклад, гравець може бачити тільки кілька типів ворогів, які атакують з одних і тих же напрямків, що призводить до відчуття рутини та одноманітності.

Рішенням цієї проблеми може бути введення більшої кількості ворогів та різноманітних типів атак. Розробники можуть створити ширший асортимент ворожих одиниць з різними характеристиками, здатними атакувати з різних напрямків та використовувати різноманітні стратегії. Наприклад, додавання ворогів з різними швидкостями руху, типами броні, здатностями до відновлення здоров'я тощо може створити більш складний та цікавий геймплей.

Крім того, розширення варіантів розміщення веж також може допомогти запобігти монотонності гри. Замість того, щоб обмежувати гравця в областях, де можна будувати вежі, розробники можуть створити більш відкриті та різноманітні місця для будівництва. Це дозволить гравцям розвивати різні стратегії оборони та експериментувати з різними комбінаціями веж.

Загалом, монотонність геймплею в Tower Defence іграх може бути успішно подолана за допомогою розширення різноманітності ворожих одиниць та варіантів розміщення веж.

1.2.2 Брак інновацій

Брак інновацій у Tower Defence іграх може стати причиною швидкої нудьги у гравців і зниження інтересу до гри. Наявність стандартних механік та відсутність нововведень можуть зробити гру малопривабливою та нецікавою на тлі

конкурентів[2]. Розробники мають великий потенціал для експериментів з новими ідеями та механіками, які можуть урізноманітнити геймплей і зробити гру більш привабливою для гравців.

Одним з способів вирішення цієї проблеми є впровадження різноманітних режимів гри. Наприклад, режим мультиплеєра дозволить гравцям взаємодіяти один з одним у реальному часі, спільно обороняти базу або змагатися між собою. Режим співробітництва може надати можливість гравцям об'єднати зусилля для подолання спільних ворогів та досягнення спільних цілей. Такі режими розвивають соціальний аспект гри і стимулюють співпрацю між гравцями.

Крім того, розробники можуть впроваджувати пряму взаємодію гравця з ігровим світом у вигляді якогось геро. Це може включати можливість керування головним персонажем, який має унікальні здібності та навички для допомоги в обороні бази або в нападах на ворогів. Такий підхід дозволить гравцям більш активно взаємодіяти з грою та відчувати себе більш інтегрованим у процес.

Загалом, вирішення проблеми браку інновацій у Tower Defence іграх вимагає від розробників творчого підходу та впровадження нових ідей та механік. Шляхом введення різноманітних режимів гри та прямої взаємодії гравця з ігровим світом, розробники можуть створити більш цікавий і захоплюючий геймплей, який буде привертати та захоплювати увагу гравців на тривалий час.

1.2.3 Чутливий баланс

Чутливість балансу у Tower Defence іграх є однією з найважливіших складових геймплею, оскільки вона безпосередньо впливає на відчуття гравця від гри та його загальний досвід. Навіть прості ігри потребують детального пропрацювання балансу, оскільки неправильно встановлений рівень складності може призвести до нудьги або втрати інтересу гравця[3].

Перш за все, важливо забезпечити гравця відчуття досягнення та задоволення від гри. Якщо гра буде занадто простою, гравець може відчути нудьгу та втратити бажання продовжувати її. Навпаки, якщо гра буде занадто складною, гравець може відчути розчарування та втратити мотивацію до завершення гри. Тому важливо

створити такий баланс складності, щоб гравець почував себе викликаним, але не занадто перевантаженим.

Крім того, розробники повинні враховувати поведінку гравців та їхній рівень навичок. Наприклад, якщо гравець успішно пройшов легкий рівень, він може очікувати більшого виклику на наступних рівнях. Тому важливо забезпечити плавний перехід від легкого до складнішого рівня, уникаючи різких перепадів у складності, які можуть здатися гравцям несправедливими або дискомфортними.

Єдиний спосіб вирішити цю проблему - це максимально кропітко підходити до питання балансу в грі. Це означає не лише тестування гри на різних етапах розробки, але й уважне врахування відгуків гравців та їхніх реакцій на різні аспекти геймплею. Розробники повинні постійно вдосконалювати баланс гри, забезпечуючи оптимальний рівень виклику та задоволення для всіх категорій гравців.

1.2.4 Монетизація

Монетизація у Tower Defence іграх, особливо на мобільних платформах, може стати складною задачею для розробників. Занадто агресивна модель монетизації, така як механіка PayWall або Pay-to-Win[4], може вразити гравців і спричинити негативне відношення до гри. У мультиплеєрних іграх це особливо важливо, оскільки нерівність у доступі до преміум-вмісту може порушити баланс гри і вплинути на загальний досвід користувача.

Розробники повинні шукати збалансований підхід до монетизації, який не тільки забезпечить їм прибуток, але й не порушить геймплей і не створить негативних емоцій у гравців. Одним із можливих рішень є використання моделей монетизації, які базуються на добровільних пожертвах гравців за додатковий контент або переваги, які не дають прямої переваги тим, хто платить. Наприклад, це може бути система косметичних предметів або можливість пришвидшення прогресу за допомогою валюти гри, яка може бути отримана як в нагороду за геймплей, так і за реальні гроші.

Загалом, жанр Tower Defence має великий потенціал для інновацій та покращень, і правильне вирішення проблеми монетизації може зробити ці ігри ще більш захоплюючими та цікавими для гравців[5]. Важливо зберігати баланс між генеруванням прибутку та збереженням задоволення від гри у гравців, щоб забезпечити успіх і тривалість життя гри на ринку.

1.3 Постановка задачі

Гра розроблена у стилі середньовічного фентезі. Головний акцент буде зроблений на аудиторію яка не знайома або малознайома з іграми. Головне завдання мати простий, інтуїтивно зрозумілий інтерфейс та легкі правила гри, які освоюються навіть початківцями. Необхідно створити достатню кількість веж та ворогів аби не викликати у новачків почуття паніки від кількості веж та ворогів.

Варто також зробити зрозумілий дизайн рівнів. Важливо зробити акцент на місцях які будуть свідчити про те де можна побудувати вежі. Також необхідно приділити увагу рельєфу та кольорам на ньому аби гравцю було розуміло які можливі шляхи є у ворогів до кінцевої точки, замість інтерфейсних вказівників або інших елементів які вибивають його з потоку.

Баланс у грі буде вибудований з ухилом у більш просту сторону з різноманітним веж. Цей фактор може компенсувати відсутню глибоку стратегічну складову, адже гравці які не знайомі з іграми та не мають великого ігрового досвіду наврядчи будуть вибудовувати складні стратегії. На додачу до цього це також спростить баланс рівнів, адже якщо не буде дуже глибокої глибини стратегії та геймплейної взаємодії об'єктів між собою.

1.3.1 Цільова аудиторія

На основі вищезгаданих проблем і поставлених завдань, ми можемо чітко сформулювати цільову аудиторію нашої гри. Шляхом виділення людей з певними характеристиками, які ідеально підходять під концепцію нашої гри, ми можемо визначити такі категорії:

- Люди без ігрового досвіду. Така аудиторія, шукаючи відпочинок або новий хобі, може відчувати себе впевнено в нашій грі, оскільки вона спрямована на просту інтеракцію та легке засвоєння.
- Люди які хочуть пограти у відносно нескладну гру. Наша гра створена для тих, хто шукає відпочинок і розвагу без надмірної складності, тому ця категорія гравців відчує себе комфортно у нашому ігровому середовищі.
- Люди які хочуть отримати від гри простий виклик. Для тих, хто цінує відчуття досягнення і прогресу без надмірних труднощів, наша гра створить ідеальний баланс між викликом та задоволенням.
- Люди які мають слабкі фізичні ігрові якості. Наша гра, яка базується на стратегії та аналізі, дозволяє людям з обмеженими фізичними можливостями, такі як сповільнена реакція та слабка моторика, відчувати себе впевнено та здатними вирішувати ігрові задачі, які перед ними ставить гра.
- Люди різного віку. Наша гра відкрита для широкого спектру гравців, незалежно від їхнього віку або статі, що дозволяє будь-кому насолоджуватися ігровим процесом.
- Люди які схильно до неквапливого геймплею. Для тих, хто шукає спокій і розміреність у своїй грі, наша гра пропонує можливість грати у своєму темпі і насолоджуватися геймплеєм без поспішання.
- Люди які мають більш аналітичне мислення. Наша гра спонукає до стратегічного мислення та аналізу, тому вона особливо приваблива для тих, хто любить розвивати та використовувати свої аналітичні навички.

Отже, на основі вищевказаних критеріїв, ми можемо більш детально сформулювати цільову аудиторію нашої гри. Наша цільова аудиторія включає переважно чоловіків та жінок у віці від 20 до 50 років, що дозволяє охопити різні життєві ситуації та потреби.

До них входять новачки, які не мають значного ігрового досвіду, але бажають спробувати щось нове та цікаве. Вони шукають гру з простим інтерфейсом та

легкими для засвоєння механіками. Такі гравці цінують можливість швидко вникнути у гру без довгих навчань та складних інструкцій.

Також наша аудиторія складається з людей, які хочуть отримати від гри відносно нескладний виклик. Вони шукають баланс між розвагою та інтелектуальним стимулом. Це можуть бути як молоді професіонали, так і студенти, які хочуть відпочити після напруженого дня.

Окрему категорію складають люди з обмеженими фізичними можливостями або ті, хто не має швидких рефлексів. Для них важливо, щоб гра не вимагала надмірної фізичної активності та дозволяла насолоджуватися процесом у власному темпі[6].

Наша гра також приваблює людей різного віку та статі, забезпечуючи рівні можливості для всіх. Вона підходить для тих, хто любить аналітичне мислення та стратегічне планування, надаючи можливість розвивати свої навички в комфортному ігровому середовищі.

Таким чином, наша цільова аудиторія включає широкий спектр гравців, які шукають відносно просту, але захоплюючу гру, яка забезпечить їм відчуття досягнення та задоволення від ігрового процесу.

1.4 Виявлення проблем

Для того аби зробити гру не схожою на інші, важливо проаналізувати переваги та недостатки які мають конкуренти. Серед них можна виділити відому серію «Kingdom Rush»[7], а також «Infinitode 2»[8].

Переваги Kingdom Rush:

- ярна візуальна стилістика та звуковий супровід;
- чудовий дизайн рівнів;
- велике різноманіття та виважена прогресія ворогів.

Основними сильними сторонами серії «Kingdom Rush» є її естетичний вигляд та атмосферний звуковий супровід, що створюють захоплююче ігрове середовище. Чудовий дизайн рівнів і добре збалансована прогресія ворогів сприяють цікавому ігровому процесу та підтримують інтерес гравців протягом усього гри.

Недоліки Kingdom Rush:

- мала кількість веж, відсутність сильної прогресії;
- проблеми з балансом складності;
- механіки Pay-to-win;
- обмежений вибір гравця у стратегіях.

Слабкими місцями «Kingdom Rush» є обмежена кількість веж і відсутність суттєвого прогресу в грі, що може швидко стати нудним для гравців. Проблеми з балансом складності та наявність механік Pay-to-win можуть негативно вплинути на загальне враження від гри. Крім того, обмежені стратегічні можливості знижують варіативність геймплею.

Переваги Infinitode 2:

- глибока стратегічна складова;
- free-to-play гра з мінімальним IAP;
- висока реграбельність;
- регулярні оновлення.

«Infinitode 2» відрізняється своєю глибокою стратегічною складовою, що робить її привабливою для гравців, які люблять виклики та планування. Free-to-play модель з мінімальними внутрішньоігровими покупками (IAP) робить гру доступною для широкої аудиторії. Висока реграбельність та регулярні оновлення забезпечують тривалу цікавість до гри.

Недоліки Infinitode 2:

- високий поріг входу;
- велика кількість Grind'у;
- відсутній акцент на візуальну або наративну частину гри;
- репетативний геймплей.

