

ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Розробка гри у жанрі Tower Defense з сетінгом в  
воєнному стилі мовою C# з використанням рушія Unity»

на здобуття освітнього ступеня бакалавра  
зі спеціальності 121 Інженерія програмного забезпечення  
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень. Використання  
ідей, результатів і текстів інших авторів мають посилання  
на відповідне джерело*

\_\_\_\_\_ (підпис)

Олександр КОВАЛЕНКО

Виконав: здобувач вищої освіти групи ПД-43

Олександр КОВАЛЕНКО

Керівник: Олесь ДІБРІВНИЙ  
доктор філософії (PhD)

Рецензент: \_\_\_\_\_

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ Ірина ЗАМРІЙ

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

\_\_\_\_\_ Коваленку Олександрю Миколайовичу \_\_\_\_\_

1. Тема кваліфікаційної роботи: «Розробка гри у жанрі Tower Defense з сетінгом в воєнному стилі мовою С# з використанням рушія Unity»

керівник кваліфікаційної роботи доктор філософії, доцент кафедри ІІЗ Олесь ДІБРІВНИЙ,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36.

2. Строк подання кваліфікаційної роботи «28» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: Існуючі методи створення ігор, технічна література, документація до ігрового двигуна Unity та мови програмування С#.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Розвиток ігрової індустрії
2. Існуючі жанри ігор
3. Інструменти розробки ігрових додатків
4. Основні вимоги та тестування гри

5. Перелік використаних технологій
6. Опис функціональних можливостей гри

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів
2. Вимоги до ігрового застосунку
3. Концепт гри
4. Програмні засоби та інструменти реалізації
5. Use Case діаграма користувача
6. Діаграма класів
7. Загальний алгоритм гри
8. Екранні форми
9. Результати роботи гри
10. Апробація результатів дослідження

6. Дата видачі завдання «28» лютого 2024 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	28.02.-06.03.2024	
2	Аналіз та дослідження існуючих аналогів	07.03-13.03.2024	
3	Підбір інструментів для розробки ігрового застосунку	14.03-20.03.2024	
4	Проектування архітектури	21.03-29.03.2024	
5	Програмна реалізація застосунку	30.03-15.04.2024	
6	Тестування застосунку користувачами	16.04-25.04.2024	
7	Оформлення роботи: вступ, висновки, реферат	29.04-05.05.2024	
8	Розробка демонстраційних матеріалів	06.05-12.05.2024	
9	Попередній захист роботи	13.05-31.05.2024	

Здобувач вищої освіти

\_\_\_\_\_ (підпис)

Олександр КОВАЛЕНКО

Керівник

кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Олесь ДІБРІВНИЙ





## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 54 стор., 40 рис., 20 джерел.

*Мета роботи* – адаптація ігрового процесу в грі жанру Tower Defense.

*Об'єкт дослідження* – процес гри в жанрі Tower Defense.

*Предмет дослідження* – ігровий застосунок в жанрі Tower Defense.

*Короткий зміст роботи:* У представленій роботі проведено аналіз існуючих ігор, таких як Kingdom Rush, Deathtrap, Revenge Of The Titans, Prime World: Defenders 2, а також функціональних можливостей цих ігор. Загальною проблемою зазначених ігор є постійна необхідність підключення до інтернету, що може стати великою незручністю для користувачів. У випадку відсутності інтернет з'єднання користувач не зможе використовувати застосунок. Ще одним суттєвим недоліком зазначених проєктів є складність для новачків при першому знайомстві з жанром.

Особливістю розробленої гри є можливість грати без підключення до інтернету на будь-якому пристрої, оскільки вона гарно відтворюється на звичайному персональному комп'ютері. Крім того, гра відзначається простотою проходження рівнів, що робить її привабливою для новачків у жанрі Tower Defense. Гра була створена за допомогою ігрового двигуна Unity, мовою програмування C#, ці засоби забезпечують відмінну взаємодію між собою.

Збереження всіх видів даних здійснюється за допомогою файлів гри, які містять всю необхідну інформацію.

Сферою використання застосунку є новачки, котрі бажають ознайомитись з жанром ігор Tower Defense.

**КЛЮЧОВІ СЛОВА:** ГРА DEATHTRAP, ЗАХИСНА СПОРУДА, TOWER DEFENSE, ПРОТИВНИК, ГРА REVENGE OF THE TITANS

## ЗМІСТ

ВСТУП.....	9
1. ОГЛЯД ІСНУЮЧИХ ПРОДУКТІВ В ЖАНРІ TOWER DEFENSE .....	10
1.1 Жанри ігор.....	10
1.2 Експансія індустрії та її важливість у сучасному світі .....	14
1.3 Змагання у сфері відеоігор, як нова фаза в еволюції галузі.....	20
1.4 Інструментарій для створення ігор.....	22
1.5 Створення ігрових додатків за допомогою рушія Unity .....	26
1.5.1 Ігровий рушій Unity .....	26
1.5.2 Можливості програмної платформи .....	27
1.5.3 Переваги та недоліки середовища розробки Unity.....	30
1.5.4 Створення ігор за допомогою мови програмування C#.....	31
1.6 Огляд аналогів .....	32
1.7 Завдання для дослідження.....	41
2. РОЗРОБКА СТРУКТУРНОЇ СКЛАДОВОЇ ГРИ .....	42
2.1 Мета розробки .....	42
2.2 Розробка програмного об'єкту.....	42
2.2.1 Діаграма прецедентів.....	42
2.2.2 Організація даних у проекті.....	45
3.1 Використані засоби у процесі створення ігрового додатку.....	48
3.2 Оцінка, планування та управління процесом створення гри.....	50
3.3 Функціональні можливості гри.....	52
4. ДЕМОНСТРАЦІЯ ЗАСТОСУВАННЯ ТА ТЕСТУВАННЯ .....	54
4.1 Характеристика користувацького інтерфейсу і процесу гри.....	54
4.2 Наступні етапи розвитку гри.....	57
ВИСНОВКИ.....	59
ПЕРЕЛІК ПОСИЛАНЬ .....	61
ДОДАТОК А.....	64
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ .....	64
ДОДАТОК Б .....	73
ЗАГАЛЬНИЙ АЛГОРИТМ ГРИ.....	73

## ВСТУП

*Обґрунтування вибору теми та її актуальність:* З розширенням ігрової культури зростає і потреба у створенні ігор різних жанрів для задоволення усіх потреб користувачів. Однак, незважаючи на це, наразі не в кожному жанрі можна знайти ігри, що відповідають високим стандартам якості та надають повноту емоційного досвіду. Тому пошук та створення гідних представників усіх жанрів є нагальною задачею для розвитку індустрії геймінгу.

Для покращення ігрового досвіду користувача потрібно розробити функціонал, конкуруючий з вже існуючими аналогами на ринку ігор.

*Ступінь вивчення проблеми:* На сьогодні існує велика кількість продуктів у жанрі Tower Defense, але не всі вони досягають достатньо високого рівня, щоб задовольнити всіх користувачів своїм функціоналом і простотою використання. Головне завдання будь-якого проекту полягає в тому, щоб привернути якнайбільше користувачів. Деякі з найвідоміших ігор у цьому жанрі, такі як Prime World: Defenders 2, Deathtrap, Kingdom Rush та інші, мають свої унікальні стилі, які можуть не зацікавити всіх гравців. Навіть серед таких популярних ігор іноді виникають проблеми, які перешкоджають розширенню ігрової спільноти в цьому жанрі. Наприклад, деякі з цих проектів можуть не відповідати смакам групи користувачів.

Існуючі ігри, безумовно, допомагають користувачам розслабитися після напруженого дня, але не всі вони підходять для цієї мети. Тому важливо розвивати галузь ігор, випускаючи нові, прості і розважальні продукти.

*Об'єктом дослідження* є роль та алгоритми, які впливають на ігровий процес у жанрі Tower Defense.

*Предметом роботи* є гра у жанрі Tower Defense, створена за допомогою Unity.

*Метою роботи* є розробка повноцінних функцій, що сприятимуть розвитку гри.



*Завданням роботи є створення першої технічно-демонстраційної версії гри, яка дозволить запуснути та вивчити її.*

*Методика дослідження* включала визначення необхідного функціоналу для правильної роботи гри та вибір стеку технологій, який забезпечив би конкурентоспроможний продукт з гарантією працездатності та можливістю подальшого оновлення та підтримки.

Розглянувши факт, що гра має працювати на операційній системі Windows, було обрано Unity для розробки з використанням мови програмування C#. Цей вибір забезпечує повноцінний та оптимізований продукт, здатний працювати ефективно на різних пристроях. Після успішного запуску на Windows, планується розробка версій для Android та iOS.

Результат розробки можна буде встановлювати традиційним шляхом через exe-файл.

*Наукова новизна* полягає у створенні проекту з повноцінним функціоналом, який забезпечить працездатність гри та унікальний стиль.

*Практичне значення результатів* полягає в можливості користування продуктом різними користувачами, що працюють на різних операційних системах, і бажають витратити час на задоволення своїх емоційних потреб у грі.

## 1. ОГЛЯД ІСНУЮЧИХ ПРОДУКТІВ В ЖАНРІ TOWER DEFENSE

### 1.1 Жанри ігор

Жанр відеогри визначається за її призначенням та функціональними можливостями, і може бути одним або кількома одночасно, навіть створюючи новий жанр у виняткових випадках. Існує вісім основних жанрів відеоігор, котрі поділяються на піджанри.

Жанр Action ігор - характеризується тим, що успіх користувача базується на розвиненості його реакції та здатності швидко реагувати на події у грі. Темп гри дуже динамічний, вимагаючи постійної уваги та швидких реакцій. Основним інструментом для досягнення прогресу в таких іграх є будь-яка зброя. Піджанри цього жанру включають аркади, шутери, ігри з елементом боротьби з противником. Цей жанр є найпопулярнішим і займає більшу частину ринку відеоігор.

3D-шутер - передбачає, що гравець, діючи самостійно або у команді з друзями, має знищувати ворогів за допомогою зброї ближнього бою і стрільби, щоб досягти цілей на різних рівнях та локаціях. Зазвичай після досягнення мети гравець отримує нові завдання та переходить на наступний рівень. Ворогами можуть бути бандити, інопланетні істоти, мутанти та інші, які зустрічаються у грі, такі як Call Of Duty, Far Cry, Cyberpunk 2077, Left 4 Dead.



Рис. 1.1 – 3D-шутер Call Of Duty

"Slashers" - це варіація ігор, дуже схожих на файтинги, але з фокусом на використанні холодної зброї. Зазвичай такі ігри розробляються для ігрових консолей, оскільки управління геймпадом більш зручне для гри такого типу - використовується не більше 10 кнопок, і їх комбінації не дозволяють використовувати звичайну клавіатуру та мишу.

Симулятори - це ігри, які моделюють керування різними процесами або транспортними засобами. Зазвичай термін "симулятор" використовується для комп'ютерних програм, які відтворюють інтер'єр кабіни апарату та дозволяють тренувати пілотів, космонавтів, машиністів поїздів. Симулятор - це комбінація програмного та апаратного забезпечення, що створює враження реальності, відображаючи реальні явища та властивості у віртуальному середовищі.

"Файтинг" - це жанр ігор, в яких основною механікою є бій на руках героїв в обмеженому просторі, так званій арені. Гравець зазвичай знаходиться на невеликій арені, не може залишати її межі, сам бій розроблений з кількох раундів з обмеженим часом. У грі також можуть бути відображені шкали для відображення життєво важливих показників гравців.

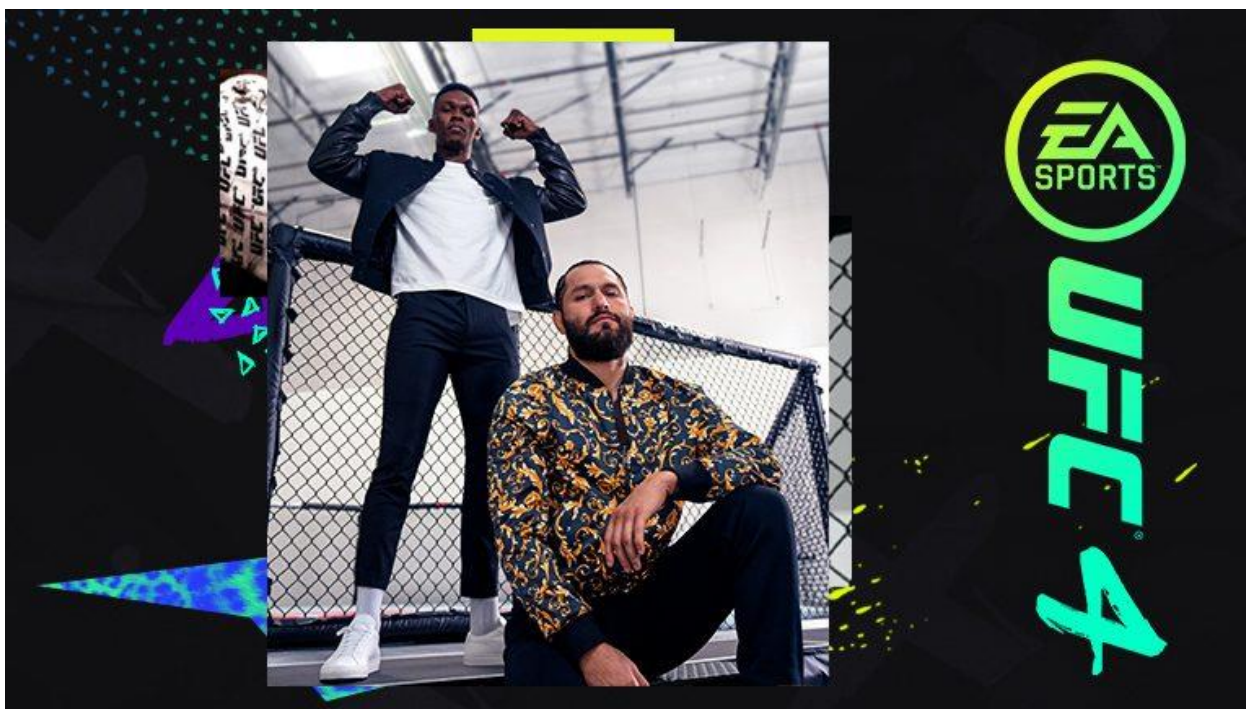


Рис. 1.2 – спортивний файтинг UFC 4

"Аркада" - це тип ігор, в яких гравцеві потрібно діяти швидко, використовуючи реакцію та особисті рефлексії. Ігровий процес простий і стабільний, часто в них присутня система бонусів, яка включає нарахування очок та поступове розблокування різних елементів гри. Термін "аркада" походить з епохи ігрових автоматів, які були поширені у торгових центрах та ігрових залах. Ці ігри були легкими для освоєння та привертання більшої кількості гравців, і згодом були портовані на ігрові консолі, залишаючись основним жанром для них.

Комп'ютерні рольові ігри, або RPG, є популярним жанром ігор, що базується на елементах ігрового процесу традиційних рольових ігор. У більшості випадків елементи RPG присутні у популярних іграх, щоб збільшити тривалість та цікавість гри. У рольовій грі гравець керує одним або декількома персонажами, кожен з яких

має свої характеристики, здібності та вміння, які можуть змінюватися протягом гри. Одним із важливих елементів геймплею є підвищення здібностей персонажів шляхом покращення параметрів та вивчення нових навичок.



Рис. 1.3 – Популярна бельгійська RPG гра Baldur`s Gate 3

Стратегічні ігри вимагають від гравця планування та розробки стратегії для досягнення певної мети, такої як перемога у військових операціях. Гравець керує не лише одним персонажем, але цілим підрозділом або групою бійців. Ці ігри поділяються на лінійні або пошагові, де гравці роблять свої ходи по черзі з обмеженим або необмеженим часом, та стратегічні ігри в реальному часі, де дії всіх гравців відбуваються одночасно.

Пригодницькі ігри - це ігри-розповіді, в яких герой рухається вздовж сюжету та взаємодіє з ігровим світом за допомогою різних предметів та спілкування з квестовими персонажами, приймаючи рішення щодо різних завдань.





