

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

## **Пояснювальна записка**

до бакалаврської роботи

на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ГРИ У ЖАНРІ TOWER DEFENSE НА UNITY ЗА  
ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ C#»**

Виконав: студент 4 курсу, групи ПД-41  
спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності/спеціалізації)

Коваленко В.М.

(прізвище та ініціали)

Керівник Трінтіна Н.А.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

Київ –2022

# ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

## НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 року

### ЗАВДАННЯ

#### НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

#### КОВАЛЕНКУ ВОЛОДИМИРУ МИКОЛАЙОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка гри у жанрі Tower Defense на Unity за допомогою мови програмування C#»

Керівник роботи: Трінтіна Н.А., к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «16» лютого 2022 року № 22.

2. Строк подання студентом роботи 03.06.2022 року

3. Вхідні дані до роботи

Методи розробки ігор;

Технічна література, з питань, пов'язаних з іграми в жанрі Tower Defense;

Офіційна документація до Unity та C#.

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

4.1 Етапи розвитку ігрової індустрії.

4.2 Перелік ігрових жанрів.

4.3 Технології для розробки ігор.

4.4 Вимоги та оцінка якості гри.

4.5 Опис використаних технологій.

4.6 Опис функціоналу гри.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

1. Титульний слайд

2. Мета, об'єкт та предмет роботи

3. Актуальність

4. Аналіз аналогів

5. Технічне завдання

6. Програмні засоби та інструменти реалізації

7. Схема взаємодії з грою

8. Практичне застосування

9. Наукова новизна та практична значимість

10. Апробація результатів дослідження

11. Висновки

12. Кінцевий слайд

13. Дата видачі завдання 11.04.2022

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	11.04-14.04	Виконано

2	Дослідження аналогів та актуальності додатку	15.04.-17.04.	Виконано
3	Аналіз та вибір інструментів для розробки додатку	18.04-22.04	Виконано
4	Проектування та реалізація	23.04-05.05	Виконано
5	Вступ, висновки, реферат	06.05-10.05	Виконано
6	Розробка обов'язкових демонстраційних матеріалів	10.05-12.05	Виконано
7	Попередній захист роботи	16.05-01.06	
8	Здача роботи	03.06	

Студент \_\_\_\_\_ Коваленко В.М.  
( підпис ) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_ Трінтіна Н.А.  
( підпис ) (прізвище та ініціали)





## РЕФЕРАТ

Текстова частина бакалаврської роботи 61 с., 42 рис., 20 джерел.

TOWER DEFENSE, ГРА PLANTS VS ZOMBIES, ІГРОВИЙ ДВИГУН UNITYЮ, ГРА KINGDOM RUSH, ІГРОВА ІНДУСТРІЯ, ВІДЕОГРА, ГРА DUNGEON DEFENSE

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

*Предмет дослідження* – програмний продукт в жанрі Tower Defense.

*Мета роботи* – підвищення зацікавленості гравця в ігровому процесі гри жанру Tower Defense за рахунок впровадження нових противників та покращення захисних споруд.

*Методи дослідження* – методи розробки ігор за допомогою існуючих додатків, ігрових двигунів та мов програмування.

У даній роботі проведено аналіз існуючих ігор, таких як Kingdom Rush, Defense Zone 3 HD, Plants vs Zombies, Bloons TD 6, Defenders 2: Tower Defense CCG, Dungeon Defense, методів та функцій роботи цих ігор.

Загальною проблемою цих ігор є постійна потреба у використанні мережі інтернет для з'єднання, що може бути не зовсім зручним. Так, якщо з'єднання відсутнє гравець не зможе використовувати гру. Також важливим недоліком цих проектів є те, що вони складні при першому ознайомленні новачка з жанром.