Основні недоліки «Infinitode 2» включають високий поріг входу, що може відлякати новачків, та велику кількість Grind'у, що може втомити гравців. Відсутність акценту на візуальну або наративну частину гри може знизити залученість гравців, які цінують естетичні та сюжетні аспекти. Репетативний геймплей також може призвести до швидкої втрати інтересу.

На основі вищенаведених переваг та недоліків ігор «Kingdom Rush» та «Infinitode 2» можна зробити висновок, що головне при створенні успішної Tower Defence гри - це обрати гарний візуальний стиль та звуковий супровід, який забезпечить привабливу атмосферу. Продуманий дизайн рівнів та різноманітність ворогів сприяють підтриманню інтересу гравців.

Водночас, важливо уникати надмірно агресивних механік монетизації, які можуть зіпсувати враження від гри. Стратегічний елемент повинен бути добре продуманим, надаючи гравцям різноманітні можливості для планування та реалізації своїх ідей. Натомість, не варто змушувати гравця триматися у грі через надмірний Grind. Замість цього, слід створити завершену історію з цікавими викликами та досягненнями.

Таким чином, шлях до успіху полягає в поєднанні захоплюючої візуальної стилістики, глибокої стратегії та зваженої монетизації, що дозволить створити гру, яка виділяється на фоні конкурентів і залучає широку аудиторію.

2 ФОРМУВАННЯ ВИМОГ ПРОГРАМНОЇ СИСТЕМИ

2.1 Постановка мети

Метою розробки є створення програмної системи для Tower Defence гри на платформах Windows та мобільних пристроях. Гра повинна бути доступною на обох платформах, забезпечуючи високу продуктивність і невеликий об'єм встановлення. Програмна система має бути масштабованою, легко переносною між платформами та забезпечувати відмінний ігровий досвід на різних пристроях.

2.2 Загальний опис системи

Система представлятиме собою однокористувацький додаток, що працює автономно на Windows та мобільних пристроях. Програмна система не вимагає підключення до інтернету та не зберігає дані у зовнішніх базах даних.

Ігровий додаток складається лише з ігрового клієнта. Це система, що містить геймплей, графіку та інтерфейс користувача. Додаток повинен забезпечувати плавну гру як на ПК, так і на мобільних пристроях, з підтримкою сенсорного керування на мобільних платформах.

2.3 Основний функціонал системи

Функціонал додатка включає:

- Ігровий процес. Захист бази гравця від хвиль ворогів шляхом будівництва оборонних веж. Різноманітні рівні та карти, що збільшують складність і пропонують різні стратегії захисту. Можливість покращення та модернізації веж, щоб ефективніше боротися з ворогами.
- Інтерфейс користувача. Інтуїтивно зрозумілий інтерфейс для побудови та управління вежами. Доступ до налаштувань гри для персоналізації досвіду гравця. Можливість перегляду статистики гри, включаючи досягнення та прогрес.
- Адаптивність та масштабованість. Підтримка різних розмірів екрану та орієнтації на мобільних пристроях. Оптимізація продуктивності для забезпечення плавного геймплею на різних платформах.

Це головні елементи, з якими гравець буде взаємодіяти, або які будуть впливати на нього.

2.4 Загальні обмеження

Програмна система має наступні обмеження:

- Ігровий додаток повинен бути встановлений на персональний чи мобільний пристрій.
- Комп'ютер та мобільний пристрій повинні відповідати мінімальним вимогам системи.
- Гра не вимагає підключення до інтернету, окрім оновлення або завантаження нових рівнів (якщо передбачено).

Вони не мають стати серйозними проблемами для гравця і вирішуються доволі просто.

2.5 Припущення та залежності

Розглядаються наступні припущення та залежності:

- Вільне місце на пристрої більше або ж дорівнює необхідному об'єму виконавчих файлів ігрового додатку.
- Пристрій під'єднаний до мережі інтернет для завантаження гри та можливих оновлень (якщо передбачено).
- Наявне необхідне обладнання для функціонування системи.

На основі вищенаведених вимог, програма повинна забезпечувати високий рівень задоволення користувачів, пропонуючи захоплюючий ігровий досвід та стабільну роботу на всіх підтримуваних платформах.

Тепер знаючи специфікацію програмного забезпечення можна визначити які характеристики необхідні для комфортної гри як в ком'ютерну версію, так і в мобільну.

2.6 Вимоги до заліза

Вимоги до заліза комп'ютера:

- CPU: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz;

- GPU: NVIDIA GeForce GTX 1050 (3GB);
- Storage: Мати не менше 2 гб вільного місця для гри;
- RAM: 16 GB;
- OS: Windows 10;

Вимоги до заліза мобільного пристрою:

- CPU: 4-х ядерний процесор з частотою від 2 ГГц;
- GPU: Підтримка технології OpenGL та Vulkan;
- Storage: Мати не менше 0.5 Гб вільного місця для гри;
- RAM: 4 GB;
- OS: Android.

Вищевказані вимоги для Tower Defence гри орієнтовані на забезпечення високої продуктивності та відмінного ігрового досвіду на різних платформах, включаючи Windows та мобільні пристрої які працюють на базі android. Вона не потребує постійного підключення до інтернету та має інтуїтивно зрозумілий інтерфейс, що дозволяє гравцям легко керувати оборонними вежами та налаштовувати гру під свої потреби. Вимоги до апаратного забезпечення гарантують плавний геймплей та масштабованість, забезпечуючи задоволення від гри як на ПК, так і на мобільних пристроях.

3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 Прийняті архітектурні рішення

При розробці гри у жанрі Tower Defence важливо визначитись з тим як будуть реалізовані її основні елементи, а також необхідно створити відповідну діаграму яка буде відображати архітектуру нашого ігрового додатку.

Візуальна стилістика гри виконана у стилі реалістичного середньовічного фентезі. Головне завдання правильно розставити акценти на рівні, та правильно маніпулювати увагою гравця. Ми нізащо не повинні відволікати його від процесу занурення у гру. Всю інформацію яку гравець може отримати з ігрового світу має звідти ж і поступати. Ніякі інтерфейсні повідомлення або інша інформація не повинна відволікати гравця.

Для гри було вирішено створити 4 типи ворогів та 5 веж. Це оптимальна кількість для того аби не перевантажити гравця інформацією, надати йому різноманітний вибір з поміж веж, а також різномайття ворогів які будуть кидати гравцю виклик. Варто зазначити що вежі між собою ще будуть ділитися на дві категорії. Перші – класичні вежі які стріляють снарядами, як тільки ворог входить в радіус стрільби вежі, на томість другий тип – вежі аури. Це такі вежі які працюють постійно і накладають певний негативний ефект на ворогів протягом всього часу як він знаходиться у межах дії башні.

Для того аби гравець не нудьгував, а також не був фрустрований необхідно правильно вибудувати баланс, створити правильну прогресію, визначити кількість і розмір хвиль, кількість здоров'я бази до моменту програшу, швидкість ворогів, DPS веж та можливі їх комбінації на рівні задля більш ефективного використання за рахунок синергії деяких веж між собою, або ж грамотним положенням вежі на рівні.

Окрім чітко поставлених цілей варто також розробити архітектуру ігрового застосунку. Без неї реалізація гри буде майже неможливою. Гра – це складна і багаторівнева система, тому без детально розробленої архітектури можна лише створити примітивний та простий додаток. Важливо приділити велику увагу цьому процесу (див. рис. 3.1).

активуються, а після смерті ворога або на кінці шляху він деактивується (див. рис. 4.2).

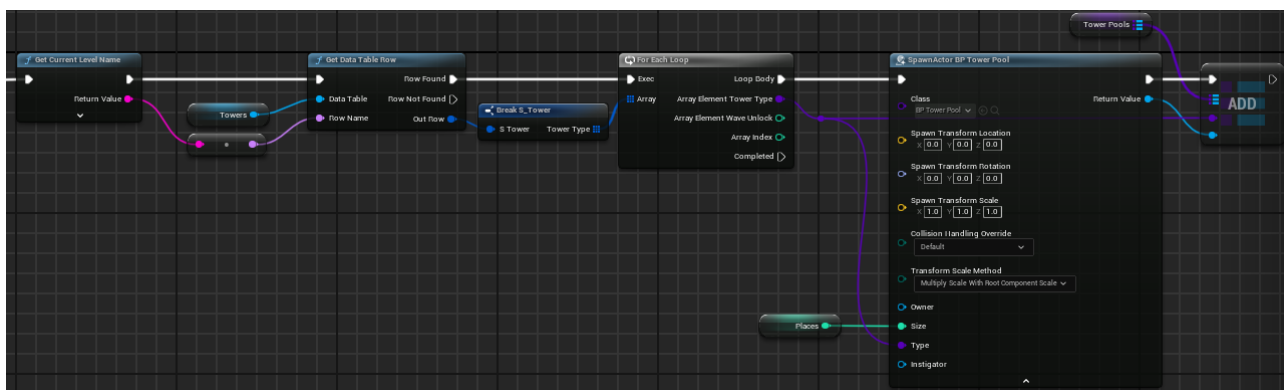


Рисунок 3.2 – Фрагмент коду реалізації Object Pool для веж

Механіка шляху для ворогів[10]. Поведінка ворогів, а саме їх спосіб слідування маршруту від початку, місця появи, до кінця була реалізована за допомогою інструменту Spline line. Цей інструмент дозволяє створювати криві різної довжини та форми. На основі цього всім ворогам була прописана логіка, у якій вони мають слідувати по одному й тому ж шляху до кінця. Цей метод було обрано оскільки спосіб із навігаційною сіткою міг призвести до неочікуваної поведінки, а також була більш складніша та непередбачуваніша в управлінні ніж Spline line (див. рис. 3.3). Можна навести приклад того, що вороги чим далі просуватимуться по рівню будуть скупчуватися в одній точці.

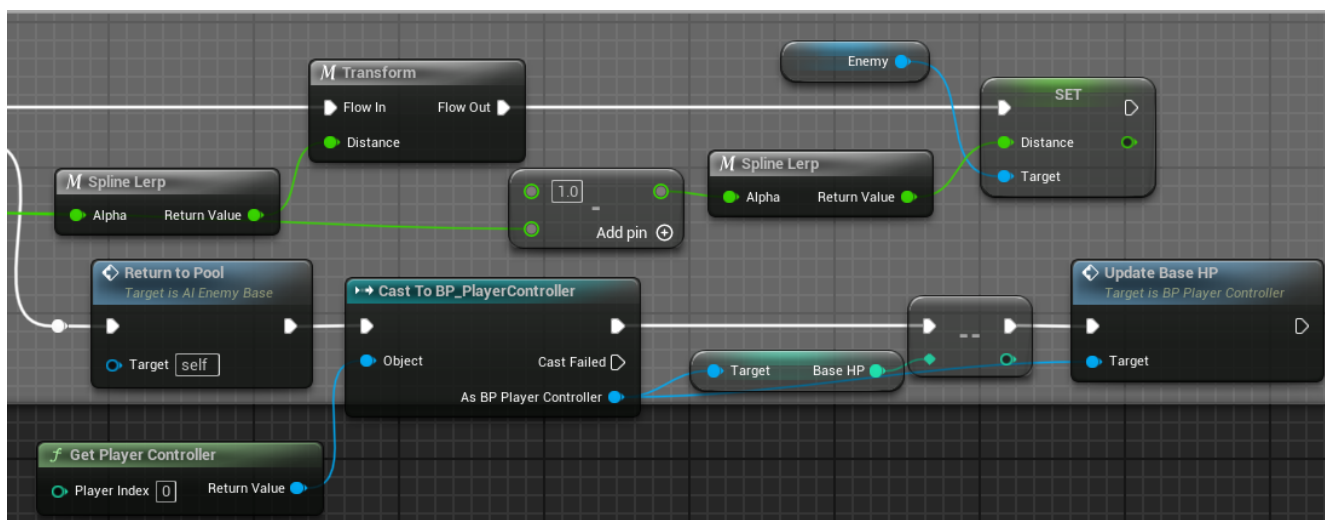


Рис 3.3 – Фрагмент коду відповідальний за рух ворогів по spline path

Організація структури ворогів в таблиці даних[11]. Для того аби було зручно налаштовувати ворогів які будуть з'являться на рівні та їх кількість було створено таблицю даних. Вона сформована за допомогою декількох структур які вкладені один в одного. Для того аби розібратись в реалізації поглянемо на кожен структуру окремо.