Рис. 1.4 – Пригодницька гра Detroit: Become Human

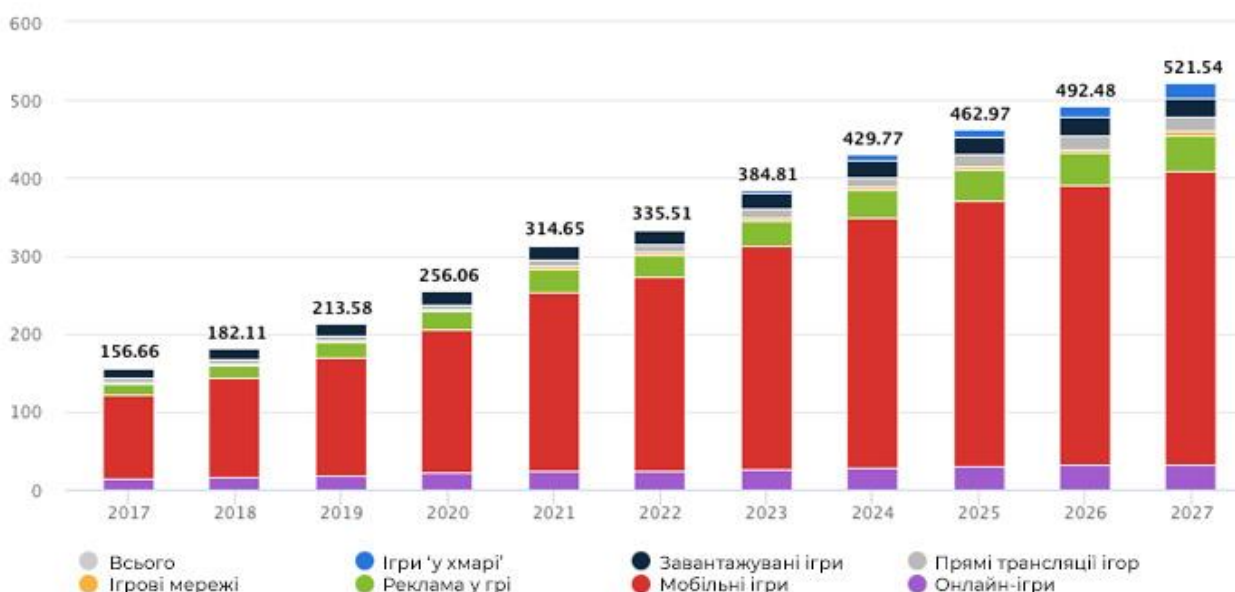
## 1.2 Експансія індустрії та її важливість у сучасному світі

У минулому ігрова індустрія не виникла як розвага, а скоріше як засіб досягнення наукових цілей. У перший період комп'ютери були великими і дорогими, тому використовувалися переважно в освітніх та наукових сферах. З появою доступних ігрових консолей та персональних комп'ютерів популярність ігор почала швидко зростати. Комерційний успіх привів до того, що розробники почали створювати ігри з метою заробітку. Сьогодні компанії, які займаються розробкою ігор, заробляють величезні кошти. У 2017 році доходи від мобільних ігор зросли до \$50,4 млрд, для ігор на персональних комп'ютерах — з \$29,4 до \$32,3 млрд, а для ігор на консолях — до \$33,3 млрд. Китай та Японія — лідери у зростанні доходів, оскільки вони мають найбільшу кількість гравців.

На (Рис. 1.5) можна спостерігати динаміку росту доходів індустрії ігор починаючи з 2017 року та закінчуючи прогнозами на дохід 2027 року.

# ДОХІД НА РИНКУ ВІДЕОІГОР

млрд дол



Джерело: Statista Market Insights

Рис. 1.5 – Загальний дохід ігрової індустрії у світі та прогноз на 2027 рр.

У 2023 році дохід геймінгової індустрії сягнув відмітки в 390 млрд дол. У 2017 році дохідність становила 156 млрд дол. Щорічно цей показник змінюється в діапазоні від 30 млрд дол. до 50 млрд дол.

Популярні видання прогнозують, що обсяг ринку до 2027 року зможе зрости до 521,6 млрд дол. Фанати різних видів ігор теж поповнюються шаленими темпами, цей показник може становити 3,5 млрд, що в 2,5 разу більше, ніж населення Індії.

Найбільш динамічним сегментом індустрії є мобільні ігри. Вони у 2023 році матимуть найбільшу частку геймінгового ринку – 286,5 млрд дол.

Світове лідерство відеоігрової індустрії як було зазначено належить Китаю. Очікується, що цього року її доходи в цій країні сягнуть 109 млрд дол.

Розвиток ігор в історії почався з винайдення перших комп'ютерних ігор ще у 1942 році. Однією з версій зачаткування ігрової індустрії вважається розробка ракетного симулятора для враження ворожих цілей. Аркадна гра була створена піонером американського телебачення Томасом Голдсмітом молодшим.

Протягом 50-60 років минулого століття почалася активна розробка перших ігор. Одним з таких прикладів є гра ОХО, що була описана в докторській дисертації А. С. Дугласа. Хоча відомості про неї, насамперед, були поширені серед студентської громади, ігри почали набирати популярність. У ті роки ігри розроблялися для різних типів комп'ютерів, які використовували текстові термінали та векторні графічні дисплеї.

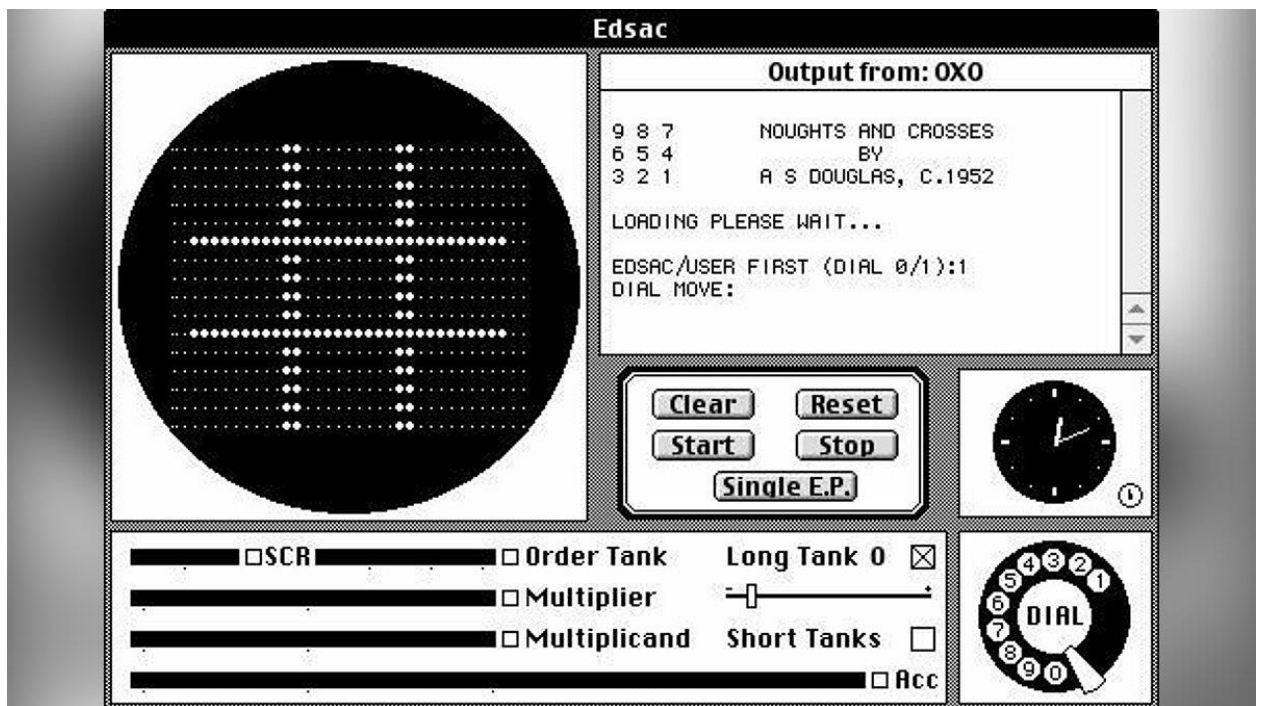


Рис. 1.6 – Гра ОХО (аналог гри Хрестики - Нулики)

У період між 1961 та 1970 роками виникла гра SpaceWar для комп'ютерів PDP-1, а також був створений перший прототип комп'ютерної миші. Ці інновації значно вплинули на подальший розвиток ігрової індустрії та комп'ютерних комплектуючих, які стали стандартом у сучасному світі.





Рис. 1.7 – Демонстрація першої комп'ютерної миші

Кілька років потому після цього періоду, відома компанія Atari випустила свою першу домашню консоль Atari 2600, що стала першою консоллю з тиражем в 1 мільйон проданих одиниць. Незабаром ринок ігор перенасичувався клонами гри "Понг", але у 1978 році гра Space Invaders принесла свіжий подих цьому сегменту. Ігрові автомати почали з'являтися всюди, а поява відомих франшиз, таких як Donkey Kong і Pac Man, ще більше збільшила популярність ігрової індустрії. Крім того, до 1982 року доходи від ігор перевищували прибутки від поп-музики та кінематографічних показів у кінотеатрах.

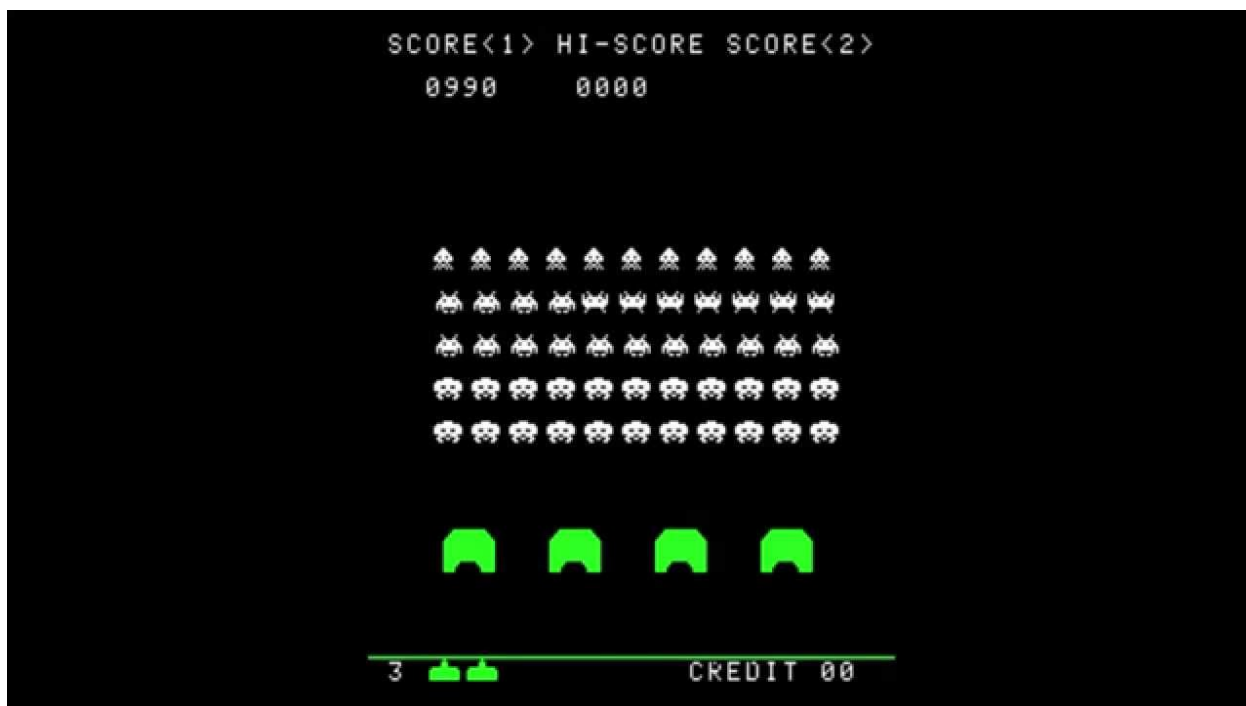


Рис. 1.8 – Відеогра для аркадних ігрових автоматів Space Invaders

Починаючи з 1981 по 1990 рік консолі втрачають свою популярність через доступність персональних комп'ютерів за більш низькою ціною. Це спричиняє початок розробки ігор з більш складною графікою, оскільки на ринку з'являються перші відеокарти, які полегшують навантаження на процесор. Важливими подіями цього періоду є 1983 рік, коли розпочалася ера 8-бітних консолей. Лідерами ринку стали відомі консолі, такі як Nintendo Entertainment System, Sega Master System, PC Engine, Atari 7800.

Також в цей час була представлена гра Final Fantasy, яка виходила у десятках нових частин і продовжує залишатися популярною й досі. Проте швидкий розвиток галузі призвів до проблем стійкості. Зброєю в рятуванні консольного ринку стала Nintendo, яка у 1985 році випустила консоль NES (або Famicom в Японії), а в СНГ вона була відома під назвою Dendy. Компанія висувала на передовий план високоякісні ігри та маркетинг. Такі ігри, як Duck Hunt, Excitebike і Super Mario, відновили інтерес до консольного геймінгу. У 1988 році Sega представила свою консоль Sega Mega Drive, спираючись на її потужність у парі.

Пізніше з'явилася компанія Sony, яка стала гігантом сучасності в галузі ігрових консолей. Її величезна історія почалася з випуску першої PlayStation у 1994 році. PlayStation відзначилася використанням CD-дисків замість картриджів, що дозволило збільшити обсяг пам'яті для деяких ігор. Це стало першим кроком до досягнення рекорду у продажах — перших 100 млн проданих консолей. Подальший розвиток індустрії отримав важливий вплив завдяки консолям PlayStation 2 (з DVD) і PlayStation 3 (з Blu-Ray).

Також варто відзначити внесок компанії Microsoft у геймінг з випуском DirectX API, яке значно спростило життя розробників ігор і дозволило зробити їх більш технологічними. Крім того, Microsoft випустила сервіс XBOX Live у 2001 році, що надавав доступ до гри через Інтернет і голосового чату за щомісячну плату, що стало обов'язковим для багатьох користувачів.

У той же період Blizzard випустила MMORPG World of Warcraft у 2004 році, яка стала піонером у використанні моделі "Щомісячна плата" для доступу до гри. На своєму піку популярності вона мала майже 14 мільйонів передплатників.



Рис. 1.9 – Онлайн гра World Of Warcraft в жанрі MMORPG

У подальшому провідні компанії розуміли потенціал портативного геймінгу і впроваджували його. Nintendo визначила своє місце з лінійкою Game Boy, тоді як Nokia і BlackBerry додавали прості ігри до своїх телефонів. Проте проривом стала компанія Apple, яка випустила магазин App Store, відкривши широкі можливості для розробників платних і так званих free-to-play ігор.

В даний момент Microsoft і Sony продовжують випускати нові консолі, розвиваючи паралельно сервіси підписки на ігри. У 2020 році кількість гравців у всьому світі перевищила 2,7 мільярда, і те, як вони витрачатимуть свої гроші, визначатиме майбутнє історії відеоігор в тому вигляді, в якому ми її бачимо зараз. На ринку існує багато гігантів індустрії зі своїми найпопулярнішими іграми, які стали настільки впливовими, що навіть стали основою для фільмів, як, для прикладу, Warcraft, Томб Рейдер, Resident Evil.

### **1.3 Змагання у сфері відеоігор, як нова фаза в еволюції галузі**

Розглянемо зміни в ігровій індустрії, де ігри з командною грою та рейтинговою системою вимагають більшого часу для досягнення результатів. Цей тренд особливо відчутний у мобільних іграх, які стають популярними у Китаї, інших країнах Азії та на Заході. Китайські видавці розширюють свій аудиторій на Захід завдяки успішності глобальних ігор, таких як PlayerUnknown's Battlegrounds. Також на Заході спостерігається підвищений інтерес до мобільних ігор жанру Battle Royale, що спричинений популярністю аналогічних ігор на ПК.

Найближчі роки визначать швидкість, з якою кіберспорт перетвориться на великий бізнес. Ключовими чинниками є успішність місцевих ліг, дотримання правил професійними гравцями, розвиток нових ігрових форматів та жорстка конкуренція. Рентабельність команди і вплив галузі на медійну сферу гри, зростання призових фондів і популярності різних дисциплін також важливі. Залежно від того, як ці фактори розвиватимуться, кіберспорт може стати мільярдним бізнесом протягом найближчого року чи двох, відповідно до різних



сценаріїв. Раніше прибутки від ігор порівнювали з кінотеатральними виручками. Однак з розвитком геймінгу, можна передбачити, що доходи від ігор перевищать кіноіндустрію в найближчі роки.

Призовий фонд чемпіонату світу «PGL Major Copenhagen» 2024 року з гри Counter Strike 2 склад 1 млн 250 тис. доларів, на якому українська кіберспортивна команда NAVI посіла перше місце та розділила грошову винагороду в 500 тис. доларів.