Особливістю розробленої гри є можливість грати в неї без інтернет з'єднання, на будь-якому пристрої, адже вона добре працює навіть на простому ПК та простота при проходженні рівнів, що допомагає новачку зацікавитись у жанрі Tower Defense. Гра була розроблена за допомогою ігрового двигуна Unity та мови програмування C#, які показують гарну взаємодію між собою.

Збереження всіх видів даних, забезпечується за допомогою файлів гри. Саме вони мають всю потрібну інформацію.

Отже, розроблено гру, яка працює за загальними правилами жанру Tower Defense та допомагає спростити перше ознайомлення новачка з жанром.

## ЗМІСТ

<b>ВСТУП</b> .....	14
<b>1. ОГЛЯД ІСНУЮЧИХ ПРОДУКТІВ В ЖАНРІ TOWER DEFENSE</b>	<b>12</b>
1.1 Поширення ігрової індустрії та її значення в сучасному світі .....	12
1.2 Змагання у сфері комп'ютерних ігор, як новий етап розвитку .....	18
1.3 Жанри відеоігор.....	20
1.4 Огляд аналогів .....	23
1.5 Інструменти розробки ігор .....	32
1.6 Розробка ігор за допомогою двигуна Unity.....	35
1.6.1 Ігровий двигун Unity.....	35
1.6.2 Можливості ігрового двигуна.....	36
1.6.3 Плюси та мінуси двигуна Unity.....	39
1.6.4 Розробка ігор за допомогою мови програмування C# .....	40
1.7 Постановка завдань дослідження .....	42
<b>2. РОЗРОБКА СТРУКТУРИ ГРИ В ЖАНРІ TOWER DEFENSE</b> .....	<b>43</b>
2.1 Завдання для розробки гри .....	43
2.2 Моделювання об'єкту програмування .....	44
2.2.1 Діаграма прецедентів для гри .....	44
2.2.2 Структура даних проекту .....	46
<b>3. ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ</b> .....	<b>49</b>
3.1 Оцінка, планування та зберігання гри.....	49
3.2 Набір використаних інструментів при розробці гри .....	50
3.3 Функціонал гри.....	52
<b>4. ПРИКЛАДИ ВИКОРИСТАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ</b>	<b>55</b>
4.1 Опис інтерфейсу гри та ігрового процесу .....	55
4.2 Результати апробації та подальший розвиток гри .....	59
<b>ВИСНОВКИ</b> .....	<b>60</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ</b> .....	<b>Ошибка! Закладка не определена.</b>



## ВСТУП

*Обґрунтування вибору теми та її актуальність:* у зв'язку з масштабністю ігрового суспільства можна зрозуміти, що є велика потреба в розробці ігор різних жанрів для задоволення потреб користувачів. Але на даний час не в кожному жанрі є гідні представники, які дають змогу отримати всю повноту емоцій за проведенням свого часу виконуючи поставлені завдання та ознайомитись з жанром Tower Defense.

Для поліпшення ігрового досвіду новачка необхідно розробити функціонал, який зможе конкурувати з наявними аналогами на ринку ігор та забезпечувати простоту при першому ознайомленні з жанром.

*Ступінь вивчення проблеми:* На сьогодні існує багато продуктів, які виконані за усіма правилами жанру Tower Defense ігор. Але не всі продукти можуть забезпечити простоту для ознайомлення з жанром та забезпечити гру функціоналом, який задовольняв би усіх користувачів, адже завдання будь-якого проекту поширити своє використання на якомога більшу кількість користувачів. До таких ігор належать Plants vs Zombies, Kingdom Rush Vengeance, Bloons TD 6 та інші, але навіть серед таких гігантів індустрії є невелика проблема, яка заважає сприяти розширенню ігрового суспільства в жанрі Tower Defense. Це те, що ці проекти виконані у своїх своєрідних стилях, які є не зовсім цікавими деяким групам користувачів та складністю у використанні для новачка жанру.