1-й рівень структури – тип ворога (див. рис. 3.4). У ньому вказується тип ворога та його кількість які будуть створені під час початку хвилі.

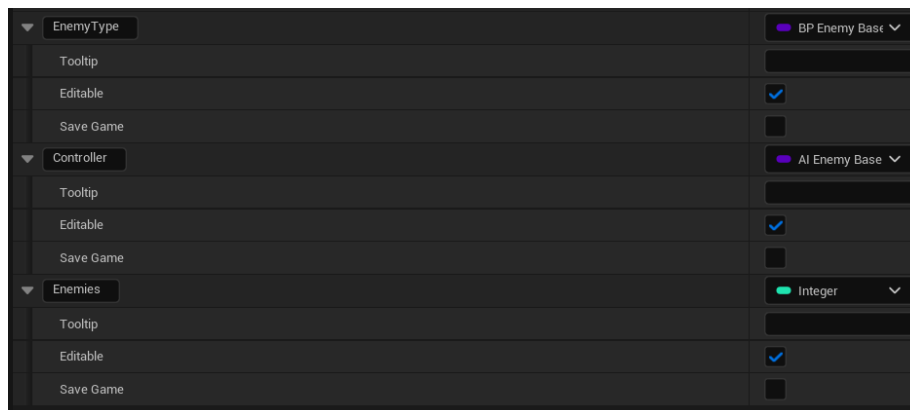


Рисунок. 3.4 – Структура групи ворогів

2-й рівень структури – структура шляху (див. рис. 3.5). У ньому вказується які вороги будуть створені на обраному шляху.



Рисунок 3.5 – Структура шляху для ворогів

3-й рівень структури – структура хвилі. Вона складається з масиву структур шляху і формує у собі інформацію про ворогів які будуть створені на одній хвилі (див. рис. 3.6).

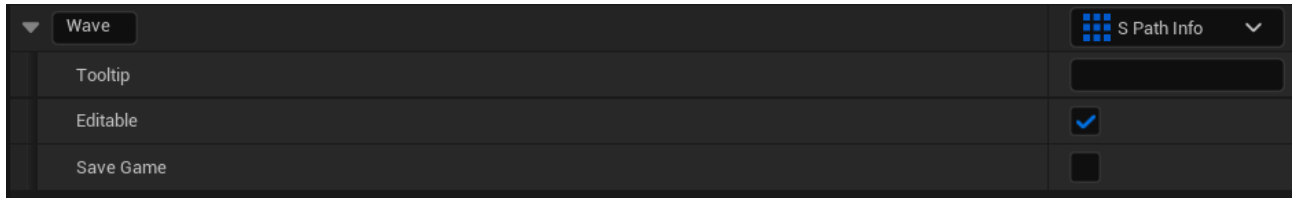


Рисунок 3.6 – Структура хвилі для ворогів.

4-й рівень структури – структура рівня. Вона відповідає за те кількість хвиль та які вони будуть на даному рівні. Це останній та найвищий рівень структури. (див. рис. 3.7)

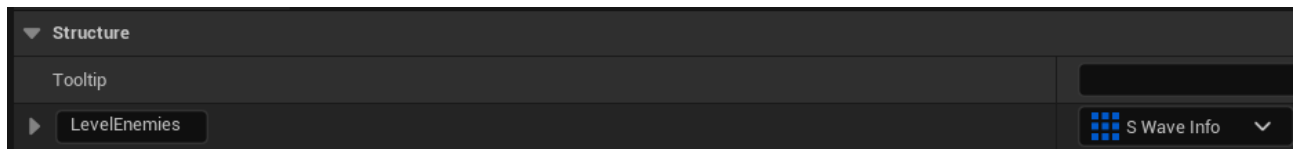


Рисунок 3.7 – Структура рівня для ворогів.

Підсумовуючи вищевказане в результаті отримаємо таку таблицю даних в якій зможемо керувати будь-якими ворогами починаючи від формування групи закінчуючи їх організацією на всьому рівні(див. рис. 3.8).

LevelEnemies	2 Array elements
Index [0]	1 members
Wave	1 Array element
Index [0]	3 members
Track	CemeteryPath
Wave	4 Array elements
Index [0]	3 members
EnemyType	BP_Ghoul
Controller	AI_EnergyBase
Enemies	1
Index [1]	3 members
EnemyType	BP_Goblin
Controller	AI_EnergyBase
Enemies	1
Index [2]	3 members
Index [3]	3 members
Delay	4.0
Index [1]	1 members

Рисунок 3.8 – Конкретний приклад налаштованих ворогів із таблиці даних

Хоч на перший погляд це здається не зручно і може навіть заплутати, але це чудовий підхід до структуризації хвиль та ворогів які у них будуть.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

4.1 UI елементи гри

4.1.1 UI головного меню

При проектуванні UI елементів головна ставка була поставлена на простоту та мінімалістичний дизайн (див. рис. 4.1). У якості заднього фону була використана картинка, яка б найбільше всього відповідала духу гри.

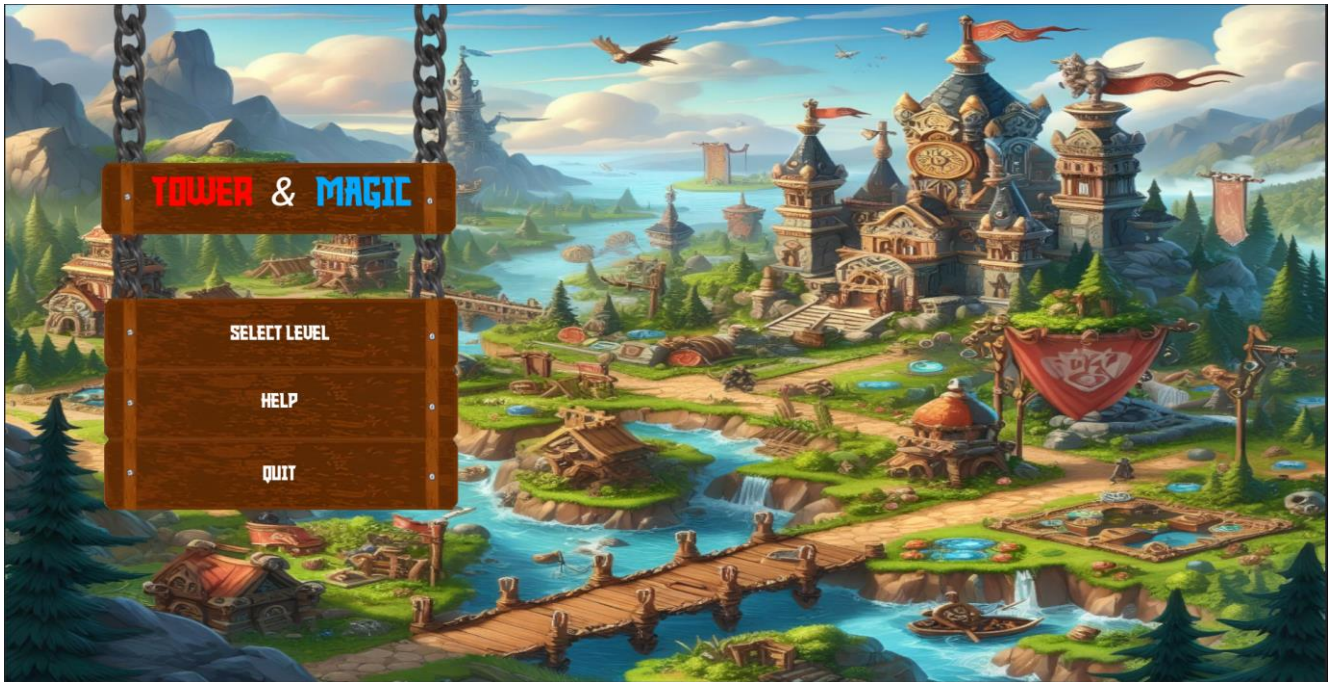


Рисунок 4.1 – Вигляд головного меню гри

Для простого та ефективного фідбеку для гравця був використаний простий прийом анімації UI елементу(див. рис. 4.2).



Рисунок 4.2 – Вигляд стану кнопки при наведенні курсора

Як можна бачити з рисунку, при наведенні на кнопку курсору миші, вона візуально змінює свій стан, що служить чудовим індикатором для гравця, що він зараз знаходиться на цій кнопці, що дозволить йому не загубитись.

4.1.2 UI вибору рівнів.

Коли гравець натискає на кнопку вибору рівнів, він переходить в відповідне меню, де перед ним постає рівень, який він може обрати, або ж гортати далі (див. рис. 4.3).



Рисунок 4.3 – Вигляд UI вибору рівня.

Перед гравцем постає не звичний список, або перерахування рівнів на одному UI елементі у вигляді блоків із цифрами, а постає затінена картинка рівня. При наведенні на неї курсором миші вона світлішає (див. рис. 4.4), після чого якщо гравець вирішив грати на цьому рівні, то він лише клацає та запускає рівень, в іншому випадку він за допомогою клавіш A/D перелистує далі у пошуку відповідного рівня якого він хоче. Також для кращого орієнтування по бокам від блоку який представляє рівень виставлено дві стілки, які слугують індикаторами того, що можна листати далі, у випадку з стрілкою вправо, що означає що є інші рівні які гравець може переглянути. А також ліва стрілка, яку не видно адже починаючи перегляд, з першого рівня не має куди рухатись назад ще більше.



Рисунок 4.4 – Ілюстрація рівня при наведеному курсорі.

У випадку якщо гравець вирішив повернутись до головного меню він може натиснути ESC.

4.1.3 UI меню будівництва

Інтерфейс будівництва відкривається при натисканні курсора миші на спеціальні місця для будівництва вежі. Інтерфейс візуально розділений на 2 половини. Верхній блок відповідає за відображення характеристик веж, які будуть побудовані. Вони з'являються у момент наведення миші на одну із веж. Нижня частина представляє із себе список веж доступних для будівництва. Воно реалізовано з модульним підходом який означає, що при зміні кількості веж або при додаванні нових вони будуть додаватися вниз списку підлаштовуючись під розміри екрану гравця (див. рис. 4.5).