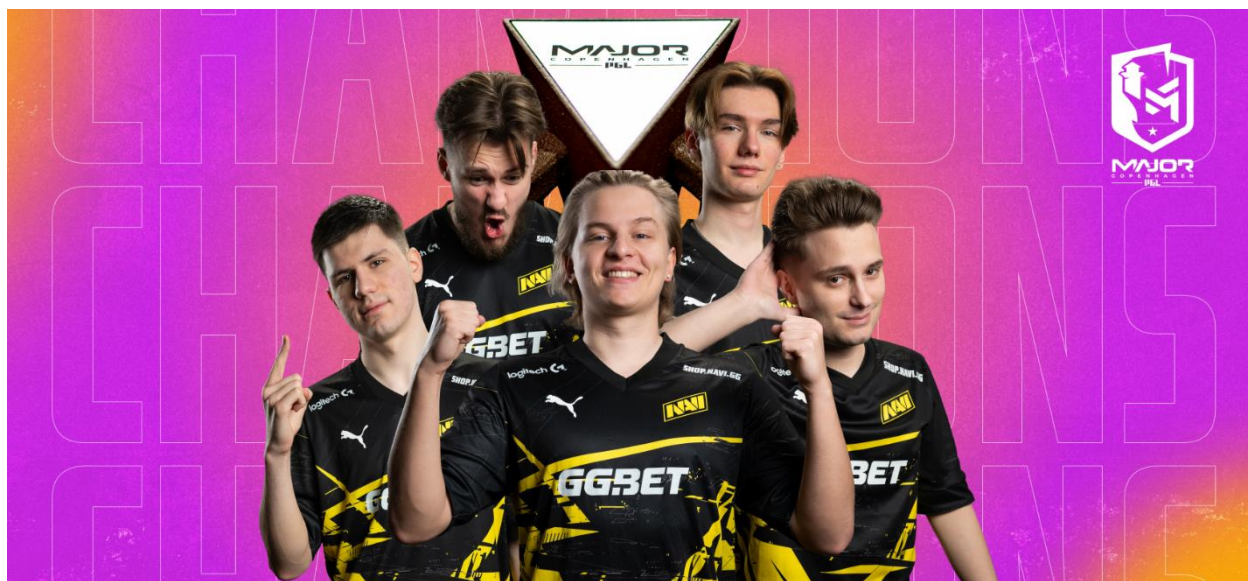


Рис. 1.10 – Переможці «PGL Major Copenhagen 2024» по Counter Strike 2

Різні джерела оцінюють світовий кіберспортивний бізнес від \$130 до \$150 мільярдів. Це включає доходи від реклами, спонсорства, продажу товарів та квитків на матчі. З такими темпами зростання, доходи від кіберспорту можуть перевершити вкладення в традиційний спорт. Додавши до цього прибутки від продажу консолей (близько \$10 мільярдів у 2017 році), комп'ютерів та периферійних пристроїв (приблизно \$23 мільярди), можна стверджувати, що кіберспорт стає значущим сектором сучасної економіки.

## **1.4 Інструментарій для створення ігор**

Для успішного створення проекту, який здобуде популярність серед мільйонної аудиторії, необхідна команда різних фахівців: програмісти, дизайнери, художники, аніматори, продюсери та інші. Їх участь в проекті охоплює всі етапи розробки. При залученні такої великої кількості розробників, необхідно забезпечити їх відповідним програмним забезпеченням, яке відповідає потребам сучасного світу розробки ігор.

### **Ігровий двигун**

Розробка кожної гри визначається вибором ігрового двигуна. Наприклад, для мобільних ігор з 2D-графікою, можливо, краще розробити власний двигун. Але при розробці проекту у повному 3D сьогодні краще скористатися готовими графічними двигунами.

Один із найпопулярніших ігрових двигунів - Unity, який з'явився у 2004 році. Зараз він став справжнім лідером у галузі, підтримуючи 25 апаратних платформ, включаючи ПК, ігрові консолі, смартфони та smart телевізори. Unity має багато переваг, включаючи безкоштовну версію для особистого користування та можливість безкоштовного отримання для студій, які розробляють ігри для PlayStation, Xbox або Nintendo.

Щодо Unreal Engine, це потужний інструмент для створення 3D програм та ігор AAA-класу. Він використовує мову програмування C++ і підтримує розробку для різних платформ. Починаючи з 2015 року, Unreal Engine став доступним безкоштовно, за умови, що дохід від проектів, створених за його допомогою, не перевищує 3000 доларів за квартал.



Рис. 1.11 – Інтерфейс ігрового двигуна Unreal Engine 5

Для вибору найкращого ігрового двигуна для розробки гри, потрібно враховувати такі фактори: їх функціональні можливості, умови використання, вартість, підтримку сторонніх інструментів розробки та зручність використання.

## **Техніки анімації**

Щоб оживити навіть 2D-моделі для анімації, особливо у грі з великою сюжетною лінією, дизайнери використовують спеціальне програмне забезпечення з технологією skeleton. Вони додають "кістки" до фігур персонажів та об'єктів, щоб швидко створювати анімації, переміщуючи ці кістки.

Також серед популярних видів анімації можна виділити покадрову анімацію, яка здебільшого використовується в платформерах та проектах, де залучена мала кількість розробників. У покадровій анімації кожен кадр створюється вручну або за допомогою спеціальних програм. Це може бути малювання різних положень об'єкту на кожному кадрі або використання комп'ютерних програм для створення та редагування кадрів. Коли кадри відтворюються з відповідною швидкістю, вони створюють враження руху.

## **Існуючі засоби для анімації**

Додаткові графічні інструменти, які можуть бути корисні при створенні гри, охоплюють широкий спектр додатків. Наприклад, для надання реалістичності 3D-моделям шкіри або інших текстур можна використовувати програму Substance Painter. Для швидкого створення рослинності в грі можна скористатися додатком GrowFX, який має велику кількість шаблонів та зручні інструменти кастомізації. Для моделювання і малювання як у 2D, так і у 3D форматах, зазвичай використовують програму Zbrush, яка дозволяє створювати моделі, натхненні глиняними фігурками.

Крім того, невід'ємною частиною процесу розробки є інструменти для обробки зображень, такі як Photoshop, він надає змогу виконувати комфортну роботу в обробці зображень протягом багатьох років і може бути корисним на будь-якому етапі створення ігор.



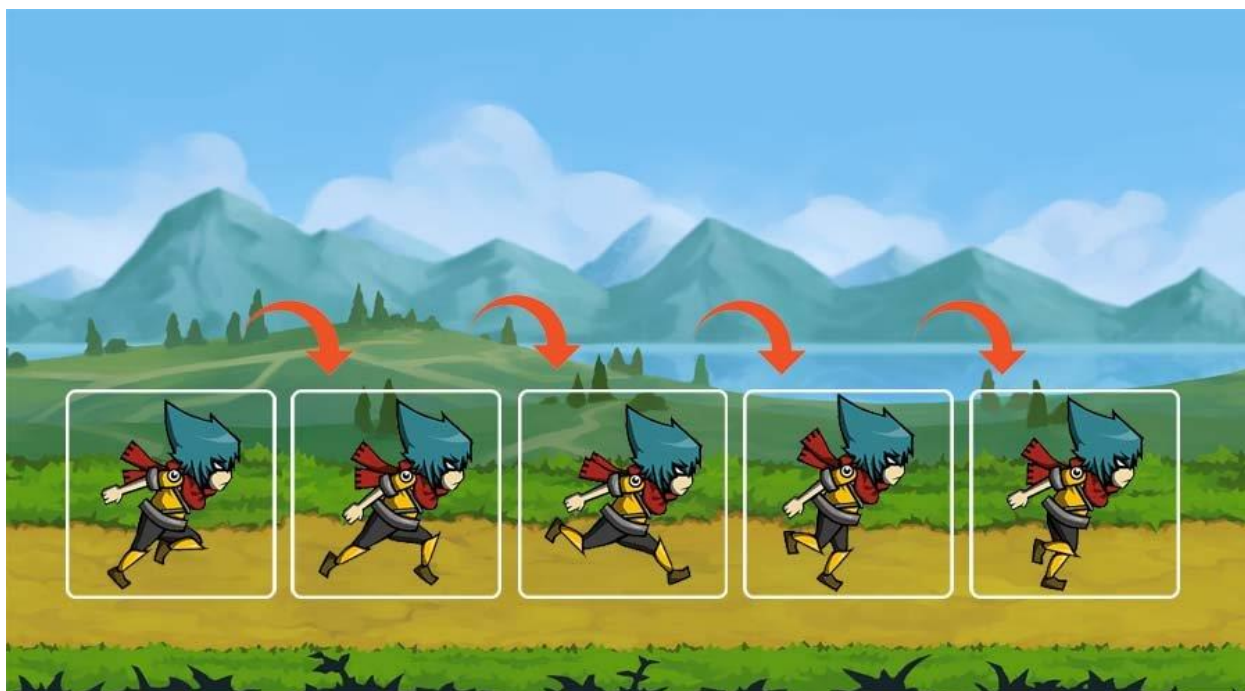


Рис. 1.12 – Покадрова анімація 2D спрайту в Unity

### **Керування версіями**

Для надійності зберігання виконаної роботи важливо мати можливість керувати версіями і мати доступ до проекту віддалено. Протягом розробки проект може зазнавати декілька змін, тому важливо мати засоби для ефективного контролю за ними. Для цих потреб існують спеціальні програми та сервіси, такі як Perforce та GitHub, які широко використовуються у галузі розробки ігор та анімацій.

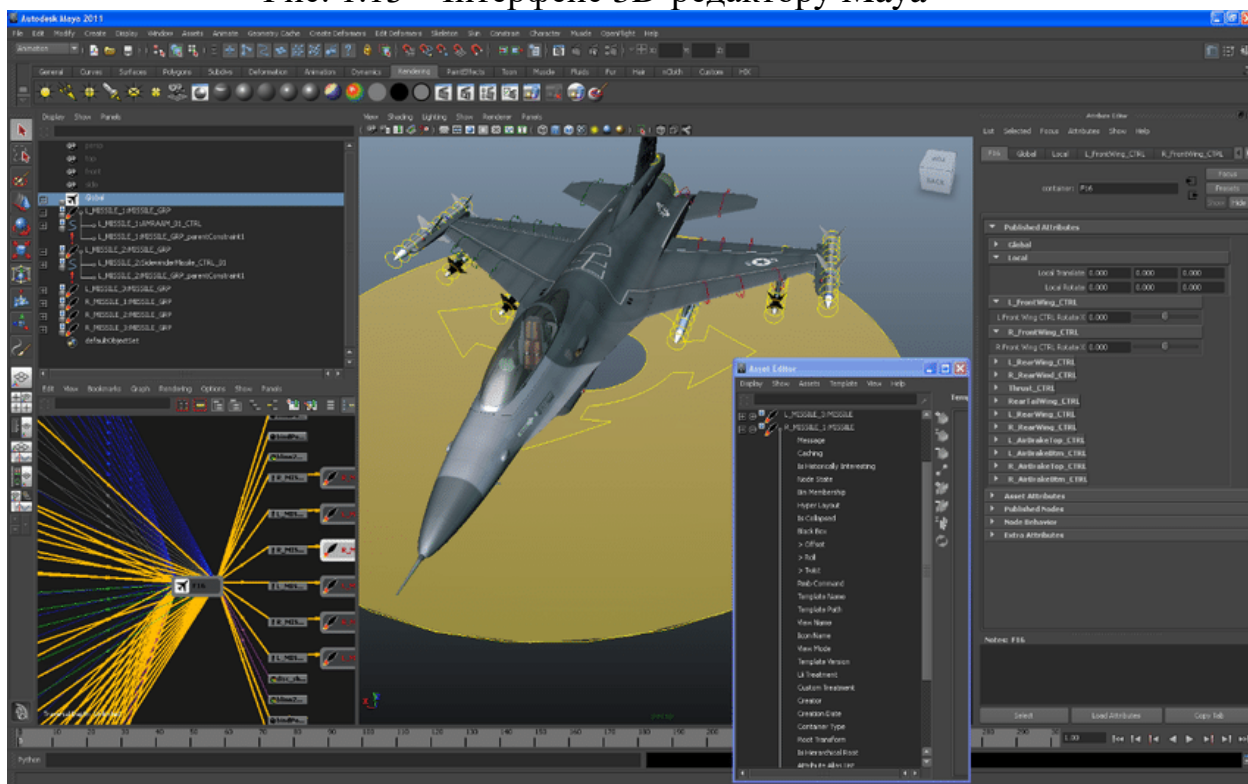
### **Інструменти для 3D-моделювання**

Розробникам ігор часто необхідне потужне середовище для 3D-моделювання. Навіть якщо гра має двовимірну графіку, візуальний супровід для ігор часто створюється у 3D. Один з відомих і широко використовуваних інструментів для цього є - 3ds Max. Професіонали з вмінням користуватися ним дуже цінуються в сфері розробки ігор, оскільки цей програмний засіб дає

можливість для всебічного використання в створенні візуальних ефектів. Також існує безліч плагінів та розширень для 3ds Max, що дозволяють швидко та ефективно створювати різноманітні візуальні ефекти.

Ще одним відомим редактором є Мауа, який гарно впорається з створенням цікавих та ефектних сюжетів. Мауа складніший у вивченні, але результат вартий зусиль навчитися працювати в цьому застосунку. Багато голлівудських студій знаходять застосування Мауа для створення спеціальних ефектів у кіно.

Рис. 1.13 – Інтерфейс 3D-редактору Мауа



## 1.5 Створення ігрових додатків за допомогою рушія Unity

### 1.5.1 Ігровий рушія Unity

Спочатку Unity був розроблений для створення ігор на Mac, але згодом його можливості розширилися й на Windows. У 2008 році з'явилася підтримка iOS, а у 2010 - Android. Лише потім Unity підкорив Xbox та Playstation.

Цей повноцінний ігровий двигун має вбудований редактор, що значно спрощує процес розробки. Завдяки Unity створено безліч популярних ігор, серед яких Sabnautica, Cities Skylines, Cuphead та багато інших. Unity поєднує в собі

простоту використання з потужністю, що робить його ідеальним інструментом для створення 2D та 3D-ігор. Платформа Unity також надає розробникам широкі можливості для підтримки та монетизації своїх проєктів.

Таким чином, Unity пройшов шлях від скромного початку на Mac до того, щоб стати одним з найпопулярніших ігрових двигунів у світі, який використовується для створення захоплюючих ігор на всіх платформах.

### **1.5.2 Можливості програмної платформи**

Unity пропонує зручний інтерфейс з безліччю налаштовуваних вікон, що робить процес розробки ігор та додатків максимально комфортним.

#### **Основні вікна:**

- Інспектор об'єкта: Відображає детальну інформацію про виділений об'єкт, дозволяючи змінювати його властивості.
- Вікно попереднього перегляду: Відображає поточний вигляд сцени в режимі реального часу.
- Оглядач ресурсів проєкту: Надає доступ до всіх ресурсів проєкту, таких як моделі, текстури, аудіо та скрипти.
- Огляд сцени: Відображає 3D-вигляд сцени, де можна розміщувати та переміщати об'єкти.
- Огляд ієрархії ресурсів: Відображає ієрархічну структуру всіх об'єктів у сцені.