Існуючі ігри звісно дозволяють користувачу скоротити час, відпочити після важкого робочого дня, але не всі ігри підходять для цього простого вбивання часу, деякі з них можуть потребувати від гравця багато витраченого часу для виконання поставленого завдання, тому потрібно розвивати ігрову індустрію насичуючи її новими простими в ознайомленні та проходженні розважальними іграми.

*Об'єктом дослідження* є процес гри в жанрі Tower Defense.

*Предметом роботи* є програмний продукт в жанрі Tower Defense.

*Метою роботи* є підвищення зацікавленості гравця в ігровому процесі гри жанру Tower Defense за рахунок впровадження нових противників та покращення захисних споруд.

*Завданням роботи* є розробка першої технічно-демонстраційної версії гри, яка дозволяє запустити та дослідити її.

*Методика дослідження:* По перше, потрібно було визначити перелік функцій, які потрібні для правильної роботи гри, вибрати стек технологій, який надасть змогу створити конкуруючий продукт з забезпеченням працездатності та можливістю подальшої підтримки продукту, для його оновлення та додавання нового функціоналу або виправлення недоліків.

Розуміючи те, що продукт буде працювати на операційній системі Windows було обрано платформу розробки Unity з мовою програмування C#, такий вибір дозволяє створити повністю працездатний продукт, який буде гарно оптимізований. Також після вдалого запуску проекту на ОС Windows, можна буде здійснити розробку версії гри для мобільних платформ на операційних системах Android та iOS.

Результат розробки проекту можна буде запускати традиційним шляхом, відкриваючи .exe файл та встановлюючи гру на жорсткий диск.

*Наукова новизна роботи:* Таким чином, наукова новизна полягає в створенні проекту, який буде мати повноцінний функціонал, який забезпечуватиме працездатність гри та власний стиль.

*Практична значущість результатів:* Даний продукт може бути використаний новачками для полегшення першого входу до ігрової індустрії та жанру Tower Defense.

## 1. ОГЛЯД ІСНУЮЧИХ ПРОДУКТІВ В ЖАНРІ TOWER DEFENSE

### 1.1 Поширення ігрової індустрії та її значення в сучасному світі

Загалом, ігрова індустрія почала свій розвиток ще за 50 років до нашого часу. Перші ігри розроблялися не як розвага, а скоріше з питання наукових цілей, оскільки на початку комп'ютери були дуже великими, дорогими і з'являлися вони в освітніх та наукових установах. Комп'ютерні ігри набули більш широкого попиту з появою ігрових консолей та перших персональних комп'ютерів, після того, як вони стали більш дешевшими. Тільки після розвитку популярності ігор їх почали також розробляти з комерційних питань. Зараз компанії, які займаються розробкою ігор заробляють дуже великі кошти. Нині ж про розвиток ігор компанії заробляють цілий стан.

Так, наприклад, доходи від мобільних ігор за 2017 рік зросли — до \$50,4 млрд. На даний час, ринок ігор для персональних комп'ютерів показує значний приріст — там доходи зросли з \$29,4 до \$32,3 млрд. Ігри для консолей закінчили 2017 рік трохи гірше — на позначці у \$33,3 млрд, але це більше ніж у 2016 році майже на 4%. Кращими країнами у зростанні доходів за підсумками 2017-го стали Японія та Китай — це країни, які мають найбільше гравців. Так, на рисунку 1.1 можна побачити динаміку росту доходів ігрової індустрії.