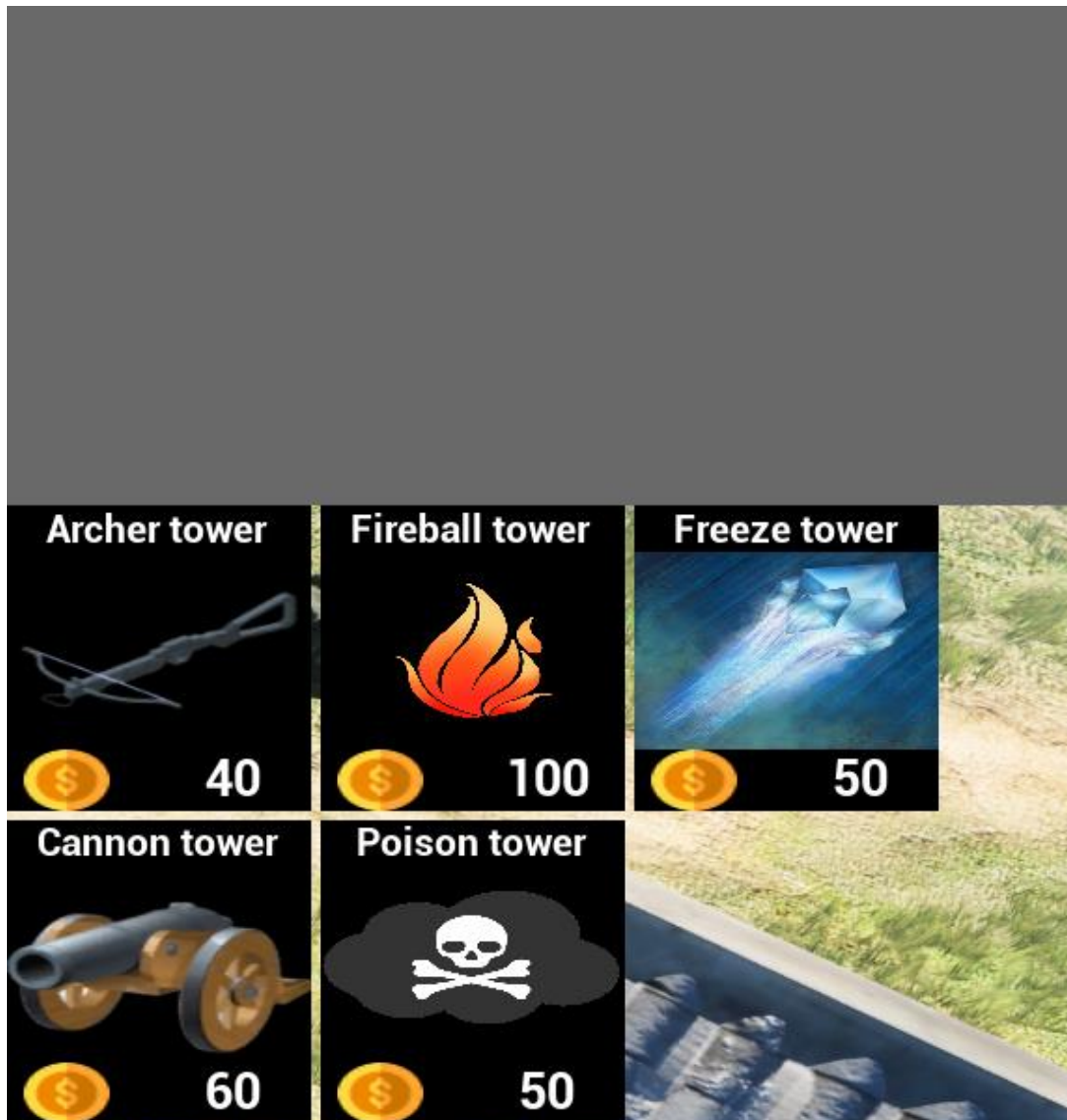


Рисунок 4.5 – Вигляд інтерфейсу будівництва.

Також варто зазначити що при наведенні миші на вежу, з'являється відповідна для цієї вежі інформація (див. рис 4.6).



Рисунок 4.6 – Приклад відображення інформації про вежу при наведенні на неї

Інформація відповідно змінюється в залежності на який тип вежі гравець навів курсор. У випадку з морозною вежею – буде вказано відсоток замедлення, а у випадку ядовитої, шкода в секунду.

4.1.4 UI гри

Було прийнято рішення, що під час гри гравця не повинно нічого відполікати. Враховуючи те, що у іграх жанру Tower Defence, як правило не має великої кількості важливої інформації за якою гравець має слідкувати, було прийнято рішення просто створити простенькі віджети які тримають його в курсі речей. До них відносяться лише дві речі: кількість золота та кількість здоров'я яке залишається у вежі (див. рис. 4.7).

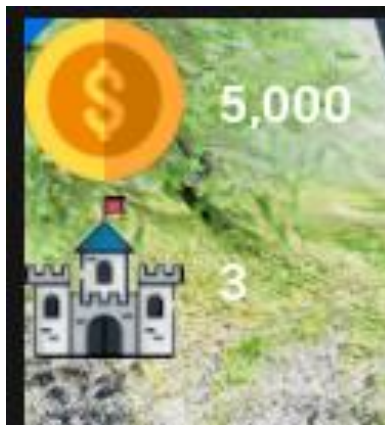


Рисунок 4.7 – Інтерфейс який відображається під час гри.

Іконки виставлені у лівому верхньому кутку і фактично не займають зайвого місця на екрані та не відволікають гравця від гри, при цьому залишаючись інформативними.

4.1.5 UI покращення вежі

При натисканні на побудовану вежу, гравцю справа відкривається меню покращення вежі. На ньому показані лише основні характеристики, які відповідно відображались і на моменті побудови вежі, але вже в цьому меню можна налаштувати тип стрільби вежі та побачити ціну покращення (див. рис. 4.8).

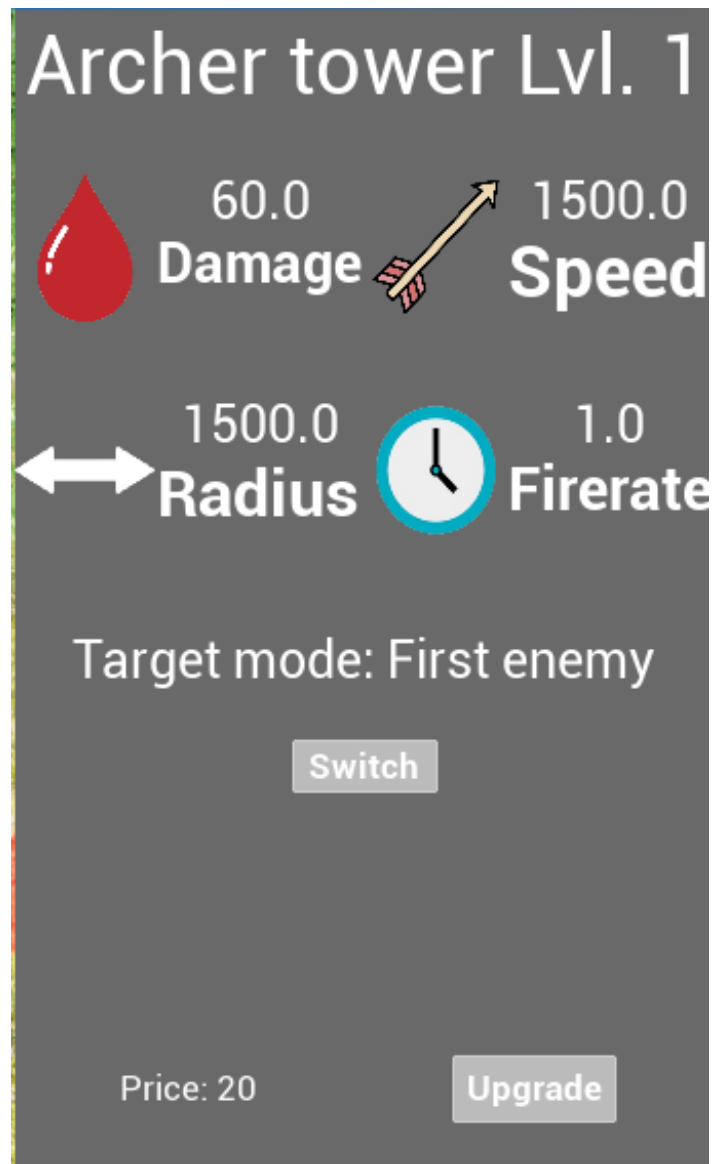


Рисунок 4.8 – Інтерфейс покращення вежі

При наведенні курсору на кнопку покращення, всі характеристики змінять свої значення і приймуть новий формат у вигляді нового потенційного значення, при цьому колір тексту зміниться на зелений. Він був обраний з метою того, аби показати що це позитивна зміна, яку зробить гравець. Саме через цю асоціацію було обрано цей колір.

При зміні режиму і відповідно при наведенні миші на кнопку змін не буде. Оскільки у вежі яка стріляє снарядами лише 2 варіанти стрільби: перша ціль, яка попала у радіус, або ж остання ціль. До того ж це порушить відповідну ментальну прив'язку покращення до зеленого кольору, адже це лише стратегічний вибір.

4.1.6 UI меню паузи, програшу, виграшу

Меню паузи, програшу та виграшу між собою майже не відрізняються. В даному прикладі (див. рис 4.9) на меню паузи зображені лише головні речі які можуть знадобитися гравцю.



Рисунок 4.9 – UI меню паузи

Це кнопки рестарту, виходу в головне меню та продовження гри. У випадку з екраном перемоги або програшу, продовження гри замінюється на вихід з гри, адже гравець успішно, або невдало завершив ігрову сесію, тому вихід з гри більш доцільніше використати в даних моментах. Так само як і в головному меню при наведенні курсора миші, вони надаватимуть візуальний фідбек.

4.2 Реалізація веж.

4.2.1 Снарядні вежі.

У грі було створено два типи веж. Вежі які стріляють снарядами та вежі які діють як аури. До першої категорії належать: стрілецька вежа, магічна вежа, вежа з пушкою. Розберемо кожен та визначимося з їх призначенням.

4.2.1.1 Стрілецька вежа

Стрілецька вежа (див. рис. 4.10) призначення бути першою вежею яка буде доступна гравцю і яку він побудує.

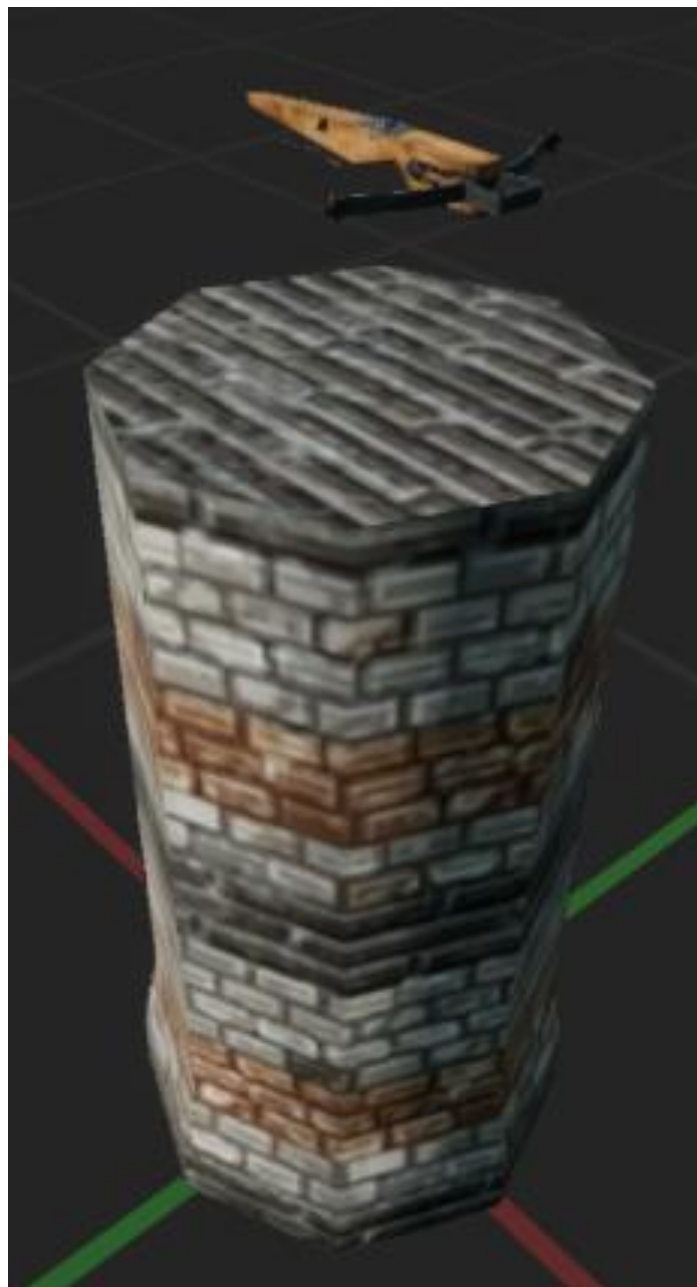


Рисунок 4.10 – Стрілецька вежа

Відповідно до її задач ціна цієї вежі та шкода яку вона наносить буде відповідно низькою. Скорострільність у неї буде помірною, а з прокачкою її рівня будуть з'являтися різні об'єкти на ній, як наприклад: ступінчаста огорожа поверх вежі, різні припаси зі стрілами та арбалетами, інше.