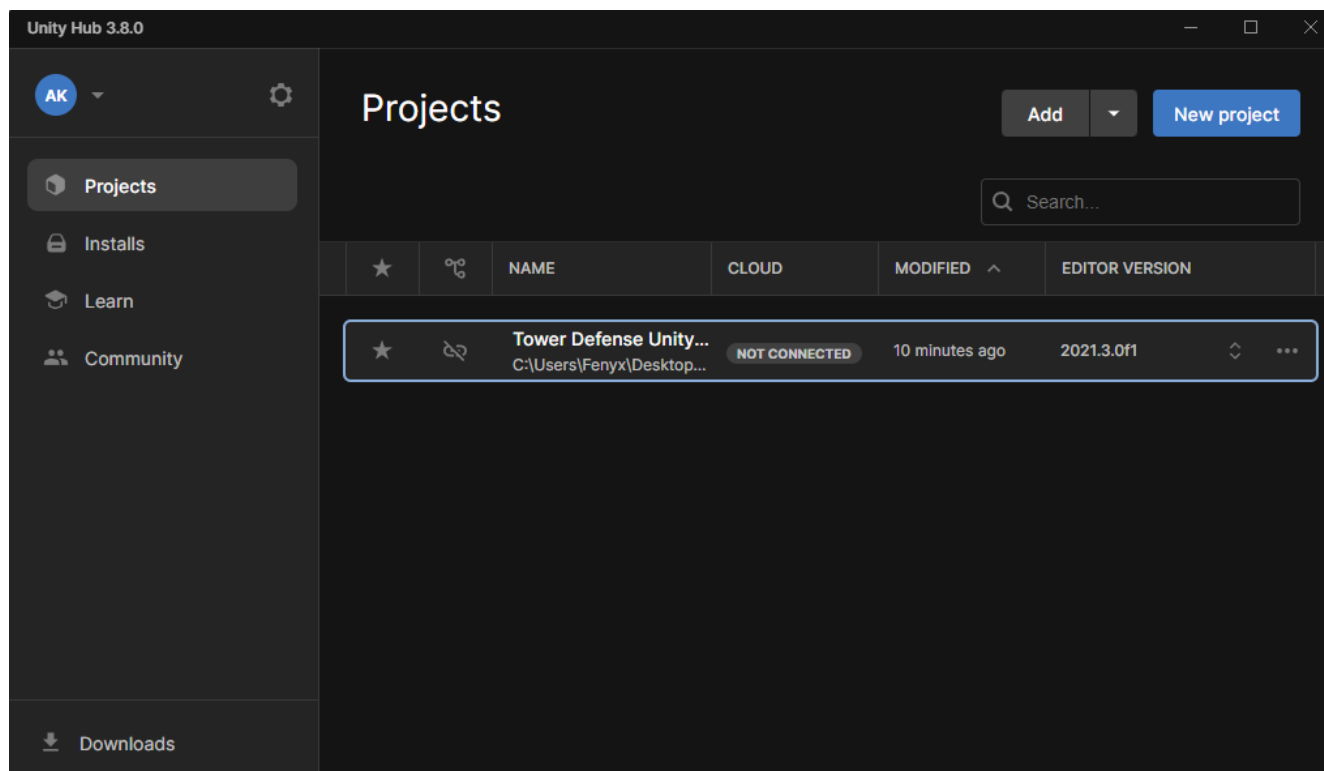


Рис. 1.14 – Інтерфейс огляду існуючих проектів та версій встановлених редакторів Unity

### Розробка проектів:

Unity використовує систему поділу проектів на сцени. Кожна сцена - це окремий файл, що містить частину ігрового світу з власними об'єктами, сценаріями та налаштуваннями.

### Сцени можуть містити:

- Деталі ігрового світу: Персонажі, предмети, ландшафт тощо.
- Ігрові об'єкти: Об'єкти без фізичної оболонки, які впливають на поведінку інших об'єктів.

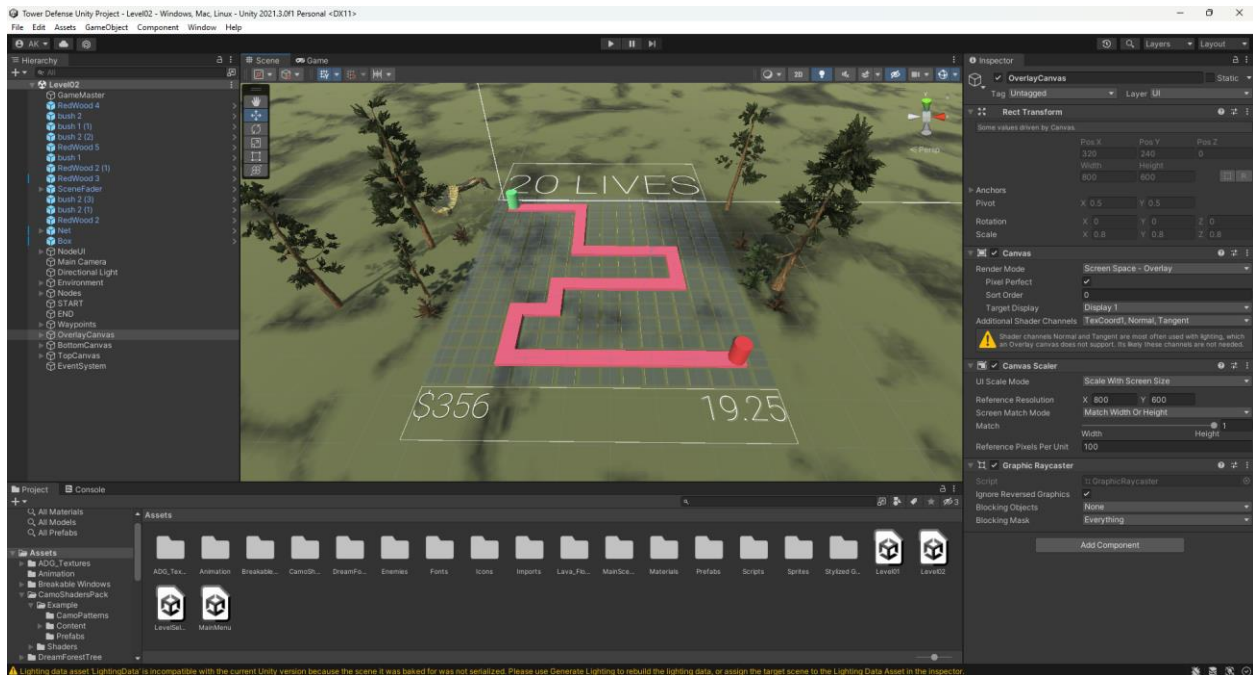


Рис. 1.15 – Інтерфейс двигуна Unity

## Компоненти:

Кожен об'єкт має компонент Transform, який зберігає його координати, поворот та розмір. Об'єкти з геометрією також мають компонент Mesh Renderer, який робить їх видимими. Моделі можна об'єднувати в набори (ассети) для зручного доступу.

## Фізика:

Unity підтримує фізику твердих тіл, тканин та Ragdoll (ганчіркова лялька). Існують два окремі фізичні рушії: один для 3D фізики і один для 2D фізики. Основні концепції ідентичні у обох рушіях (за винятком додаткового виміру у 3D), але вони реалізовані за допомогою різних компонентів. Наприклад, для 3D фізики існує компонент Rigidbody, а для 2D фізики - відповідний компонент Rigidbody 2D.

Також, редактор Unity має систему успадкування об'єктів. Дочірні об'єкти повторюють всі зміни, внесені до батьківського, що сприяє більш ефективному та зручному створенні нових об'єктів та їх властивостей.

## **Скрипти:**

До об'єктів можна додавати скрипти у вигляді окремих компонентів. Скрипти дають можливість розширити функціональність об'єктів та створювати складну ігрову логіку.

В цілому, Unity пропонує потужний та зручний інструментарій для розробки 2D та 3D-ігор та додатків, який використовується як початківцями, так і досвідченими розробниками.

### **1.5.3 Переваги та недоліки середовища розробки Unity**

#### **Переваги ігрового двигуна Unity:**

- Легкий для освоєння редактор та інструментарій, що дозволяє навчитися розробляти ігри навіть новачкам. Широка спільнота розробників завжди готова допомогти та відповісти на питання через форуми, відеоуроки, книги та курси.
- Сучасний рівень графіки, який може конкурувати з платними аналогами. Unity пропонує такі можливості, як відкладене освітлення, стандартні постпроцесингові ефекти, SSAO та швидку обробку лайтмапів.
- Велика спільнота розробників та велика кількість випущених ігор.
- Unity Asset Store, де можна придбати готові фрагменти коду, спрайти, асети та звуки.
- Легкість розробки та портабельність на різні платформи, включаючи Windows, Linux, macOS, Android, iOS, консолі PlayStation, Xbox, Nintendo, та VR- та AR-пристрої.

### **Недоліки середовища розробки Unity:**

- Потреба у базових навичках програмування для розробки на Unity, оскільки це не конструктор ігор, а середовище програмування.
- Деякі компоненти можуть здаватися складними для освоєння, а розроблені ігри можуть бути об'ємними, що може вплинути на швидкість завантаження та вимоги до обладнання.
- Розробник не має доступу до вихідного коду власної гри без звернення до інженерів Unity.
- Неінтегрованість зовнішніх сервісів та бібліотек, яка потребує ручного налаштування розробником.
- Використання мови програмування C# для Unity.

### **1.5.4 Створення ігор за допомогою мови програмування C#**

Процес створення ігор на Unity включає два основні етапи:

1. Розробка візуальної частини у середовищі Unity.
2. Написання коду на мові програмування C#.

Мова програмування C# була розроблена Microsoft і є однією з найпопулярніших мов програмування. Ігровий двигун Unity відповідає за багато технічних аспектів, таких як фізика та графіка, щоб розробники могли сконцентруватися на концепції гри. Однак без написання скриптів на C# гра залишиться неповноцінною, тому професійні розробники рекомендують використовувати саме цю мову для роботи з Unity.

C# є мовою, яка активно розвивається і має обширну документацію, яка допомагає вирішувати складні завдання. Unity використовує інтегроване середовище розробки Mono, що є кроссплатформеною реалізацією Microsoft .NET,

тому C# виступає основною мовою програмування, на якій базується платформа .NET і всі бібліотеки Unity.

Згідно з рейтингами DOU, C# займає п'яте місце серед мов програмування, популярних серед програмістів в Україні, а також на четвертому місці у світовому рейтингу популярності мов програмування за даними PYPL (Popularity of Programming Language). Це свідчить про високу популярність C# та його подальший розвиток.

Worldwide, Oct 2023 :

Rank	Change	Language	Share	1-year trend
1		Python	28.05 %	+0.1 %
2		Java	15.88 %	-1.0 %
3		JavaScript	9.27 %	-0.3 %
4		C#	6.79 %	-0.2 %
5		C/C++	6.59 %	+0.3 %
6		PHP	4.86 %	-0.4 %
7		R	4.45 %	+0.4 %
8		TypeScript	2.93 %	+0.1 %
9	↑↑	Swift	2.69 %	+0.7 %
10	↓	Objective-C	2.29 %	+0.2 %

Рис. 1.16 – Рейтинг мов програмування за даними PYPL

## 1.6 Огляд аналогів

Під час виконання проекту важливо зрозуміти очікування гравців та проаналізувати переваги та недоліки існуючих ігор у жанрі Tower Defense. Для



цього було обрано кілька екземплярів, таких як Deathtrap, Kingdom Rush, Prime World: Defenders 2, Revenge Of The Titans.

Аналізуючи ці ігри, можна помітити, що вони в основному доступні на платформах Android та iOS. Це відбувається через стрімке розширення мобільного геймінгу, оскільки мобільні пристрої забезпечують зручність та доступність для гравців у будь-якому місці. Проте, слід врахувати, що стаціонарні платформи також залишаються популярними серед багатьох гравців, які віддають перевагу геймінгу на ПК або ігрових консолях.

Важливо проаналізувати переваги та недоліки цих ігор для використання цих знань у розробці власного проекту на стаціонарних платформах.

### **Kingdom Rush**

Першою грою для огляду є Kingdom Rush, з самого релізу гра отримала велику кількість позитивних відгуків від користувачів та критиків. Серед ігрової спільноти серія ігор Kingdom Rush признана однією з кращих в своєму жанрі. Сюжет розвертається навколо темного мага, який захопив королівство та полонив більшість населення, для здійснення своїх злодіянь.

Геймплей має класичне представлення жанру Tower Defense, уздовж задалегідь встановленого шляху, яким йде суперник, є порожні місця, де гравець може будувати вежі. У грі існує чотири типи веж - башта магів, лучників, казарми та артилерія, всі вежі мають свої характеристики та здібності, а також два типи заклинань - виклик підкріплення в вигляді живої сили та вогненний дощ. На початку кожного рівня гравець має певну суму монет, щоб купити перші вежі. Мета гравця – знищити всіх ворогів до того, як вони пройдуть до кінця, позначеного синім щитом. Вбиваючи ворогів, гравець отримує додаткові монети для будівництва нових веж і поліпшення наявних.

Складність гри полягає в тому, що вороги йдуть з різних напрямлень, а ресурсів, за допомогою яких відбувається захист, в обмеженій кількості, тому гравцю треба використовувати їх з розумом. Також серед особливостей геймплею,

є наявність герою, яким гравець може керувати та захищати кінцеву точку більш ефективно. Героїв в грі достатня кількість, тому, для кожного рівня можна підібрати потрібного для успішного відбиття атаки противника.

Загалом у грі присутні 26 рівнів та 48 видів ворогів, що робить сюжет більш цікавим та захоплюючим. Після проходження рівня, гравцю стають доступними два додаткові режими - «Героїчний» і «Залізний виклик», які є складнішими порівняно зі звичайним режимом. За проходження кожного рівня гравець отримує зірки, які можна використовувати для придбання покращень для веж та заклинань.



Рис. 1.17 – Kingdom Rush

Як зазначено вище, у грі доступні три рівні складності: легкий, середній та важкий. Кожен рівень має свої особливості та відрізняється за складністю завдань і модифікаціями, такими як відсутність героїв, обмежена кількість життів або безмежна хвиля противників.



Рис. 1.18 – Демонстрація геймплею Kingdom Rush

Отримавши досвід гри, можна зрозуміти, що це повноцінний проект з цікавими ігровими механіками та функціями, який користується популярністю серед гравців усього світу. Проте, він має декілька недоліків:

#### **Недоліки:**

- Велика кількість механік, які можуть погано вплинути на ігровий досвід.
- Складність проходження рівнів

#### **Переваги:**

- Різноманітність босів та ворогів
- Приємний гейм-дизайн та дизайн рівнів
- Наявність додаткового озброєння у вигляді героїв
- Захоплюючий ігровий процес
- Вибір захисних споруд та їх покращення



## Deathtrap

Наступною для розгляду відеогрою стане Deathtrap, як і в попередньому прикладі, все зав'язано на класичному представленні жанру Tower Defense. Гравець обирає головного персонажа для проходження рівнів, а саме: стрілець, воїн та чарівниця. Три типи героїв мають свої особливі здібності та відрізняються по стилю гри.

Кожен рівень має декілька передбачуваних маршрутів, по яким проходять ворожі атаки. Для ефективнішого захисту користувач повинен будувати різноманітні пастки та покращувати їх за допомогою особливого ресурсу «Есенції», який можна отримати знешкоджуючі ворога та пройдені хвили. Перемогти та завершити рівень можна знищивши всіх противників та за допомогою недопущення їх певної кількості до кінцевої точки. Робота пасток, за допомогою яких потрібно ранили та вбивати ворогів, доповнюється пасивними та активними вміннями героїв, які можуть вільно пересуватися картою та підсилювати пастки різними способами.

Гра має доступ до кооперативного проходження рівнів для 4-ьох гравців та редактор карт. Тобто користувач може самостійно налаштувати рівень та види ворогів, яких треба буде знищити.



Рис. 1.19 – Відеогра Deathtrap

Оцінивши досвід гри, можна з впевненістю сказати, що це повноцінний продукт з досить цікавим геймплеєм та приємним і зрозумілим для користувача інтерфейсом, що є важливим компонентом в отриманні позитивного ігрового досвіду. Але є й недоліки:

**Недоліки:**

- Складність рівнів
- Відсутність української локалізації

**Переваги:**

- Різноманітність противників
- Вибір озброєння та її покращення
- Наявність RPG (role-play-game) складової
- Можливість кооперативного проходження гри

**Prime World: Defenders 2**

Друга частина, крім класичних механік, відрізняється набагато кращою графікою та величезною кількістю нових спецефектів. Крім того, розробник активно підтримує саме другу частину, постійно вдосконалюючи її і проводячи івенти. Згідно з сюжетом, у фентезійному світі триває війна за енергію Прайм між прихильниками технологій. Суть кожної місії полягає в захисті об'єкта, артефакту або родовищ «Прайму». Ми захищаємося від створінь, яких тут називають чудищами. Важливе відмінність від більшості Tower Defense - чудищам потрібно не лише дійти до об'єкта, а й повернутися до свого відновлювального пункту, йдучи тим же шляхом. Це трохи полегшує наше завдання та дає більше часу для адаптації до зміни умов. Чудищ можна знищити двома способами - заклинаннями та вежами. Заклинання створюються персонажами яких немає на карті, вони не потребують

ресурсів, але мають відкат. Вежі воюють постійно, але потрібні ресурси для будівництва.