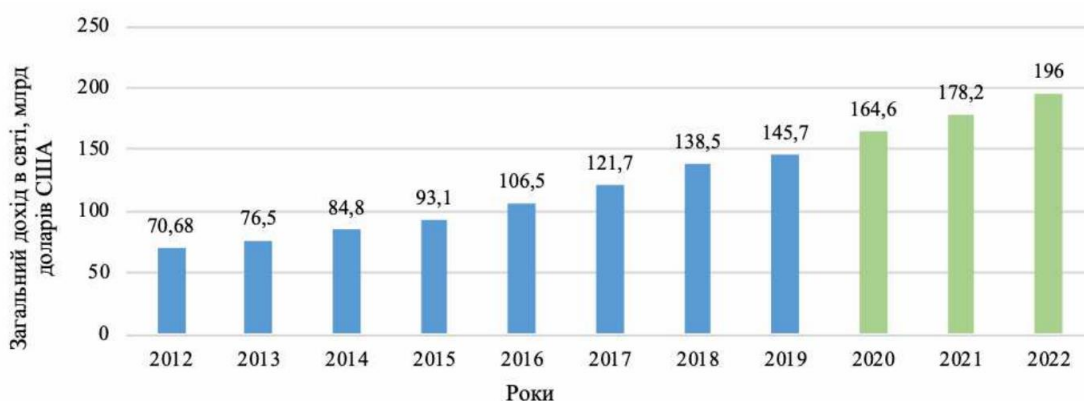


Рис. 1.1 - Загальний дохід ігрової індустрії у світі та прогноз на 2022 рр.

Тому потрібно детальніше розглянути історію розвитку ігор і зазначити важливі ігри, які стали передовими для розвитку ігрової індустрії. Вперше світ побачив першу подобу комп'ютерної гри 1942 року. Однією з версій початку ігрової індустрії можна вважати розробку двох людей - Томаса Голдсмита Молодшого та Східного Рі, саме вони створили перший ракетний симулятор. З їхньої розробки фактично можна вважати початок розвитку.

Початок активного розвитку перших ігор займає 50-60 роки минулого століття. Так гра ОХО (аналог хрестів-нуликів) була описана в докторській дисертації А. С. Дугласа. Але, незважаючи на це, ігри стали популярнішими саме серед студентів. Ігри того часу розроблювались для кожного комп'ютера. Як монітори були текстові термінали та векторні графічні дисплеї.