4.2.1.2 Магічна вежа.

Магічна вежа призначена бути другою вежею яка стане доступна гравцю (див. рис. 4.11). Її особливість полягає в АоЕ шкоді яку її снаряди можуть наносити ворогам, якщо вони будуть відповідно поруч у момент зіткнення снаряду з ворогом.



Рисунок 4.11 – Магічна вежа.

Ця вежа буде у якості снарядів використовувати вогняні шари, з відповідним візуальним ефектом горіння (див. рис. 4.12), що надає більшій правдоподібності грі. Також створено ефект невеличкого вибуху при зіткненні шару з ворогом, що виглядатиме більш показово і буде більш реалістичним явищем в контексті гри.

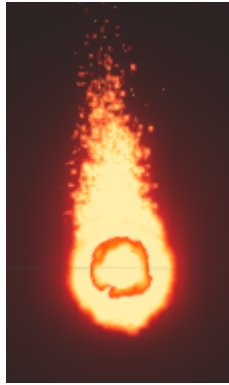


Рисунок 4.12 – Візуальний ефект вогняного шару, снаряду вежі

Для того аби реалізувати вибух і замінити візуальний ефект на інший необхідно було це зробити у C++. Далі наведено фрагмент коду який це реалізує:

Fireball.h:

```
#pragma once
#include "CoreMinimal.h"
#include "ProjectileBase.h"
#include "NiagaraComponent.h"
#include "Fireball.generated.h"

UCLASS()
class YOURGAME_API AFireball : public AProjectileBase
{
    GENERATED_BODY()

public:
    AFireball();

    UFUNCTION(BlueprintCallable, Category="VFX")
    void ProjectileBlastVFX(UNiagaraSystem* NewVFX);

protected:
    UPROPERTY(EditAnywhere, BlueprintReadWrite, Category="VFX")
    UNiagaraComponent* CurrentVFX;
};
```

Fireball.cpp:

```
#include "Fireball.h"
#include "NiagaraFunctionLibrary.h"
AFireball::AFireball()
{
    CurrentVFX = CreateDefaultSubobject<UNiagaraComponent>(TEXT("VFX"));
```

```

CurrentVFX->SetupAttachment(RootComponent);
}
void AFireball::ProjectileBlastVFX(UNiagaraSystem* NewVFX)
{
    if (CurrentVFX && NewVFX)
    {
        CurrentVFX->SetAsset(NewVFX);
        CurrentVFX->Activate(true);
    }
}

```

Зручність заміни візуального ефекту в Unreal Engine полягає в тому що велику частину роботи він бере на себе, залишаючи розробнику простий функціонал

4.2.1.3 Вежа з пушкою

Вежа з пушкою відноситься до більш пізнього виду вежі (див. рис. 4.13). Її ціль в тому що вона стріляє повільно, але її снаряди завдають велику шкоду.

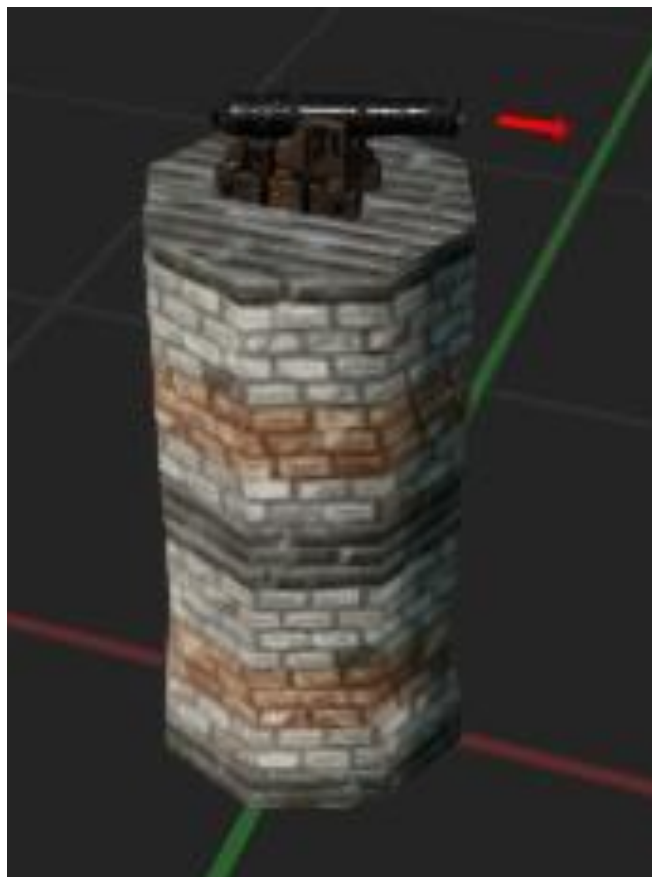


Рисунок 4.13 – Вежа з пушкою.

Її головна особливість в тому, що вона має збільшений радіус ураження. Вона чудово підходить у ситуаціях, коли вона має велике покриття шляху по якому йдуть

вороги, адже це означає постійне та безперервне нанесення шкоди ворогам. У іншому ж випадку, інші вежі б просто простоювали або були б не такі ефективні.

4.2.2 Вежі аури

Веж з аурним типом взаємодії було створено двоє. Це вежа морозу, яка сповільнює своїх ворогів на певний відсоток від їх поточної швидкості пересування, а також вежа яду, яка наносить постійну шкоду при перебуванні ворога у радіусі її дії.

4.2.2.1 Морозна вежа

Морозна вежа сама по собі не наносить шкоду сама по собі (див. рис 4.14). Її головна ціль це сповільнювати ворогів.



Рисунок 4.14 – Вежа морозу

Це робить її стратегічно важливою вежею і її правильне використання здатне змінити хід гри при правильному її використанні. Якщо поставити її поруч з

магічною вежою, то це може збільшити кількість шкоди в секунду для останньої, адже вороги які попадають у її радіус будуть сповільнені і отримуватимуть шкоду від одного снаряду, або ж використати її з вежою яду, яка буде постійно наносити їм шкоду протягом довшого часу.

4.2.2.2 Вежа яду

Вежа яду створена таким чином щоб наносити певну кількість шкоди в секунду будь-якому ворогу який зайде в її радіус (див. рис. 4.15).



Рисунок 4.15 – Вежа яду

Вона створена з метою надати певний компроміс у випадку, якщо ворогів буде багато, а позиціонування веж, або ж швидкість ворогів буде налаштована таким чином, що снаряди магічної вежі не зможуть зачепити достатню кількість ворогів одночасно. В даному випадку на допомогу прийде ця вежа, яка завжди нанесе свою шкоду. Її ефективність зростає з рівнем покриття шляху, яким йдуть вороги, а також тривалістю, як довго вони цей шлях будуть проходити.

4.3 Вороги

У грі створено 4 типи ворогів, кожен з них має власну ціль і відрізняється від інших або характеристиками, або метою з якою їх вводять.

4.3.1 Гуль

Першим ворогом з яким має зіткнутися гравець це гуль (див. рис 4.16). Він представляє з себе дещо ослаблену версію середнячка. Він має з'являтися на початку рівня і приносити середню кількість грошей.



Рисунок 4.16 – Ворог гуль

Головна ціль цього ворога – дати гравцю отримати певну кількість золота на початку гри, аби встигнути модернізувати або побудувати додаткові вежі на початку гри. Він не повинен викликати у гравців труднощі, а навпаки створити відчуття переваги і задоволення.

4.3.2 Гоблін

Гоблін – це приклад звичайного середнього моба який використовується всюди як масовка (див. рис 4.17).



Рисунок 4.17 – Ворог гоблін

Головна ціль гобліна це бути масовим ворогом. Він має середні характеристики, разом з небагато збільшеною швидкістю і зменшеною кількістю здоров'я. Його основна ціль це бути заглушкою між різними групами ворогів. Прикладом цього може слугувати як і перерва між тематичними групами ворогів,

як наприклад: початкова хвиля ворогів – гобліни – хвиля сильних але повільних ворогів – гобліни – бос.

4.3.3 Ліч

Ліч – це моб який архитепічно відноситься до категорії босів у даному жанрі (див. рис 4.18). Це як правило виражено в його великій кількості здоров'я та низькій швидкості, а нагорода за його голову висока. Він як правило не запускається у великій кількості, а слідує певному шаблону. Вперше з'являється в середині рівня, де він подається гравцю у якості боса, а далі двоє у кінці, у якості перевірки гравця.



Рисунок 4.18 – Ворог ліч

Важливо при плануванні балансу та ворогів, які будуть знаходитися у хвилях планувати кількість та тип ворогів які будуть його супроводжувати.

4.3.4 Зомбі

Зомбі – це слабкі, але масові противники, у яких є великий спектр застосування. (див. рис. 4.19).

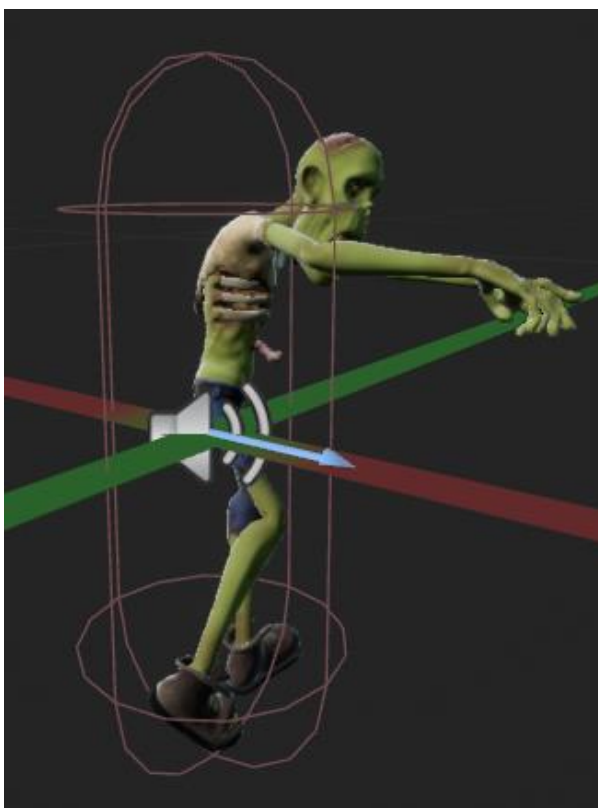


Рисунок 4.19 – Ворог зомбі

Їх головна мета в тому, щоб виступити в якості хвилі-перепочинку. Це слугує в деякій мірі елегантним рішенням для створення хвилі. З одного боку це дасть гравцю спосіб перепочити, відчути силу зачистивши велику кількість слабких ворогів, отримати багато золота, а також використати цих ворогів для того аби мотивувати гравця побудувати інший тип веж, адже вежі аури чудово підходять аби справитися з цими ворогами, в той час як його конфігурація з іншими вежами може не впоратися з цим. Це також слугує в деякій мірі покаранням за те, що гравець сконцентрувався на вежах які наносять велику кількість шкоди в одну ціль, але це не має значення якщо вони просто не встигатимуть вбивати ворогів.

5 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Обґрунтування вибору тестування

Розроблена система це складний програмний продукт. Тим паче враховуючи що це ігровий продукт, це означає що в ньому обов'язково знайдуться баги, які будуть псувати враження та геймплей для гравця. Саме для цього необхідним етапом при розробці гри є її подальше тестування. Надійність цього етапу розробки не варто недооцінювати, адже якщо проблеми у грі будуть накопичуватися, то гравцю буде набагато важче погрузитися в ігровий процес, натомість його увагу будуть постійно відволікати, в кращому випадку, дрібні баги та візуальні артефакти, а в гіршому, він не зможе навіть продовжувати гру через критичні баги. Саме тому необхідно провести функціональне тестування системи, і у випадку наявності проблем, опрацювати їх.