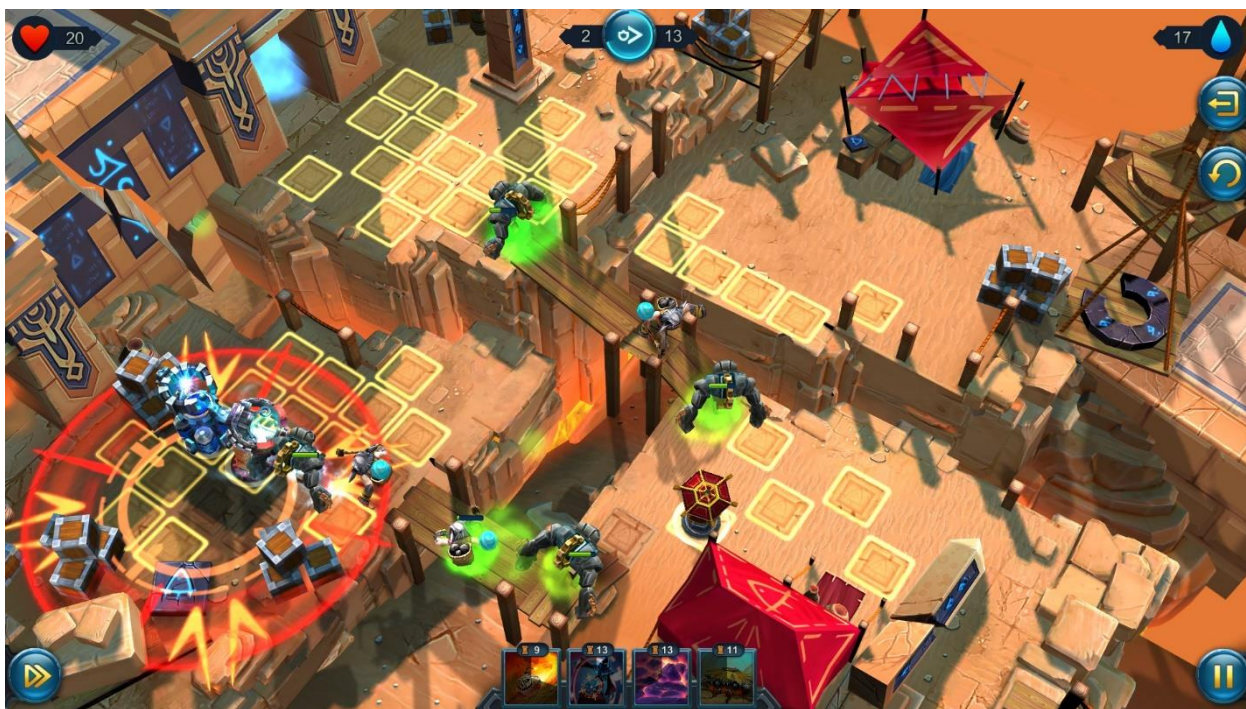


Рис. 1.20 – Ігрова карта одного з рівнів Prime World: Defenders 2

У розпорядженні користувача 41 герой, 21 заклинання та 54 вежі. Також серед цікавих механік є наявність системи рун, щоб створювати для веж унікальні збірки. Окрім прайму, який потрібен для будівництва веж, з чудовищ випадає срібло, його також дають за проходження місій та вбивство босів. За срібло між місіями можна купувати випадкові набори карт.

Це частково карткова гра. У тому сенсі, що всі герої, заклинання та вежі представлені картами. З карт перед місією ми складаємо колоду, якою і граємо. Звичайно, певні типи веж та заклинань ефективні проти певних ворогів. На високих рівнях можна скласти більш-менш універсальну колоду карт, для ефективного проходження рівнів.

Оглянувши даний екземпляр, можна сказати, що він має свої особливості в плані механік та цікавий дизайн. Ідея створення гри в такій стилістиці це дійсно вдалий варіант для зацікавлення користувача, але є й недоліки:

**Недоліки:**

- Складність рівнів
- Тривалість гри

**Переваги:**

- Велика кількість ворогів та босів
- Вибір озброєння (Вежі, герої, заклинання)
- Наявний PvP-режим (Гравець – проти - гравця)
- Стилiстика гри
- User-Friendly інтерфейс

**Revenge Of The Titans**

Ця гра має схожість з вже розглянутою грою Kingdom Rush. Суть кожної місії полягає в знищенні наступних хвиль прибульців. Штаб гравця знаходиться в центрі мапи, він сам по собі не має користі, але якщо його знищити - місія провалена.

Гравець може будувати заводи переробки, які перетворюють кристали на валюту, а також безліч засобів оборони - турелі, міни і засоби для бомбардування. Гра має механіку періодичних катастроф - від вибухів, що зносять все на карті, до раптового заморожування всіх рухомих об'єктів.





Рис. 1.21 – Гра Revenge Of The Titans

Звичайно, певні захисні рішення є найбільш ефективними проти певних прибульців, але головна особливість гри в дослідженнях. Між місіями можна відкрити одну нову технологію, їх більше, ніж місій, тому доведеться обирати. Економічна модель в грі непогана, тому новачки не відчують складності в проходженні гри. Окрім сюжету є виживання з передумовками та безкінечний режим. Серед недоліків та переваг застосунку можна виділити:

#### **Недоліки:**

- Перевантажений інтерфейс
- Користувач повинен мати неабияку швидкість реакції для проходження гри

#### **Переваги:**

- Цікавий сетінг та деталізація ігрового світу
- Різноманітність ворогів та босів
- Вибір озброєння та покращень



Отримавши ігровий досвід в проектах цього жанру, буде можливість коректно скласти план та вимоги для майбутнього застосунку, враховуючи всі недоліки та переваги оглянутих ігор.

### **1.7 Завдання для дослідження**

В момент формулювання концепції гри було встановлено, що перед тим, як переходити до проектування та вибору інструментів, потрібно провести дослідження ігрової сфери та вивчити додатки у схожому жанрі з подібним функціоналом для майбутнього проекту.

Для підвищення ефективності розробки проекту було визначено список основних завдань, який включав в себе дослідження технологій розробки та вибір необхідних інструментів., які дозволять реалізувати цей проект у майбутньому:

- Аналіз ігор конкурентів для встановлення їх плюсів та мінусів і виявлення основних вимог для створення власного продукту;
- Вивчення варіантів реалізації та переліку наявних технологій для розробки ігор;
- Визначення способів реалізації функцій;
- Аналіз структури проекту та необхідності розробки власних класів та скриптів мовою програмування C#.

## 2. РОЗРОБКА СТРУКТУРНОЇ СКЛАДОВОЇ ГРИ

### 2.1 Мета розробки

Наразі комп'ютерний ігровий ринок переживає помітне зростання популярності. Обсяг ігрової індустрії за 2023 рік склав 385 мільярдів доларів, що перевищує обсяг кіноіндустрії та спортивної індустрії. Це свідчить про велику популярність ігор поміж ком'юніті гравців, які роблять їх необхідною складовою свого життя.

Однак цей розвиток також створює виклики для розробників, такі як насиченість ринку, складність створення нового функціоналу та ускладнення процесу генерації ідей. Незважаючи на це, фанати ігор продовжують шукати нові та захоплюючі ігри з вражаючою графікою, унікальними функціями та привабливими інтерфейсами, що відповідають їхнім потребам та вподобанням.

**Мета роботи** – спрощення ігрового процесу в грі жанру Tower Defense за рахунок скорочення ігрової сесії для виконання завдання.

### 2.2 Розробка програмного об'єкту

#### 2.2.1 Діаграма прецедентів

Створення будь-якої системи, додатку чи гри спрямоване на задоволення потреб користувачів та допомогу їм у виконанні завдань. Здавалося б, можна просто зібрати всі потреби користувачів з онлайн-джерел та створити "ідеальний" продукт. Але такий підхід складний й не гарантує успіху. Тому важливо мати чіткий список ключових вимог та використовувати відповідні технології для їх реалізації.

Use Case (варіанти використання) – це популярний метод опису функціональності гри через послідовність дій, які виконує система у відповідь на

дії користувача. Цей метод допомагає визначити ключові функції, що забезпечують цінність для користувача.

Use Case допомагають:

- Визначити ключові функціональні вимоги проекту
- Спростити керування процесом розробки
- Чітко зрозуміти потреби користувачів та принципи роботи системи
- Забезпечити чітку та логічну структуру проекту

На етапі аналізу та проектування Use Case допомагають зрозуміти цілі користувачів та принципи роботи компонентів системи для досягнення необхідної функціональності.

Таким чином, Use Case – це цінний інструмент для опису функціональності систем, додатків та ігор, який допомагає розробникам створювати продукти, що відповідають потребам користувачів.

Створення Use Case допомагає не лише описати функціональність гри, але й протестувати її, оцінюючи правильність реалізації вимог. Діаграма Use Case складається з акторів (користувачів гри) та самих варіантів використання (Use Case), що описують, яку користь отримує користувач від певної дії. Актори позначаються фігурками людей, Use Case - овалами, а їхня взаємодія відображається стрілками.

Метою проекту є розробка гри жанру Tower Defense з типовим функціоналом та дотриманням усіх правил жанру. Основне завдання - забезпечити якісний ігровий досвід завдяки функціям, розробленим спеціально для цієї гри. Гра орієнтована на звичайних геймерів, зацікавлених жанром Tower Defense, але невдоволених наявними проектами.

Демонстрація всіх можливих дій користувача на (Рис. 2.1)

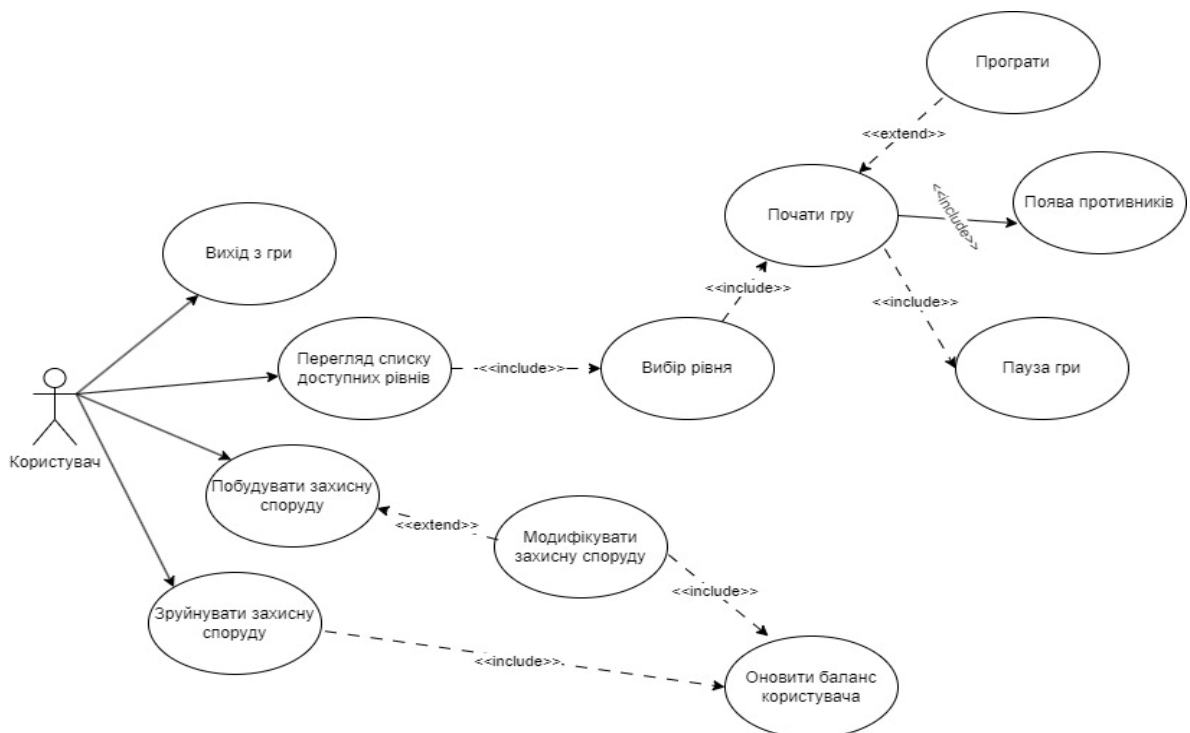


Рис. 2.1 – UML діаграма прецедентів

Діаграма класів у UML є важливим інструментом для моделювання структури програмного проекту. Вона відображає класи програми, їх атрибути та методи, а також взаємозв'язки між цими класами. Кожен клас зображується у вигляді прямокутника, в якому вказуються його ім'я, атрибути та методи. Взаємозв'язки між класами показуються за допомогою стрілок, які вказують на тип відносин між ними, такі як асоціація, агрегація або композиція.

Під час розробки діаграми класів спочатку визначаються основні класи програми, їхні атрибути та методи. Пізніше до діаграми можуть додаватися нові класи та відносини між ними під час продовження розробки проекту. Це дозволяє програмістам краще розуміти структуру програми та взаємодію між її компонентами.

Діаграми класів є важливою частиною проектування програмних систем, оскільки вони дозволяють візуалізувати структуру системи та взаємозв'язки між її складовими частинами.

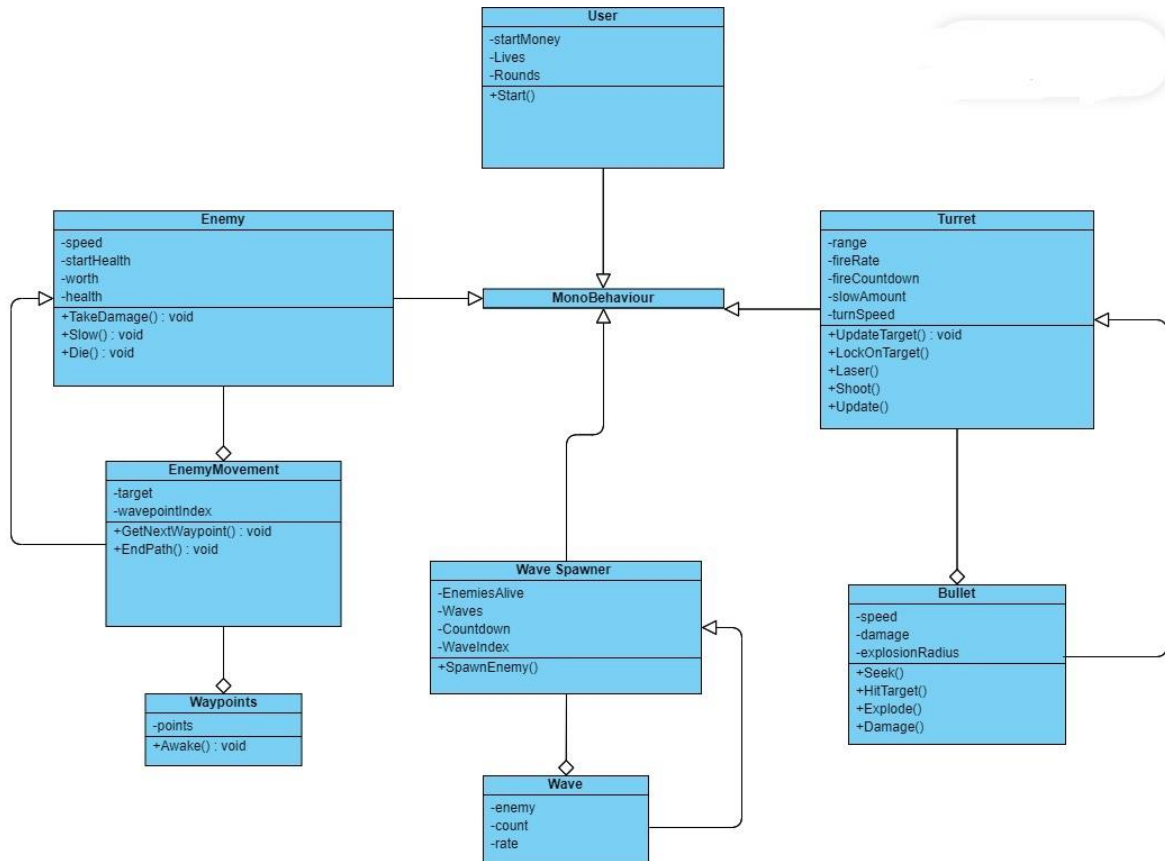


Рис. 2.2 – Діаграма класів

### 2.2.2 Організація даних у проєкті

Важливо розуміти, що правильна організація структури даних має велике значення для ефективної роботи алгоритмів обробки даних. Грамотно спроектована структура даних дозволяє оптимізувати час та використання пам'яті системи для виконання найважливіших операцій.

У випадку з розробкою гри для ПК, дані будуть зберігатися в файлах гри. Наприклад, для визначення шляху, яким рухаються вороги, буде використовуватися клас Waypoints. В цьому класі зберігаються точки, до яких рухаються вороги, а також алгоритм, який визначає наступну точку шляху.

```

1  using UnityEngine;
2
3  public class Waypoints : MonoBehaviour {
4
5      public static Transform[] points;
6
7      void Awake ()
8      {
9          points = new Transform[transform.childCount];
10         for (int i = 0; i < points.Length; i++)
11         {
12             points[i] = transform.GetChild(i);
13         }
14     }
15 }
16
17

```

Рис. 2.3 – Реалізація пошуку точок шляху

Для визначення механізму руху противника був створений клас EnemyMovement, де задано параметри переміщення противника та розроблено метод, який визначає наступний шлях по розробленій мапі та кінцеву точку призначення. Крім того, для зручності розробки було включено інші класи, які мають в собі опис типових механік застосунку, де кожен елемент описується згідно передбаченої логічної поведінки його роботи. На рисунку 2.4 показано загальний перелік створених класів, які в свою чергу мають опис функціоналу гри.