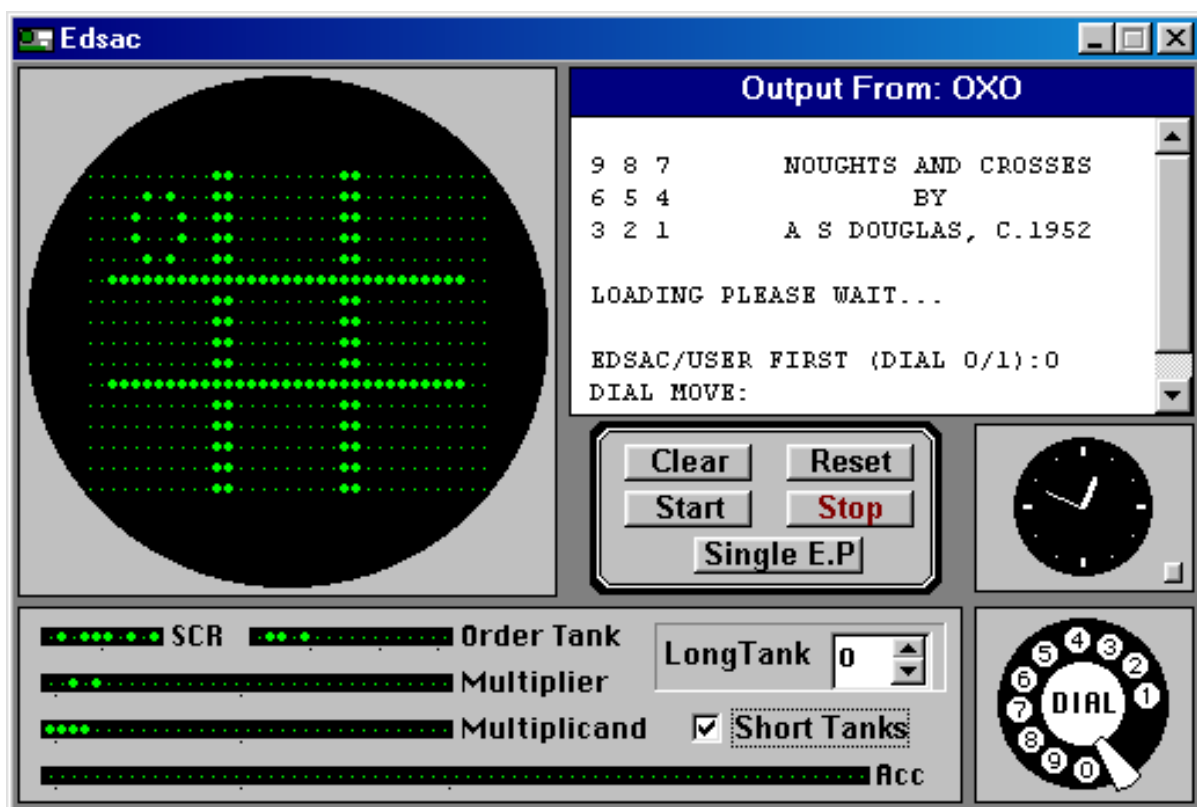


Рис. 1.2 – Гра ОХО, аналог хрестиків-нуликів

Під час наступного періоду 61 - 70-тих рр. гра SpaceWar була розроблена для комп'ютерів PDP-1, також було створено перший прототип комп'ютерної миші. Це

значною мірою вплинуло на подальший розвиток ігрової індустрії та комплектуючих, які вже так звичні в наш час.