Для того аби провести повне тестування розробленої гри необхідно провести функціональне тестування її двох елементів: Інтерфейс та геймплей.

5.2 Вимоги до тестування інтерфейсу

Головне вимоги які ставляться до інтерфейсу – це бути зрозумілим та відзивним, аби гравець розумів що на будь-яку його дію є реакція і ця реакція має бути змістовна та інтуїтивно зрозуміла. Аби протестувати всі елементи інтерфейсу в результаті нам необхідно:

- Перевірити натискання на елементи інтерфейсу (кнопки в інтерфейсі головного меню, меню будівлі та покращення веж, меню паузи, перемоги та програшу).
- Перевірити коректність роботи елементів інтерфейсу які мають логіку при наведенні миші.
- Перевірити, чи весь текст можливо прочитати (розмір, колір, розташування тексту).
- Перевірити чи змінюється характеристики вежі при покращенні в реальному часі (При наведенні на кнопку покращення з'являється

порівняння у наступному вигляді «поточні хар-ки – покращені хар-ки». При натисканні кнопки, інтерфейс відразу оновлює значення).

- Перевірити час відгуку взаємодії користувача з інтерфейсом.

В результаті якщо всі вищевказані перевірки пройдуть успішно, то це гарантуватиме гравцю приємну взаємодію з UI частиною гри.

5.3 Вимоги до тестування геймплею

Головні вимоги які ставляться до геймплейних елементів гри – їх коректність та відповідність до вже задуманих гейм-дизайнером. Для тестування кожної механіки необхідно вручну провести тестування всіх можливих взаємодій у грі. До них належать:

- Можливість обрати та розпочати рівень у головному меню. При виборі рівня гра має переходити з головного меню до рівня.
- Можливість будувати вежі на виділених для цього місцях. Під час будовання, при наведенні на тип вежі має з'являтися привид вежі, яка буде побудована.
- Можливість покращити рівень вежі за золото. При покращенні вежі вона має візуально змінюватися. Для кожного рівня різні візуальні зміни.
- Можливість знищувати побудовані вежі. Після знищення вежі на звільненому місці має бути можливість заново коректно побудувати нову вежу.
- Можливість зміни режиму стрільби вежі.
- Можливість заробляти золото за вбивство ворогів. Кількість золота має не лише збільшуватись візуально у віджеті, а й фактично відповідати своєму значенню. При відсутності золота не мають будуватися, або покращуватися вежі.
- Можливість розпочати «Хвилю». При початку хвилі іншу розпочати не можна. Перевірити, що інші хвилі розпочинаються лише у випадку коли всі вороги знищені, або дійшли до бази, щоб в результаті в ігровому світі на шляху не було ворогів.

- Можливість рухати камеру за допомогою клавіш WASD. Також можливість рухати камеру у діагональних напрямках.

В результаті, якщо всі тести будуть пройдені, це буде гарантувати нам, що гравець не стикнеться з критичними та навіть звичайними багами у процесі своєї гри.

5.4 Результати тестування

Тестування проводилось шляхом повного перебору та тестуванням усіх функцій вручну. Хоч це здається часозатратним, але те, як гра спроектована, спрощує задачу, адже гра не генерує динамічного контенту, який не можна буде передбачити. Все, що тестер у грі може зробити, може бути легко та точно відтворено і як наслідок задокументовано та виправлено. Далі наведена частина баг-репортів різної тяжкості та типу які були знайдені за час тестування гри, котрі наведені у таблицях нижче (див. табл 1.1 – 1.7).

Таблиця 1.1 – Баг-репорт №1

Назва багу	Баг відображення інтервалу пострілу
Опис	Некоректне відображення значення яке відповідає за інтервал між пострілами вежі
Тип	UI
Серйозність та пріоритет	S4/P3
Кроки для відтворення багу	Розпочати рівень – побудувати вежу – натиснути на вежу.
Фактичний результат	Значення десяткового числа відображаються повністю.
Очікуваний результат	Значення десяткового числа відображаються до 2 значень після коми.

У даному баг-репорті, викрилась помилка з некоректним виведенням значень при відображенні не цілого числа, а десяткового, який може приймати значення, які в свою чергу будуть мати багато чисел після коми, що псуватиме єдиний стиль інтерфейсу.

Таблиця 1.2 – Баг-репорт №2

Назва багу	Баг відображення інтервалу пострілу при покращенні вежі.
Опис	Некоректне відображення значення яке відповідає за інтервал між пострілами вежі при виведенні інформації при покращенні вежі.
Тип	UI
Серйозність та пріоритет	S4/P3
Кроки для відтворення багу	Розпочати рівень – побудувати вежу – натиснути на вежу – навести курсор на кнопку покращення.
Фактичний результат	Значення десяткового числа відображаються повністю.
Очікуваний результат	Значення десяткового числа відображаються до 2 значень після коми.

У даному баг-репорті, викрилась помилка з некоректним виведенням значень при відображенні не цілого числа, а десяткового, який може приймати значення, які в свою чергу будуть мати багато чисел після коми, що псуватиме єдиний стиль інтерфейсу. У даному випадку це відбувалось при наведенні на кнопку покращення вежі, адже за прорахунок цих значень у кодї гри відповідали різні функції.

Таблиця 1.3 – Баг-репорт №3

Назва багу	Баг провалення ворогів під текстури.
Опис	При старті хвилі, вороги йдучи своїм шляхом провалюються під текстури. Виникає лише за умови коли текстура йде вверх, уворюючи невелику гірку.
Тип	Gameplay
Серйозність та пріоритет	S2/P1
Кроки для відтворення багу	Розпочати рівень – запустити хвилю.
Фактичний результат	Вороги йдучи своїм шляхом опиняються під текстурою.
Очікуваний результат	Вороги повинні рухатись поверх текстури створюючи ілюзію ходьби по маршруту.

У даному баг-репорті викрилась критична помилка, яка виникла через те, що компонент який представляв з себе криву, був накладений з недостатньою точністю на ландшафт, що в свою чергу змушувало ворогів які йдуть ним провалюватися.

Таблиця 1.4 – Баг-репорт №4

Назва багу	Баг відсутності снарядів у вежі.
Опис	Стріляючи по ворогам, вежа чекає доки активний снаряд зіштовхнеться з ворогом, або зникне, щоб зробити новий постріл.
Тип	Gameplay
Серйозність та пріоритет	S1/P1
Кроки для відтворення багу	Розпочати рівень – побудувати вежу - запустити хвилю.
Фактичний результат	Вежа чекає доки її активний снаряд зіткнеться з ворогом.
Очікуваний результат	Вежа повинна вистрілити наступним снарядом, не чекаючи повернення попереднього.

У даному баг-репорті викрилась помилка з налаштуванням у веж розміру object-pool'a. Саме через це некоректне значення у вежах був лише один об'єкт який вони використовували у якості снарядів і чекали наступний.

Таблиця 1.5 – Баг-репорт №5

Назва багу	Баг будівлі веж
Опис	Нові вежі після побудови першої (окремо для кожного типу) не з'являються у вигляді привида і не будуються.
Тип	Gameplay
Серйозність та пріоритет	S1/P1
Кроки для відтворення багу	Розпочати рівень - натиснути на місце побудови вежі – побудувати вежу – закрити інтерфейс – натиснути на інше місце побудови вежі – навестись на той самий тип вежі АБО натиснути на той самий тип вежі
Фактичний результат	Вежа не будується і не відображає привида
Очікуваний результат	Вежа повинна відображати привида та коректно побудуватись.

У даному баг-репорті викрилась помилка зі зчитуванням та розрахунками розміру object-pool'a веж. Саме через це у менеджері веж не було об'єктів які вони могли б використати у якості потенціальної вежі, або ж вежі привида яка з'являється при наведенні на елемент інтерфейсу..

Таблиця 1.6 – Баг-репорт №6

Назва багу	Баг відтворення звуку ворогів при періодичній шкоді
Опис	При побудові вежі яка наносить періодичну шкоду, вороги некоректно програють звук шкоди.
Тип	Gameplay
Серйозність та пріоритет	S3/P1
Кроки для відтворення багу	Розпочати рівень - натиснути на місце побудови вежі – побудувати вежу «Яду» – закрити інтерфейс – розпочати рівень
Фактичний результат	Під час заходу ворога в радіус дії башні, він програє звук отримання шкоди без затримки безліч раз за секунду
Очікуваний результат	Ворог повинен програвати звук з певним інтервалом, навіть якщо шкода наноситься дуже часто.

У даному баг-репорті викрилася помилка з некоректною логікою початку звуку, у даному випадку звук шкоди, ворога. Адже оскільки шкода яку наносила вежа обраховувалася кожний ігровий такт, триггер початку звуку також починався кожен раз.

Таблиця 1.7 – Баг-репорт №7

Назва багу	Баг морозної вежі
Опис	Якщо побудувати 2 морозні вежі, і ворог потрапляє в радіус дії обох, то він зупиняється
Тип	Gameplay
Серйозність та пріоритет	S1/P1
Кроки для відтворення багу	Розпочати рівень - натиснути на місце побудови вежі – побудувати вежу «Морозу» – закрити інтерфейс – натиснути на місце побудови вежі поруч – побудувати вежу «Морозу» – закрити інтерфейс – розпочати хвилю
Фактичний результат	Під час заходу ворога в радіус дії веж, ворог зупиняється повністю
Очікуваний результат	Ворог повинен мати обмеження на максимальне сповільнення, і продовжувати йти

У даному баг-репорті викрилась проблема, яка полягала в тому, що обрахунок швидкості ворога був не зовсім коректний, і в свою чергу якщо він складав у сумі 100% то ворог завмирав, не маючи мінімальної швидкості руху.

6 ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

6.1 Вибір сервісу для публікації гри

Для розміщення інди-ігор є безліч сервісів та платформ. У нашому випадку нам підійде модель у якій ми публікуємо гру у якомога більшій кількості сервісів. Оскільки наша гра це однокористувацький Tower Defence то він не потребує для роботи серверів та будь-якого супроводу чи адміністрування. Головна ціль – щоб про гру дізналось якомога більше людей та купило її. Цінник на цю гру варто виставляти не великий. Базуючись на вищевказаних потребах чудовими кандидатами стануть Steam та Epic Game Store. Обидва сервіси мають свої переваги і недоліки.

6.1.1 Steam

Steam — це найбільша цифрова платформа для розповсюдження відеоігор, яка належить компанії Valve Corporation. Вона була запущена в 2003 році і з того часу стала основним вибором для геймерів та розробників відеоігор. Steam пропонує широкий спектр функцій, включаючи онлайн-дистрибуцію, цифрове управління правами (DRM), соціальні мережі та сервіси для спільноти, автоматичні оновлення ігор та інші інструменти.

До переваг публікації гри на Steam можна віднести:

- Велика аудиторія. Steam має понад 120 мільйонів активних користувачів на місяць, що робить його найбільшою платформою для відеоігор у світі. Це забезпечує великий потенційний ринок для гри.
- Підтримка спільноти. Steam надає розробникам інструменти для взаємодії зі спільнотою гравців, включаючи форуми, відгуки та огляди, що може допомогти у покращенні гри та збільшенні її популярності.
- Маркетингові можливості. Steam пропонує різні інструменти для маркетингу, такі як участь у розпродажах, можливість демонструвати гру на головній сторінці, створення трейлерів та інших промо-матеріалів.

- Аналітика та звіти: Платформа надає детальні звіти про продажі, поведінку гравців та інші важливі метрики, що дозволяє розробникам краще розуміти свою аудиторію і адаптувати маркетингові стратегії.