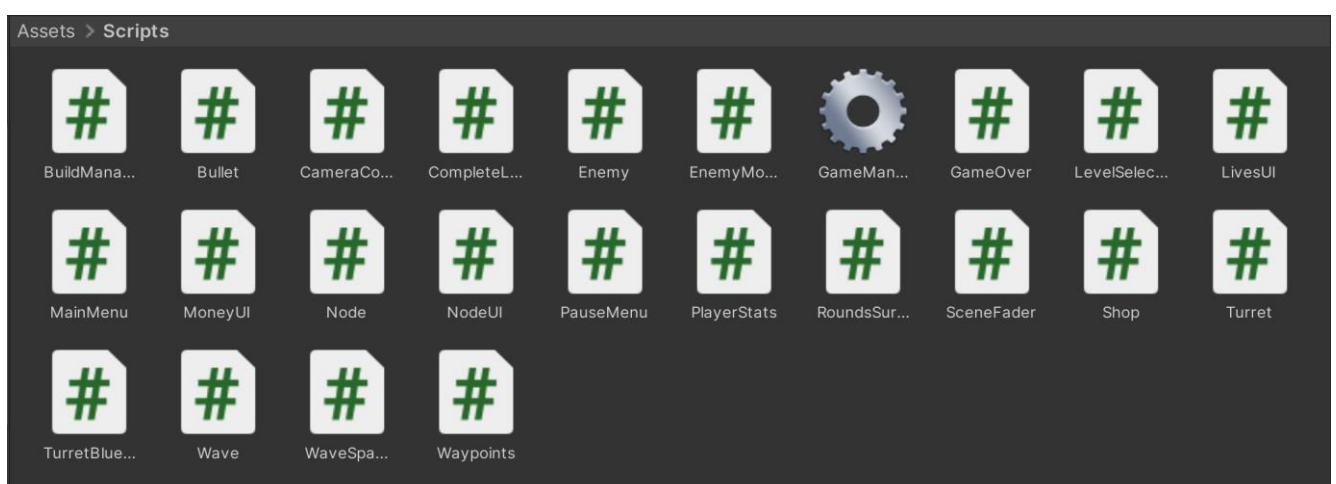


Рис. 2.4 – Класи проекту

Для збереження та обробки інформації про противника, був створений клас Enemy (Рис. 2.5), в якому описано дані стосовно його характеристик, а саме: стартова швидкість, початкове здоров'я, кошти отримані за його знищення та ряд методів, які характеризують його стан отримання шкоди та смерті.

```
3
4 public class Enemy : MonoBehaviour {
5
6     public float startSpeed = 10f;
7
8     [HideInInspector]
9     public float speed;
10
11     public float startHealth = 100;
12     private float health;
13
14     public int worth = 50;
15
16     public GameObject deathEffect;
17
18     [Header("Unity Stuff")]
19     public Image healthBar;
20
21     private bool isDead = false;
22
23     void Start ()
24     {
25         speed = startSpeed;
26         health = startHealth;
27     }
```

Рис. 2.5 – клас Enemy, опис характеристик ворога

### 3. РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СКЛАДОВОЇ ГРИ

#### 3.1 Використані засоби у процесі створення ігрового додатку

Розробка ігор часто вимагає вибору відповідного ігрового двигуна. Однак, враховуючи тему цієї роботи, необхідно створити гру за допомогою одного з найпопулярніших ігрових двигунів - Unity, використовуючи мову програмування C#.

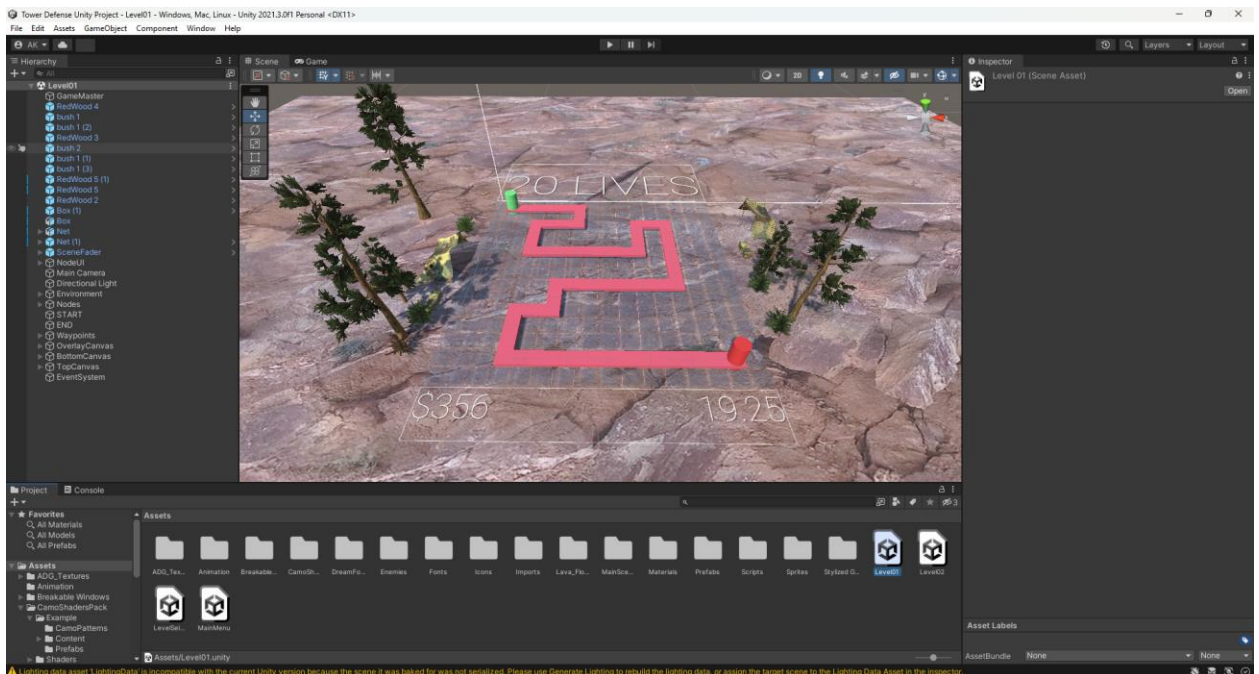


Рис. 3.1 – Ігровий двигун Unity

Unity рекомендується для використання саме з мовою програмування C#, що свідчить про його потужність та переваги порівняно з іншими аналогами:

- Має зручний інтерфейс;
- Відмінна сумісність з мовою програмування C#;
- Широкий вибір асетів та доступ до багатьох довідників та книг;
- Підтримує інтеграцію з системою контролю версій GIT.



Microsoft Visual Studio обрано в якості редактора коду через його відмінну інтеграцію з Unity та спільні переваги з цим ігровим движком.

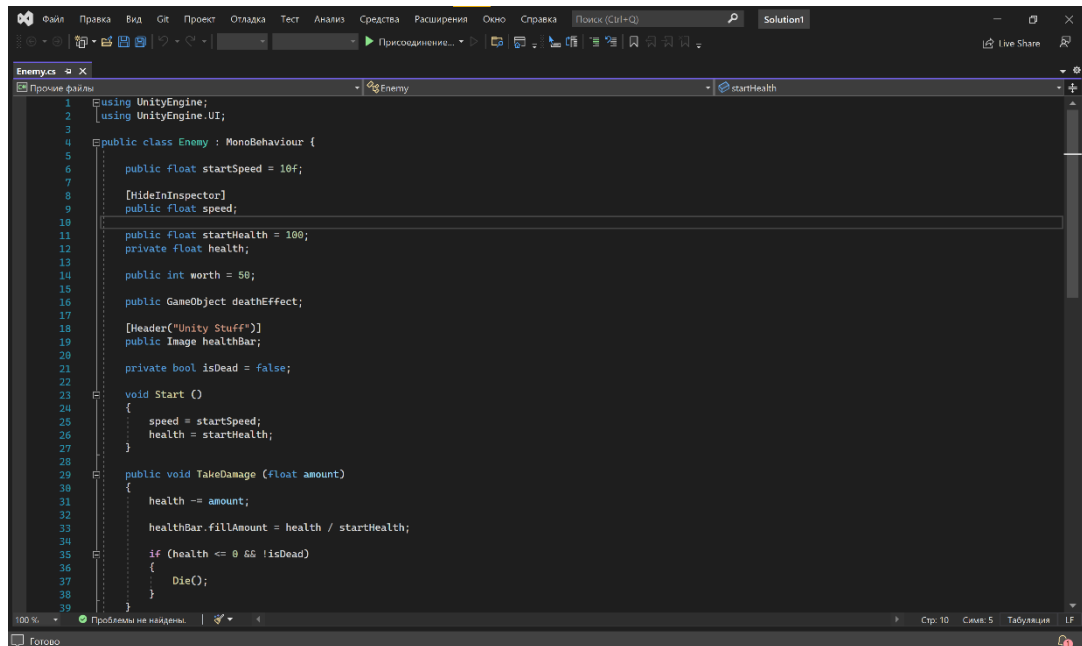


Рис. 3.2 – Редактор коду Microsoft VS

Для ефективного контролю за виконанням окремих завдань, які є важливими для успішної розробки проекту, а також для планування та документування прогресу роботи, був використаний інструмент Usрасу. Цей інструмент відзначається своєю простотою та зручністю в експлуатації, що дозволяє забезпечити зручний та організований менеджмент у ході розробки проекту.

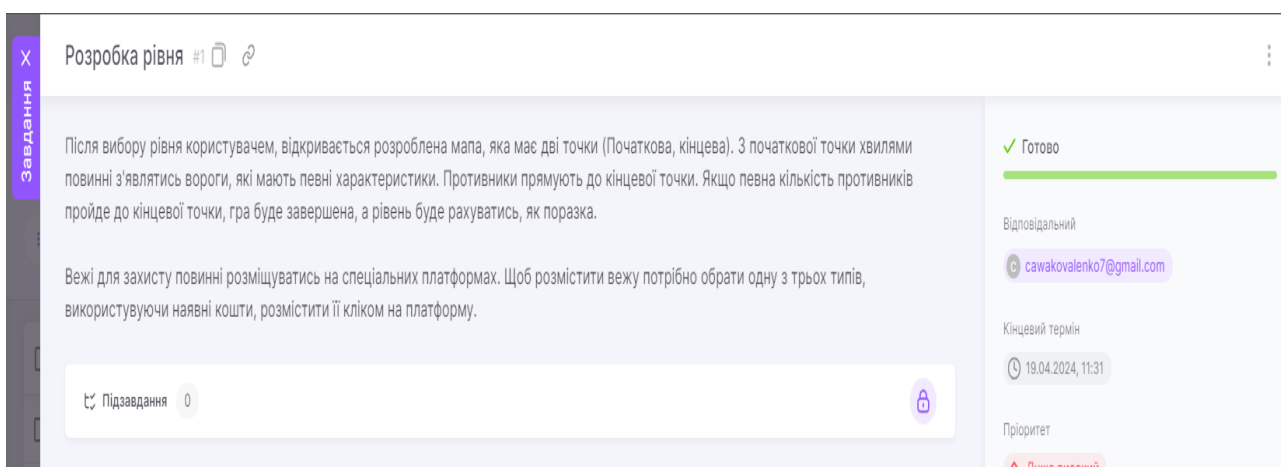


Рис. 3.3 – Засіб для менеджменту поставлених завдань Usрасу

### 3.2 Оцінка, планування та управління процесом створення гри.

Після визначення вимог та архітектури системи, виникла необхідність сформулювати список завдань, які потрібно було виконати. Головною метою розробки було забезпечити мінімально необхідний функціонал та декілька рівнів для демонстрації проекту. У зв'язку з обмеженими часовими рамками для розробки, було важливо використати цей час ефективно, щоб створити працюючий проект, який відповідав би основним вимогам та елементам, необхідним для гри.

Отже, було необхідно скласти список пріоритетних завдань, виконання яких дозволить визнати проект як мінімально життєздатний продукт.

<b>Функціональні вимоги</b>	
<b>№</b>	<b>Вимога</b>
1	Розробка рівня
2	Розробка противника
3	Реалізація руху противника по мапі
4	Розробка споруд для захисту
5	Реалізація логіки захисних споруд
6	Реалізація використання споруд та противників на мапі
7	Реалізація початкової та кінцевої точки рівня

Рис. 3.4 – Функціональні вимоги

Задачі, які були поставлені, були перенесені до системи управління проектами Usрасу для полегшення відстеження їх стану розробки. Ця система допомагає більш ефективно контролювати прогрес роботи над конкретними завданнями (див. Рис. 3.5).

The screenshot shows the 'Завдання' (Tasks) section of the Uspasy application. A sidebar on the left contains navigation options like 'Стрічка новин', 'Компанія та люди', 'Центр комунікацій', 'Завдання', 'Групи', 'CRM та продажі', 'Крамниця', 'Налаштування', 'Тариф та оплата', and 'Підтримка'. The main area displays a table of tasks with columns for 'НОМЕР', 'НАЗВА ЗАВДАННЯ', 'КІНЦЕВИЙ ТЕРМІН', 'ВІДПОВІДАЛЬНИЙ', 'ПРІОРИТЕТ', and 'СТАТУС'. The tasks are numbered 1 through 7, with various descriptions, deadlines, assignees (cawakovalenko7@gmail.com), and statuses (Готово, Залповано, У роботі).

НОМЕР	НАЗВА ЗАВДАННЯ	КІНЦЕВИЙ ТЕРМІН	ВІДПОВІДАЛЬНИЙ	ПРІОРИТЕТ	СТАТУС
7	Реалізація початкової та кінцевої точки рівня	05.05.2024, 23:00	cawakovalenko7@gmail.com Користувач	Високий	Готово
6	Реалізація використання споруд та противників...	28.04.2024, 20:30	cawakovalenko7@gmail.com Користувач	Середній	Залповано
5	Реалізація логіки захисних споруд	27.04.2024, 23:00	cawakovalenko7@gmail.com Користувач	Середній	У роботі
4	Розробка споруд для захисту	27.04.2024, 21:00	cawakovalenko7@gmail.com Користувач	Середній	У роботі
3	Реалізація руху противника по мапі	25.04.2024, 21:00	cawakovalenko7@gmail.com Користувач	Середній	Готово
2	Розробка противника	22.04.2024, 17:00	cawakovalenko7@gmail.com Користувач	Високий	Готово
1	Розробка рівня	19.04.2024, 11:31	cawakovalenko7@gmail.com Користувач	Високий	Готово

Рис. 3.5 – Управління процесом створення гри в Uspasy

Задачі було детально описано, з урахуванням функцій, які вони повинні виконувати та з чітко визначеними термінами та пріоритетністю їх виконання (Рис. 3.6).

The screenshot shows the detailed view of a task titled 'Розробка споруд для захисту #4'. The task description reads: 'Розробити вежі/башти - основні засоби для захисту. Вони здатні стріляти по ворожим одиницям на визначену відстань. Функціонал може включати в себе різні типи веж: звичайні вежі, що стріляють снарядами, або спеціалізовані вежі, такі як лазерні або ракетні. Реалізувати можливість ідентифікації цілі, шкоду яку башта завдасть противнику.' The task is currently 'У роботі' (In progress) with a progress bar. The assignee is 'cawakovalenko7@gmail.com', the deadline is '27.04.2024, 21:00', and the priority is 'Невисокий' (Low). There are no sub-tasks, comments, or watchers listed.

Рис. 3.6 – Опис завдання

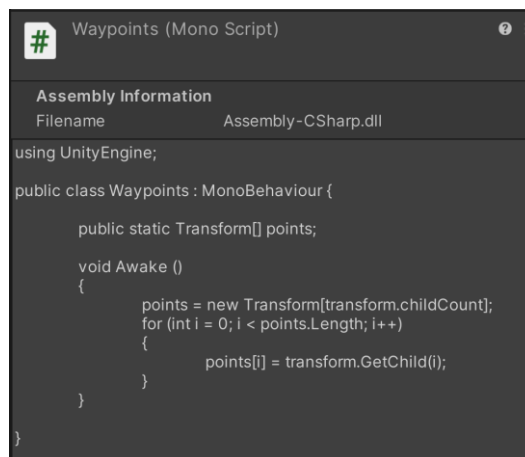
### 3.3 Функціональні можливості гри

При описі функціоналу гри було відокремлено декілька ключових функцій:

- Генерація ворогів на карті (Wave Spawner);
- Визначення маршрутів руху (Waypoints);
- Перехід на наступний рівень (Level Selector);
- Завершення рівня за певних умов (Complete Level);
- Облік статистики гравця на рівні (Player Stats);
- Рахунок раундів хвиль ворогів (Rounds Survived).

Усі ці функції реалізовані за допомогою відповідних скриптів, які мають особисті методи та поля для реалізації необхідної логіки.

Для реалізації пошуку шляху використовуються спеціальні точки, до яких рухаються вороги. Скрипт визначає наступний напрям, по якому повинні дістатися вороги, і після досягнення цієї точки встановлює нову ціль, до якої вони прямують далі.



```
Waypoints (Mono Script)
Assembly Information
Filename Assembly-CSharp.dll
using UnityEngine;
public class Waypoints : MonoBehaviour {
    public static Transform[] points;
    void Awake ()
    {
        points = new Transform[transform.childCount];
        for (int i = 0; i < points.Length; i++)
        {
            points[i] = transform.GetChild(i);
        }
    }
}
```

Рис. 3.7 – Скрипт Waypoints, пошук шляху для ворога

Наприклад, для спавну нових хвиль противників використовується скрипт Wave Spawner, що містить метод SpawnWave(), який стежить за кількістю ще живих противників. Коли всі хвилі ворогів знищені, відбувається перехід на наступний рівень або до головного меню гри через відповідний інтерфейс.

```

IEnumerator SpawnWave ()
{
    PlayerStats.Rounds++;

    Wave wave = waves[waveIndex];

    EnemiesAlive = wave.count;

    for (int i = 0; i < wave.count; i++)
    {
        SpawnEnemy(wave.enemy);
        yield return new WaitForSeconds(1f / wave.rate);
    }

    waveIndex++;
}

void SpawnEnemy (GameObject enemy)
{
    Instantiate(enemy, spawnPoint.position, spawnPoint.rotation);
}

```

Рис. 3.8 – Генерація ворогів на карті

Перехід до наступного рівня активується після того, як усі хвилі ворогів для поточного рівня будуть знищені, і їхня кількість відповідатиме заданому значенню. Таким чином, коли всі умови виконані, відбувається перехід до нового рівня.

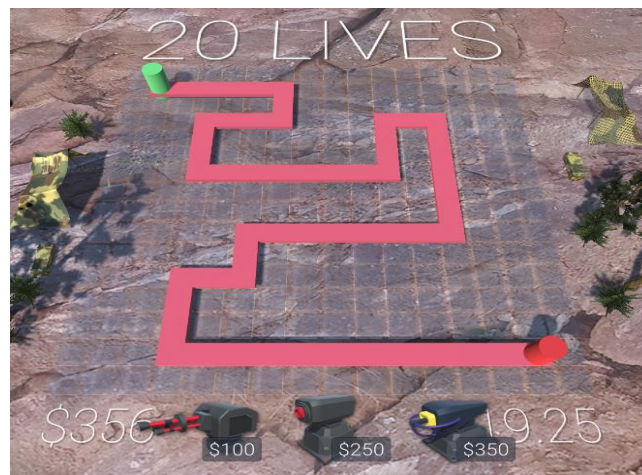


Рис. 3.9 – Відображення статистики користувача на поточному рівні

Щоб користувач орієнтувався в особистій статистиці, такої як кількість грошей для будівництва захисних споруд, кількість хвиль ворогів та час до наступу нової хвилі противника, створений спеціальний скрипт, який обробляє ці дані. Вся ця інформація відображена на екрані для всіх рівнів у вигляді інтерфейсу, стилізованого під елемент гри, що забезпечує органічне візуальне поєднання з основним геймплеєм.

## 4. ДЕМОНСТРАЦІЯ ЗАСТОСУВАННЯ ТА ТЕСТУВАННЯ

### 4.1 Характеристика користувацького інтерфейсу і процесу гри

Після першого запуску кожної гри користувачі спочатку бачать меню та його інтерфейс. Було важливо створити зрозумілий і естетично приємний інтерфейс в єдиному стилі. У меню присутні дві кнопки: «Play» та «Quit». Ці кнопки розташовані на зображеннях захисних споруд з гри, щоб гравці одразу могли побачити основні елементи, з якими їм доведеться взаємодіяти.

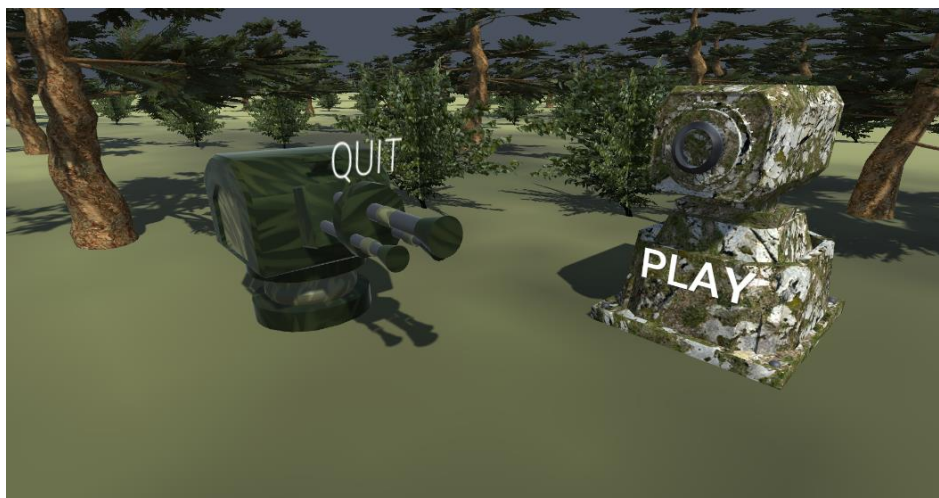


Рис. 4.1 – Головне меню

При натисканні кнопки «Quit» гра завершиться, а при натисканні кнопки «Play» користувач перейде до нового меню для вибору рівня. Це меню містить список рівнів, які доступні для гравця (Рис. 4.2).

Після вибору рівня починається гра. Гравець побачить поле битви з двома основними точками: точкою виходу противника і кінцевою точкою. Якщо вороги досягають кінцевої точки, гра завершується невдачею. Вороги з'являються з першої точки і будуть різноманітними як за зовнішнім виглядом, так і за характеристиками.

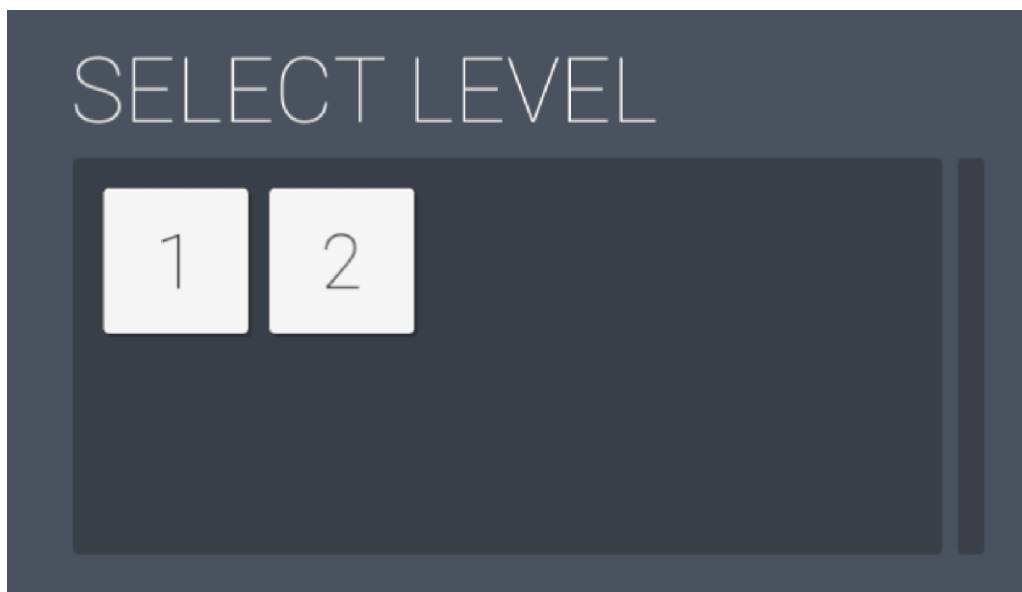


Рис. 4.2 – Меню вибору ігрового рівня

Інтерфейс гри включає також відображення статистики на ігровому полі. Сюди входить інформація про кількість ворогів, які можуть досягти кінцевої точки без провалу гри (на першому рівні гра закінчиться, якщо хоча б один ворог досягне кінця). Внизу ігрового поля відображаються доступні кошти для будівництва захисних споруд і час до появи наступної хвилі ворогів. Також у нижній частині головного інтерфейсу розташовані доступні для розміщення на мапі захисні споруди.



Рис. 4.3 – Ігровий інтерфейс



Щоб розмістити захисну споруду на ігровому полі, потрібно вибрати одну з доступних споруд і натиснути на панель, щоб встановити її для захисту. Повторне натискання на вже розміщену споруду дозволяє продати її або покращити (Рис. 4.4).



Рис. 4.4 – Взаємодія з захисними спорудами

Щоб гравець міг визначити, яка захисна споруда покращена, її зовнішній вигляд змінюється: вона стає більшою, а кулі, якими вона стріляє, набувають іншого вигляду. Покращення також впливає на характеристики зброї, підвищуючи шкоду, яку вона завдає певним типам ворогів (Рис. 4.5).



Рис. 4.5 – Процес проходження гри

Якщо рівень провалено, відкриється меню, з якого можна спробувати пройти рівень ще раз або повернутися до головного меню.

На другому рівні гравця зустрічає вже знайомий інтерфейс, оформлений в одному стилі. Однак, для підвищення привабливості кожен рівень або кілька рівнів варто змінювати фон і кольорову палітру відповідно до нового фону.

Після успішного завершення рівня з'являється меню з двома кнопками: «Continue» для переходу до наступного рівня і «Menu» для виходу в основне меню (Рис. 4.6).

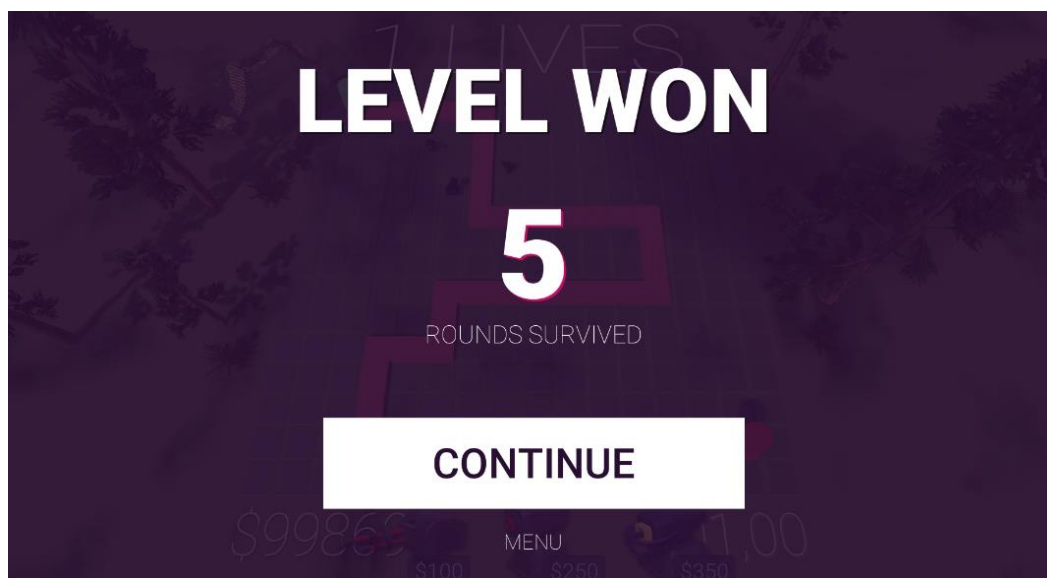


Рис. 4.6 – Успішне проходження ігрового рівня

## 4.2 Наступні етапи розвитку гри

Щоб підтвердити готовність продукту до використання та виявити можливі недоліки, які могли бути пропущені під час розробки, було необхідно провести апробацію проекту або його тестування серед певної групи користувачів.

Для цього було організовано тестування серед шести користувачів, які відзначили як сильні сторони продукту, так і деякі його недоліки.

Виявлено кілька переваг:

- Стабільність роботи додатку і коректне виконання функцій;
- User-Friendly інтерфейс та підібрана палітра кольорів;

- Простота використання ;

Однак були вказані й недоліки:

- Невелика різноманітність ворогів і захисних веж;
- Відсутність вибору складності гри;
- Обмежена кількість рівнів;

Після отримання цієї інформації, яка підтвердила, що продукт відповідає мінімальним вимогам і має високу якість використання, був розроблений план подальшого розвитку гри та її вдосконалень:

- Додавання нових рівнів з різними стилями;
- Розширення асортименту захисних споруд;
- Поліпшення ігрового інтерфейсу;
- Розширення функціональних можливостей;
- Збільшення різноманітності ворогів.

Цей план спрямований на усунення виявлених недоліків і впровадження нових функцій, що створять процес гри більш захоплюючим і приємним для гравців.

## ВИСНОВКИ

Підсумовуючи, можна відзначити, що попит на нові ігри стабільно підвищується. Гравці в постійних пошуках чогось нового, що дозволить їм здобувати гарні враження та емоції і наділятиме ігри новими можливостями, сюжетами і функціоналом.

Робота була спрямована на створення ігрового продукту в жанрі Tower Defense в воєнному стилі застосовуючи ігровий двигун Unity і С# як мову програмування. Це дозволить новачкам у галузі ігрової індустрії поглибити своє розуміння цієї галузі та ознайомитися з незвичним жанром.

1. Проведено аналіз актуальності проекту, який підтвердив, що для набуття позитивного досвіду геймерам потрібні прості ігри, які дозволяють новачкам ознайомитися з гральним середовищем. Встановлено, що геймерська спільнота постійно шукає нові ігри.

2. Досліджено сучасні технологічні рішення, які застосовуються в індустрії ігор. Вибір інструментів для розробки цього проекту, таких як Unity, Unity Asset Store, середовище розробки Microsoft Visual Studio, мову програмування С#, Uspasy, виправдано шляхом порівняння з популярними альтернативами.

3. Було виконано аналіз наявних стандартних проектів. Це дозволило виявити основні недоліки цих проектів і встановити власну мету реалізації проекту. Провівши аналіз аналогів було сформульовано мінімальні вимоги, що допомогли завершити проект вчасно і досягти поставлених цілей. Опис вимог допоміг розробити функціонал та основні методи, що визначають логіку появи противників та роботу захисних споруд, що є головним аспектом гри.

4. Описано програмне забезпечення та інструментарій, використані для розробки гри. З'ясувалося, що ігровий двигун Unity забезпечує простоту розробки та ефективну роботу додатку на різних ПК, що дозволяє запускати

гру навіть на менш потужних платформах. Було обрано мову програмування C#, оскільки вона гарантує правильність розробки методів та функцій, необхідних для цього проекту. Також для ефективного менеджменту та відстеження статусу виконання проекту використовувався Uspace, що дозволило легко описувати вимоги з детальними характеристиками та відмічати виконання завдань.

5. Для оцінки якості та відповідності проекту було проведено практичні тестування з участю шести досвідчених гравців. Вони виявили як переваги, так і недоліки проекту, що допомогло визначити напрямки подальшої розробки та вдосконалення. Основні переваги включають:

- Легкість використання ігрового додатку;
- Надійна працездатність гри та коректне виконання функцій;
- Привабливий інтерфейс і добре обрана палітра кольорів.

Результати досліджень бакалаврської роботи були представлені на всеукраїнських науково-технічних конференціях:

Коваленко О.М., Зінченко О.В.. Інноваційні підходи до розробки гри: Використання штучного інтелекту для оптимізації геймплею: Матеріали Всеукраїнської науково-практичної конференції «Сучасні інтелектуальні інформаційні технології в науці та освіті». Збірник тез. 15.05.2024, ДУІКТ, м.Київ. К.:ДУІКТ – ст. 341-343.