До недоліків цієї платформи можна віднести наступне:

- Комісія Steam бере 30% від продажів, що може значно зменшити прибуток який ми отримуємо з продажі гри.
- Висока конкуренція. Через велику кількість ігор на платформі нові ігри можуть губитися серед великої кількості іншого контенту, що ускладнює завдання привернення уваги гравців.

Підсумовуючи вищесказане у висновку маємо, що Steam є чудовою платформою для публікації гри, особливо враховуючи велику аудиторію та різноманітні інструменти для маркетингу і взаємодії зі спільнотою. Проте, варто враховувати високий рівень конкуренції та значну комісію за продажі.

6.1.2 Epic Games Store

Epic Games Store (EGS) — це цифрова платформа для розповсюдження відеоігор, яка була запущена компанією Epic Games у 2018 році. Платформа швидко здобула популярність завдяки вигідним умовам для розробників та агресивній маркетинговій політиці.

До переваг публікації гри на цій платформі можна віднести:

- Комісія Epic Games Store бере лише 12% від продажів, що значно менше у порівнянні зі Steam. Це дозволяє нам отримувати більше прибутку від кожної проданої копії гри.
- Безкоштовне розповсюдження для Unreal Engine за умови, що гра розроблена на рушії Unreal Engine, Epic не бере комісії за використання рушія, що ще більше знижує витрати.
- Менша конкуренція. Платформа має меншу кількість ігор у порівнянні зі Steam, що може допомогти нашій грі привернути більше уваги.

- Маркетингова політика Epic games активно підтримує розробників через рекламні акції, безкоштовні роздачі ігор та інші маркетингові кампанії, що допомагає збільшити видимість гри.

В свою чергу до негативних моментів цієї платформи можна віднести наступне:

- Мала аудиторія платформи. Хоча платформа швидко зростає, вона все ще має меншу аудиторію порівняно зі Steam, що може обмежити потенційний ринок.
- Обмежені функції спільноти. EGS поки що не надає таких розширених функцій для взаємодії зі спільнотою, як Steam, що може ускладнити отримання зворотного зв'язку від гравців.
- Відсутність інструментів для модифікацій. На відміну від Steam, EGS поки що не пропонує підтримку модифікацій ігор, що може бути важливим для деяких жанрів ігрових продуктів.

На основі вищезазначених пунктів можна дійти висновку, що Epic Games Store є вигідною платформою для публікації, особливо для розробників, які використовують Unreal Engine, завдяки низькій комісії та різним маркетинговим інструментам. Проте, менш розвинена інфраструктура для взаємодії зі спільнотою та менша аудиторія можуть бути суттєвими недоліками.

В результаті для нашої гри варто розглянути можливість публікації гри на обох платформах — Steam та Epic Games Store. Кожна з них має свої унікальні переваги і недоліки, які можуть доповнювати одна одну. Steam забезпечує доступ до великої аудиторії та різноманітні інструменти для взаємодії з гравцями, тоді як EGS пропонує вигідні фінансові умови та маркетингову підтримку. Публікація на обох платформах може допомогти максимізувати потенціал гри та забезпечити її успіх, отримуючи при цьому максимальний прибуток.

ВИСНОВКИ

Результатом роботи є ігровий застосунок у жанрі Tower Defence, який відповідає всім поставленим вимогам, надає гравцю тривалий позитивний досвід. Система складається з мобільного та комп'ютерних додатків.

В процесі розробки було проведено аналіз предметної області, а також інших популярних ігор в цьому жанрі. В результаті виявлені позитивні та негативні аспекти цих додатків були враховані при проектуванні власного ігрового застосунку, на основі цього було поставлено задачу та сформовано основні вимоги для розробки програмного продукту. Також були детально пропрацьовані головні моменти які пов'язані з даним ігровим жанром, а саме стилістика гри, основні механіки, баланс, різноманіття ігрових ситуацій для гравця.

Після аналізу та формування задачі було спроектовано та розроблено архітектуру системи. Результатом цього стала детальна діаграма прецедентів, у якій було визначено функціональні вимоги, розуміння взаємодії системи з користувачем, визначення границь системи.

Програмна система була протестована сучасними методами тестування. Це один із головних аспектів кінцевої фази розробки, адже недостатньо готовий програмний продукт, не зважаючи на його потенціал, не розкриє себе в достатній мірі для гравця, а в гіршому випадку це віддтогне його та сформує негативне враження про цей жанр, знищить інтерес гравця до цього жанру.

Гра також має великий простір для подальшого покращення у різних напрямках. В залежності від вектору розробки гру можна розвивати у багатьох різних напрямках в залежності від поставлених цілей. Починаючи з розширення кількості ворогів та веж для урізноманітнення геймплею, закінчуючи введенням різних специфічних механік під кожен рівень поглиблюючи стратегічну глибину.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Why is the Tower Defence genre so stale? [Електронний ресурс] – Режим доступу до ресурсу: https://www.reddit.com/r/truegaming/comments/bxz63v/why_is_the_tower_defense_genre_so_stale/ (дата звернення: 5.05.2024).
2. What makes TD games boring or repetative? [Електронний ресурс] – Режим доступу до ресурсу: <https://devforum.roblox.com/t/what-makes-td-games-boring-or-repetitive/2807796> (дата звернення: 5.05.2024).
3. Game balance of the Tower Defense genre on the case of Crush Link TD [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@gabvanfet/game-balance-of-the-tower-defense-genre-on-the-case-of-crush-link-td-a00bac3d0c00> (дата звернення: 5.05.2024).
4. Price to pay: The damage caused by Pay-to-win games [Електронний ресурс] – режим доступу: <https://flavio-lee.medium.com/price-to-pay-the-damage-caused-by-pay-to-win-games-74cc533bf9d8> (дата звернення: 5.05.2024).
5. What makes a good tower defence game [Електронний ресурс] – режим доступу до ресурсу: https://www.reddit.com/r/gamedesign/comments/lwxjr0/what_makes_a_good_tower_defense_game/ (дата звернення: 5.05.2024).
6. Accessible games for everyone [Електронний ресурс] – доступ до ресурсу: <https://userway.org/blog/accessible-games/> (дата звернення: 5.05.2024).
7. Review Kingdom Rush [Електронний ресурс] – доступ до ресурсу: <https://www.lifeisxbox.eu/review-kingdom-rush/>
8. A honest review of Infintode 2 [Електронний ресурс] – доступ до ресурсу: <https://www.taptap.io/post/4644706>
9. Object Pool Design pattern in Unreal Engine [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.epicgames.com/community/learning/tutorials/Eylj/object-pool-design-pattern-in-unreal-engine> (дата звернення: 5.05.2024).
10. How to Move a Character Along a Spline in Unreal Engine 5 [Електронний

ресурс] – URL: <https://dev.epicgames.com/community/learning/tutorials/Za2a/how-to-move-a-character-along-a-spline-in-unreal-engine-5> (дата звернення: 5.05.2024).

11. Answers HQ [Електронний ресурс] – URL: <https://docs.unrealengine.com/4.26/en-US/BlueprintAPI/DataTable/> (дата звернення: 5.05.2024).

ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016310727

Дата перевірки:
02.06.2024 14:13:14 EEST

Тип перевірки:
Doc vs Library

Дата звіту:
02.06.2024 14:25:21 EEST

ID користувача:
100012353

Назва документа: 2024_Б_ПІ_ПЗПІ-20-2_Гребінченко_М_В_скорочений

Кількість сторінок: 70 Кількість слів: 9217 Кількість символів: 68769 Розмір файлу: 2.53 MB ID файлу: 1016107495

3.87% Схожість

Найбільша схожість: 1.17% з джерелом з Бібліотеки (ID файлу: 1008278791)

Пошук збігів з Інтернетом не проводився

3.87% Джерела з Бібліотеки

183

Сторінка 72

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи

9

ДОДАТОК Б

Лістинг програмного коду

```
#pragma once

#include "CoreMinimal.h"
#include "TowerBase.h"
#include "PoisonTower.generated.h"

UCLASS()

class TDGAME_API APoisonTower : public ATowerBase
{
    GENERATED_BODY()

public:
    ATower();

protected:
    virtual void BeginPlay() override;

public:
    virtual void Tick(float DeltaTime) override
    UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "Tower")
    float Range;
    UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "Tower")
    float Damage;
    UFUNCTION(BlueprintCallable, Category = "Tower")
    void DealDamage();

private:
    void GetOverlappingEnemies(TArray<AActor*>& OutEnemies);

};

#include "PoisonTower.h"
#include "Kismet/GameplayStatics.h"
#include "GameFramework/Actor.h"
```

```

#include "Engine/World.h"

#include "Components/SphereComponent.h"

#include "GameFramework/Pawn.h"

#include "GameFramework/DamageType.h"

APoisonTower::ATower()
{
    PrimaryActorTick.bCanEverTick = true;

    Range = 1000.0f; // Example range, can be set in the editor
    Damage = 10.0f; // Example damage, can be set in the editor
}

void APoisonTower::BeginPlay()
{
    Super::BeginPlay();
}

void APoisonTower::Tick(float DeltaTime)
{
    Super::Tick(DeltaTime);

    DealDamage();
}

void APoisonTower::DealDamage()
{
    TArray<AActor*> OverlappingEnemies;

    GetOverlappingEnemies(OverlappingEnemies);

    for (AActor* Enemy : OverlappingEnemies)
    {
        if (Enemy)
        {
            UGameplayStatics::ApplyDamage(Enemy, Damage * GetWorld()-
>DeltaTimeSeconds, GetInstigatorController(), this,
UDamageType::StaticClass());
        }
    }
}

```

```

    }
}
}
void APoisonTower::GetOverlappingEnemies(TArray<AActor*>& OutEnemies)
{
    FVector Location = GetActorLocation();
    TArray<FOverlapResult> OverlapResults;
    FCollisionShape CollisionShape = FCollisionShape::MakeSphere(Range);
    bool bOverlap = GetWorld()->OverlapMultiByChannel(
        OverlapResults,
        Location,
        FQuat::Identity,
        ECC_Pawn,
        CollisionShape,
        FCollisionQueryParams(FName(TEXT("TowerOverlap")), false, this)
    );
    if (bOverlap)
    {
        for (auto& Result : OverlapResults)
        {
            AActor* OverlappedActor = Result.GetActor();

            if (OverlappedActor && OverlappedActor-
>IsA(APawn::StaticClass()))
            {
                OutEnemies.Add(OverlappedActor);
            }
        }
    }
}
}


```

ДОДАТОК В

Слайди презентації

Харківський національний університет радіоелектроніки
Кваліфікаційна робота бакалавра

Ігровий програмний застосунок у жанрі Tower Defence



Виконав: ст. гр. ПЗПІ-20-2 Гребінченко М.В.
Керівник: ас. кафедри ПІ: Матвеев Д.І.

Актуальність

Global Mobile Tower Defense Games Market

Year	Revenue
2021	US\$ 937.62 Mn
2030	US\$ 2,311.27 Mn

2022-2030 CAGR: 10.38%

Driver

- Easy to learn and sense of accomplishment attracting millennials

Trend

- Growing demand of 3D games.

Restraint

- Legal regulation and market saturation



Tower Defense Mobile Game Market

Segment by Application - Android System, IOS System
Segment by Type - Free to Play, Pay to Play
By Company - Supercell, Niantic, Tencent, Machine Zone(MZ), Netmarble, King, EA Mobile, Mixi, GungHo



M = millions and B = billions | Valuates Reports*

Market Segmentation

By Product	By Application	By Age Group	By Region
<ul style="list-style-type: none"> Pay to Play Free to Play 	<ul style="list-style-type: none"> Android IOS Others 	<ul style="list-style-type: none"> < 18 Years 18-35 Years 35-60 Years > 60 Years 	<ul style="list-style-type: none"> North America Europe Asia Pacific South America Middle East & Africa

Мета роботи

- Аналіз ігор-конкурентів у жанрі Tower Defence.
- Забезпечити розвагу гравця шляхом залучення використовуючи цікаві механіки, складні завдання та стратегії.
- Дослідити можливості нового ігрового двигуна.
- Побудова спільноти гравців, які будуть об'єднані навколо гри.
- Отримати комерційний прибуток через продажі гри.