Коваленко О.М., Замрій І.В.. Психологія дизайну: як вивчення людської поведінки впливає на UI/UX: Матеріали Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в ІКТ». Збірник тез. 24.04.2024, ДУІКТ, м.Київ. К.:ДУІКТ ст. 133-135.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Відеоігри захоплюють світ [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://www.epravda.com.ua/publications/2023/05/18/700238/> - Дата звернення: 15.05.2024
2. Історія розвитку комп'ютерних і відео ігор: Від Пакмена До Кіберспорту. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://computergameshistory.blogspot.com/> - Дата звернення: 15.05.2024
3. Про стан ігрової індустрії, слоти та очікування від геймдеву України. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://gamedev.dou.ua/articles/gamedev-dou-podcast-3/> - Дата звернення: 15.05.2024
4. Роль ігрової індустрії як складової інформаційних технологій [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://essuir.sumdu.edu.ua/handle/123456789/86132> - Дата звернення 15.05.2024
5. Види комп'ютерних ігор: жанри та монетизація [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://tsn.ua/cybersport/vidi-komp-yuternih-igor-zhanri-ta-monetizaciya-1710949.html> - Дата звернення: 16.05.2024
6. Кращі ігри в жанрі Tower Defense для смартфонів [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://igate.com.ua/news/18170-luchshie-igry-v-zhanre-tower-defence-dlya-smartfonov> - Дата звернення: 16.05.2024

7. Яку програму вибрати для створення гри? [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://itproger.com/ua/news/kakuyu-programmu-vibrat-dlya-sozdaniya-igri> - Дата звернення: 16.05.2024
8. Хочу розробляти ігри: яку мову програмування вчити першою? Поради спеціалістів. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://gamedev.dou.ua/articles/first-programming-language-in-gamedev/> - Дата звернення: 16.05.2024
9. Розробка ігор для мобільних пристроїв з Unity та WebGL [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://it-rating.ua/rozrobka-igor-dlya-mobilnih-pristroiv-z-unity-ta-webgl> - Дата звернення: 16.05.2024
10. Документація Unity3D [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://docs.unity3d.com/Manual/> - Дата звернення: 17.05.2024
11. Перехід на Unity, як програмісту потрапити в геймдев. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://dou.ua/lenta/articles/go-to-unity/> - Дата звернення: 17.05.2024
12. Офіційна документація C#. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://learn.microsoft.com/ru-ru/dotnet/csharp/> - Дата звернення: 17.05.2024



13. Про середовище програмування C# [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://foxminded.ua/seredovyshche-prohramuvannia-si-sharp/> - Дата звернення: 17.05.2024
14. Середовище розробки Visual Studio. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://visualstudio.microsoft.com/ru/> - Дата звернення: 17.05.2024
15. Документація Visual Studio [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://learn.microsoft.com/ru-ru/visualstudio/windows/?view=vs-2022> - Дата звернення: 17.05.2024
16. Найвідоміші відеоігри, створені вітчизняними розробниками. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://www.redbull.com/ua-uk/best-ukrainian-games> - Дата звернення: 28.04.2022
17. Kingdom Rush – Tower Defense on Steam. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: [https://store.steampowered.com/app/246420/Kingdom\\_Rush\\_Tower\\_Defense/?l=ukrainian](https://store.steampowered.com/app/246420/Kingdom_Rush_Tower_Defense/?l=ukrainian) - Дата звернення: 17.05.2024
18. Deathtrap [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://store.steampowered.com/app/310510/Deathtrap/> - Дата звернення: 19.05.2024
19. Prime World: Defenders 2. Безкоштовна комп'ютерна гра для ПК. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: [https://store.steampowered.com/app/429540/Prime\\_World\\_Defenders\\_2/](https://store.steampowered.com/app/429540/Prime_World_Defenders_2/) - Дата звернення: 19.05.2024
20. Revenge Of The Titans – Tower Defense on Steam. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: [https://store.steampowered.com/app/93200/Revenge\\_of\\_the\\_Titans/](https://store.steampowered.com/app/93200/Revenge_of_the_Titans/) - Дата звернення: 19.05.2024

## ДОДАТОК А

### ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



## РОЗРОБКА ГРИ У ЖАНРІ TOWER DEFENSE З СЕТИНГОМ В ВОЄННОМУ СТИЛІ МОВОЮ C# З ВИКОРИСТАННЯМ РУШІЯ UNITY

Виконав: студент 4 курсу,  
групи ПД-43  
Коваленко Олександр Миколайович  
Керівник роботи  
доцент кафедри ПЗ, доктор філософії, Дібрівний Олесь Андрійович

Київ – 2024

### МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** – адаптація ігрового процесу в грі жанру Tower Defense.
- **Об'єкт дослідження** – процес гри в жанрі Tower Defense.
- **Предмет дослідження** – ігровий застосунок в жанрі Tower Defense.

## ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Проаналізувати предметну галузь.
2. Провести аналіз існуючих аналогів ігор в жанрі Tower Defense.
3. Визначити функціональні та нефункціональні вимоги до гри.
4. Обрати інструментарій для розробки гри.
5. Створити прототип інтерфейсу гри.
6. Розробити ігровий застосунок.
7. Провести тестування гри.

3

## Аналіз аналогів

Характеристики	Deathtrap	Revenge Of The Titans	Kingdom Rush	Prime World: Defenders 2	Edge Of Tomorrow
<u>Платформи</u>	PC, macOS, Xbox One, PlayStation Network	iOS, Android, PC, Linux	iOS, Android, Nintendo Switch, PC	Android, PC	PC, Android, iOS, PlayStation Network, Nintendo Switch
<u>Складність ігрового процесу</u>	Середня	Середня	Дуже складна	<u>Складна</u>	Легка
Види ворогів	Багато	Багато	Багато	Багато	Невелика кількість
Тривалість ігрової сесії	Середня	Довга	Довга	Середня	Коротка
Монотонність геймплею	+	+	-	-	+
Система апгрейдів	+	-	+	-	+

4

## Концепт гри

- Після початку рівня гравцеві надається певний час для розміщення захисних споруд.
- З початкової точки на мапі починають рухатися противники. Якщо гравець правильно і в достатній кількості розмістив захисні споруди, противники будуть знищені.
- Гравець під час гри має можливість модифікувати споруди, тим самим покращуючи їх характеристики.
- Гра закінчується при знищенні всіх хвиль противників або після того як один з противників дійде до кінцевої точки

5

---

## Вимоги для ігрового застосунку

### Функціональні вимоги:

1. Реалізувати декілька рівнів для демонстрації основних функцій.
2. Розробити противників та забезпечити їх характеристиками.
3. Реалізувати логіку проходження шляху по мапі від початкової точки до кінцевої.
4. Розробити засоби для збереження результатів виконаного завдання.
5. Реалізувати показ статистики гравця.
6. Розробити захисні споруди, з можливістю їх модифікації та видалення.

### Нефункціональні вимоги:

1. Гра має бути інтуїтивно зрозумілою та легкою для вивчення користувачем.
2. Гра має бути адаптована для мобільних пристроїв з різними розмірами дисплеїв та мати зручне керування сенсорним екраном.

6

## Програмні засоби та інструменти реалізації

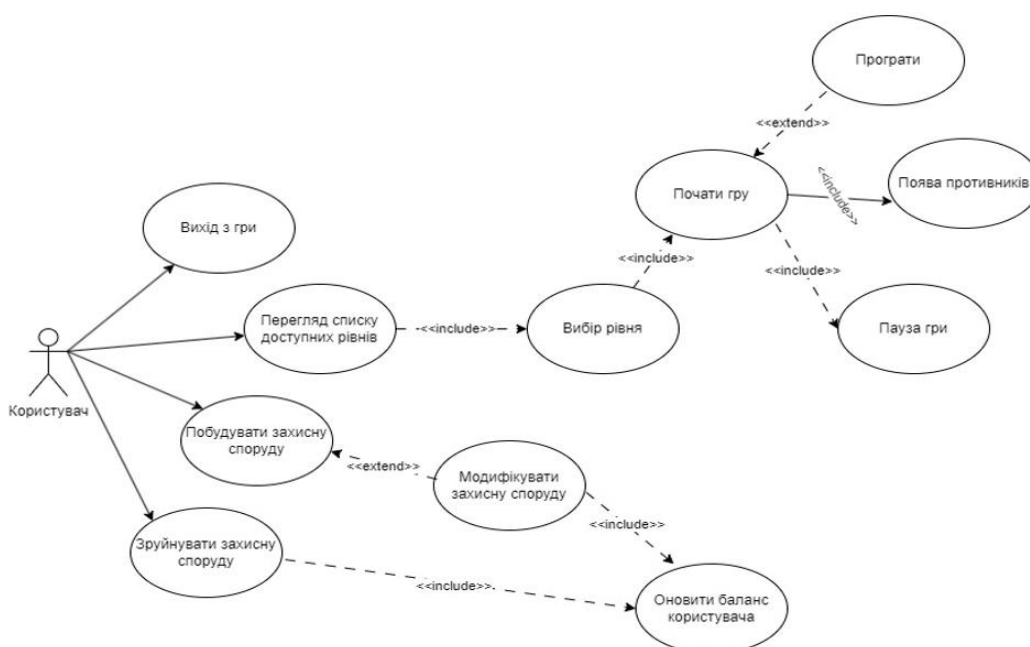


Microsoft Visual Studio



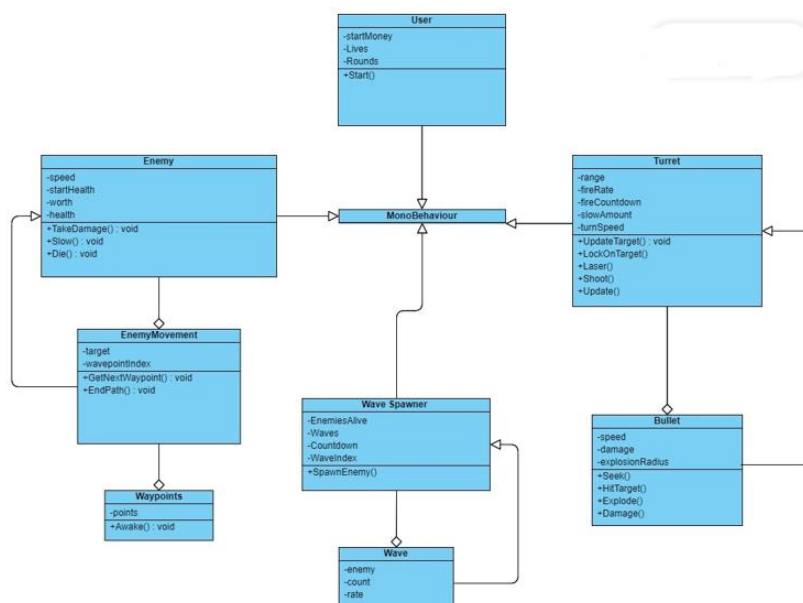
7

## Use Case діаграма користувача



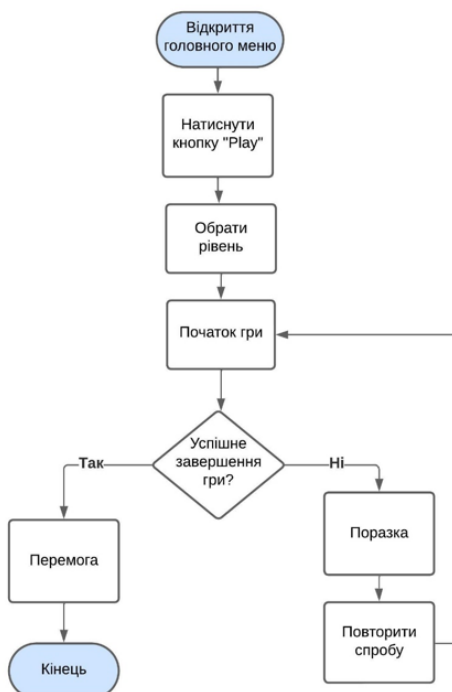
8

## Діаграма класів



9

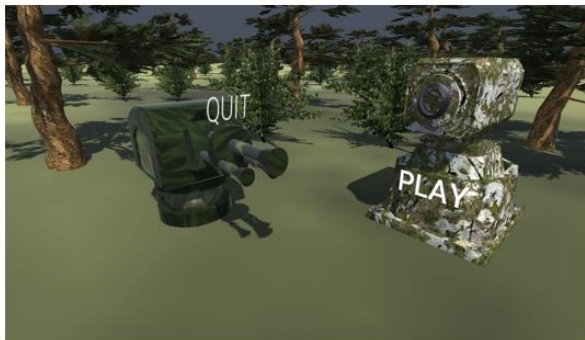
## Загальний алгоритм гри



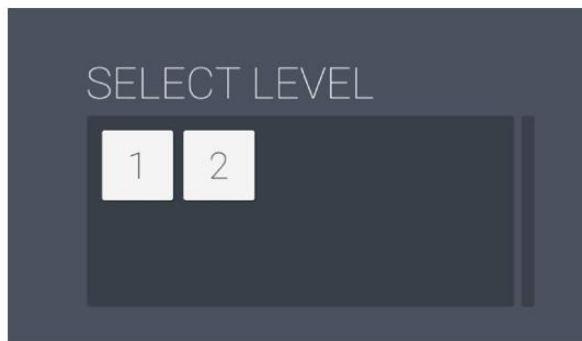
10



## Екранні форми



Головне меню



Меню вибору рівня

11

## Екранні форми



Ігрова мапа з  
протівниками



Розміщення  
захисних споруд

12

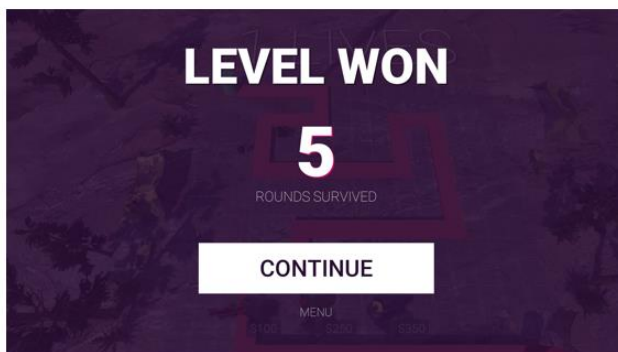
## Екранні форми



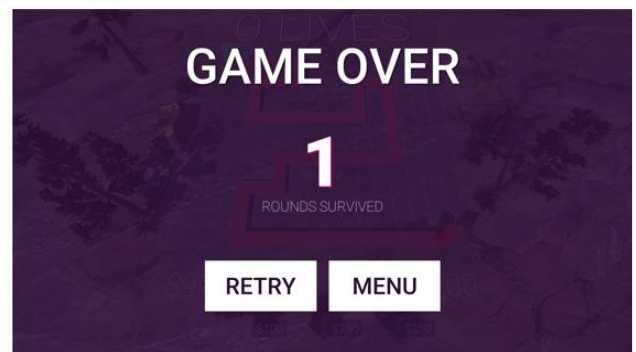
Взаємодія з захисною спорудою

13

## Екранні форми



Меню програшу



Меню перемоги

14

## Результати роботи гри



15

## **Апробація результатів дослідження**

1. Коваленко О.М., Замрій І.В. Інноваційні підходи до розробки гри: Використання штучного інтелекту для оптимізації геймплею: Матеріали Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в ІКТ». Збірник тез. 24.04.2024, ДУІКТ, м.Київ. К.:ДУІКТ –ст. 341-343.
2. Коваленко О.М., Замрій І.В. Психологія дизайну: як вивчення людської поведінки впливає на UI/UX: Матеріали Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в ІКТ». Збірник тез. 24.04.2024, ДУІКТ, м.Київ. К.:ДУІКТ ст. 133-135.

16

## Висновки

1. Проаналізовано та обґрунтовано актуальність обраної теми та визначено, що ігровий ринок перенасичений складними іграми, що потребують багато часу для успішного завершення рівня або завдання, тому є потреба в проектах, які підвищать зацікавленість до ігрової індустрії та жанру Tower Defense.
2. Проаналізовано аналоги в жанрі Tower Defense, зокрема: Kingdom Rush, Deathtrap, Revenge Of The Titans, визначено недоліки та переваги. При аналізі аналогів виявлено, що сучасна індустрія потребує новинок, які спростять перше ознайомлення новачка з жанром Tower Defense.
3. Досліджено програмні засоби та інструменти, які можуть бути використані при розробці ігор в жанрі Tower Defense, в якості засобів було обрано ігровий двигун Unity та Microsoft Visual Studio для написання коду мовою програмування C#.
4. Реалізовано функціонал та інтерфейс, який допоможе отримати перший досвід з використання гри, так наприклад реалізовано появу противників на мапі та проходження їх шляху, а для успішного виконання завдання недопущення їх до кінцевої точки, розроблено захисні споруди, що допоможуть знищити противника.

17

Дякую за увагу!

18

ДОДАТОК Б  
ЗАГАЛЬНИЙ АЛГОРИТМ ГРИ