Робота з даними. Аналіз конкурентів

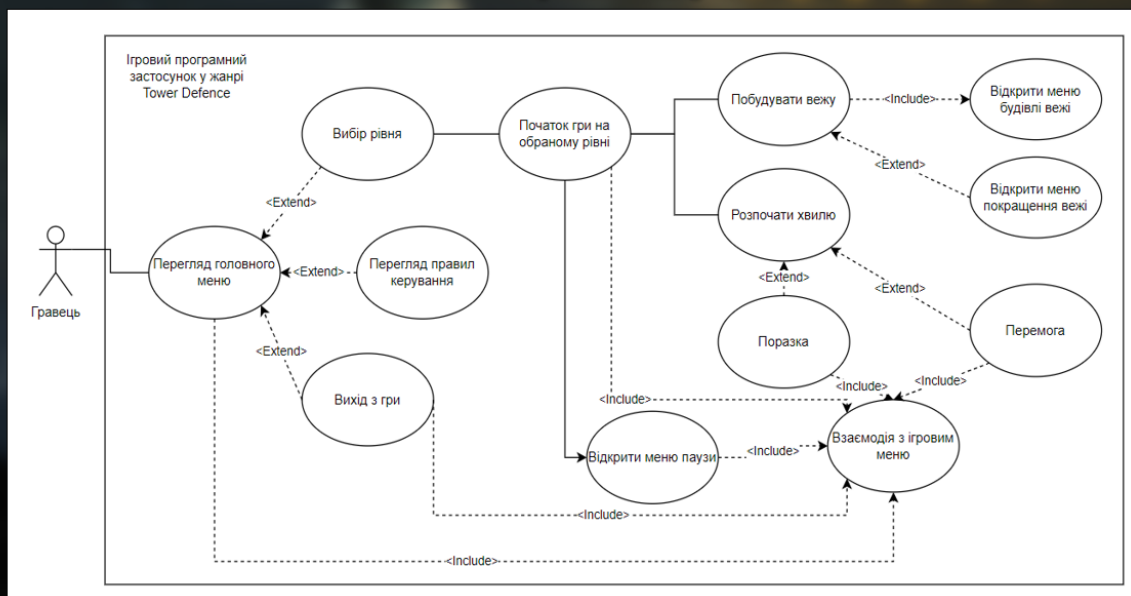


Постановка задачі

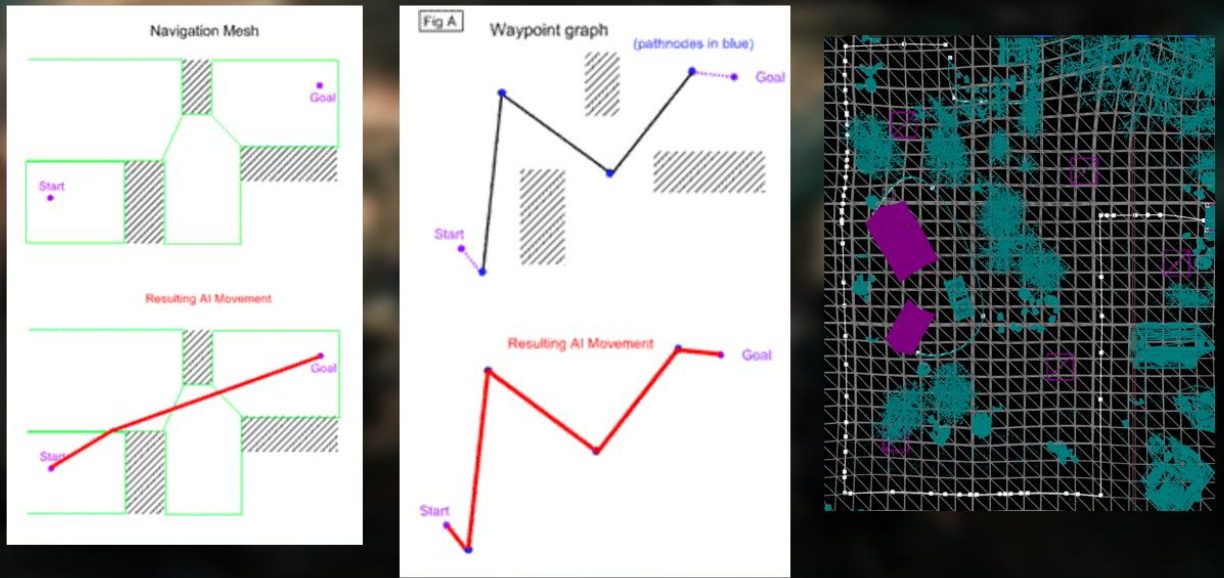
Необхідно створити гру у жанрі Tower Defence. Гра повинна відповідати наступним критеріям:

- Гра повинна бути доступна на більшості сучасних платформ, Windows та Android.
- Гра має бути оптимізованою для роботи на різних пристроях з різними технічними характеристиками.
- Гра повинна мати зрозумілий і простий у використанні інтерфейс.
- Прогрес повинен стимулювати гравців до подальшої гри.

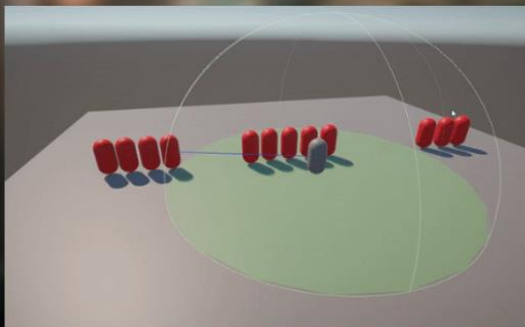
Use-Case проектування



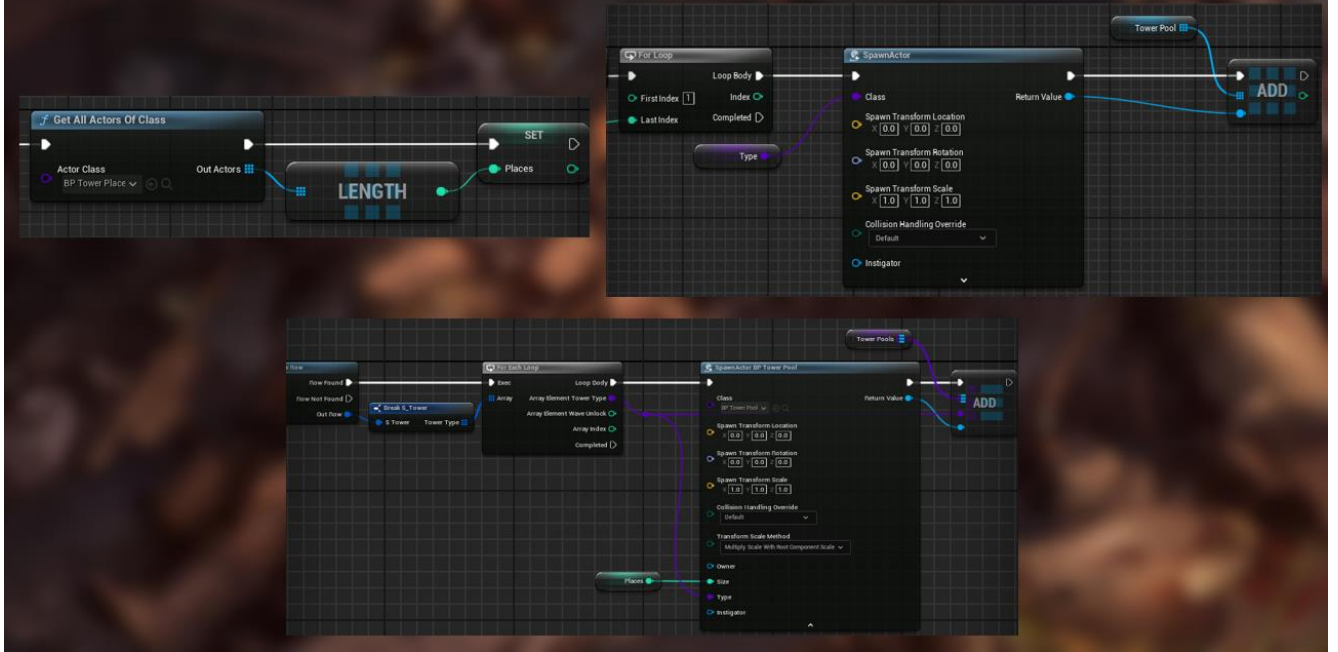
Реалізація ігрових механік: Рух ворогів



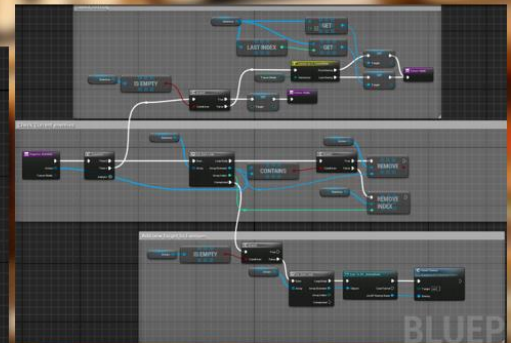
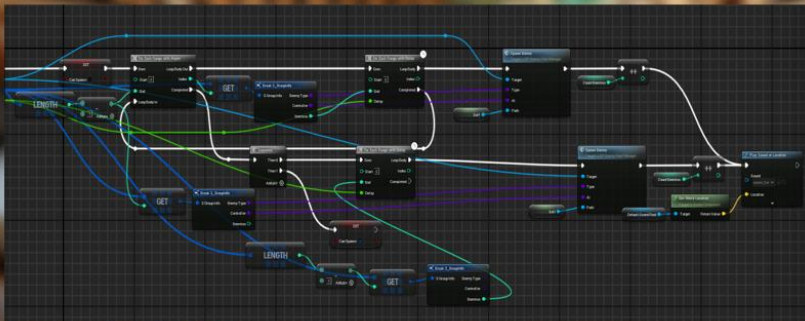
Реалізація ігрових механік: Вежі та вибір цілей



Архітектурні рішення: Об'єкт пул веж



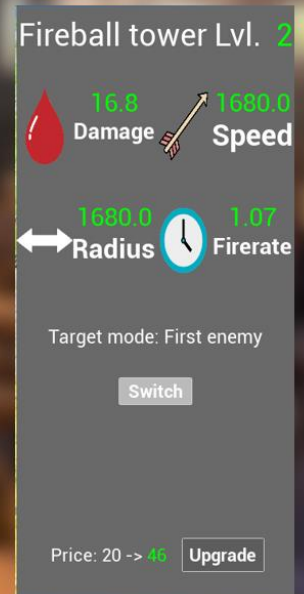
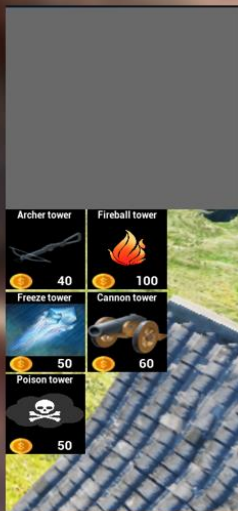
Програмне рішення: Спавн ворогів та вибір їх вежами



Интерфейс головного меню



Интерфейс гри



Висновки

- Проведено аналіз ігор конкурентів у жанрі Tower Defence
- Спроековано діаграму прецедентів гри.
- Розроблено та втілено архітектурні підходи до розробки ігрового застосунку.
- Програмно реалізовано та протестовано ігровий додаток.

Перейдемо до презентації