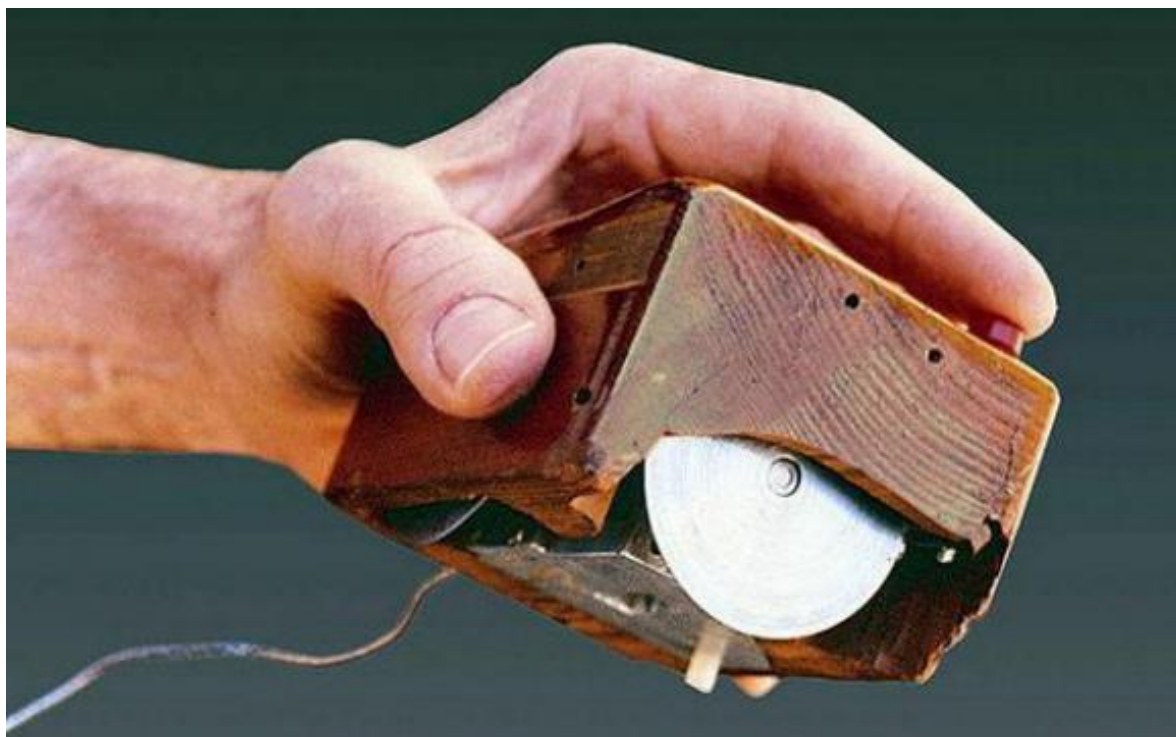


Рис. 1.3 – Перша комп'ютерна мишка

Згодом, через декілька років відома на той час компанія Atari створила свою першу домашню консоль Atari 2600, яка стала першою консоллю з тиражем в 1 млн проданих консолей.

Незабаром ігровий ринок був перенасичений свого роду клонами гри «Понга», проте в 78 році, як ковток свіжого повітря для ігрового ринку був гри Space Invaders. Всюди почали встановлюватися ігрові автомати, а поява франшиз таких, як Donkey Kong і Pac-Man розпочало, ще більший розвиток ринку. До 1982 року сфера ігор вже приносила більше грошей, ніж поп-музика і показ фільмів в кінотеатрах.



Рис. 1.4 – Відома гра Pac-man

У період з 1981 по 1990 рік консолі зазнають падіння популярності, тому що персональні комп'ютери стають доступнішими через нижчу ціну. Розпочинається розробка ігор з складнішою графікою, оскільки з'являються перші відеокарти, які допомагають полегшити навантаження на процесор. Значними датами того часу є 1983 р. - це початок ери 8-бітних консолей. Гігантами ринку консолей того часу були відомі консолі і нашого часу такі, як Nintendo Entertainment System, Sega Master System, PC Engine, Atari 7800. Тоді вперше з'явилася гра Final Fantasy, яка має вже з десяток нових частин та в яку грають і досі.

Але нажаль, ігрова індустрія зростала дуже швидко, так що забезпечувати її стійкість виявилось проблематично.

Тому спасителем ринку консолей стала Nintendo. У 1985 році вони розробили консоль NES або Famicom в Японії, а на ринку СНГ відомий її клон – Dendy. Компанія розраховувала на високоякісні гри і маркетинг. Такі ігри, як Duck Hunt, Excitebike і Super Mario фактично відродили консольний ринок.

У 1988 році відома компанія Sega представила свою розробку Sega Mega Drive. Причому піар кампанія була побудована навколо її потужності.

### Estimated Total Console Sales by Manufacturer (1970-2020)

Manufacturer	Home Console sales	Handheld Console Sales	Total Sales
Nintendo	318 M	430 M	754 M
Sony	445 M	90 M	535 M
Microsoft	149 M	-	149 M
Sega	64-67 M	14 M	81 M
Atari	31 M	1 M	32 M
Hudson Soft/NEC	10 M	-	10 M
Bandai	-	3.5 M	3.5 M

Рис. 1.5 – Орієнтований обсяг продажів консолей з 1970 по 2020 рр.

Трохи пізніше з'явився вже гігант нашого часу, компанія Sony, початком її великої історії ігрових консолей була перша PlayStation випущена в 1994 році. Замість картриджів вона вже використовувала CD диски, це було потрібно для збільшення обсягу пам'яті для деяких ігор. Вона ж стала першою консоллю, яка продала перші 100 млн проданих консолей. Надалі, саме Sony буде вносити корективи в подальший розвиток індустрії зі своїми консолями PlayStation 2 (DVD) і PlayStation 3 (Blu-Ray).

Також відома компанія Microsoft теж внесла свій вклад в геймінг і випустила DirectX Арі, яке спростило життя розробникам ігор і дозволило робити їх більш технологічними. Згодом корпорація використає букву «X» в назві своєї нової консолі.

Однак розвиток інтернету і мобільних телефонів сильно збільшили популярність і дохід ігрової індустрії з десятків до сотень мільярдів доларів. Це добре ілюструють два приклади:

Так у 2001 році Microsoft запускає сервіс XBOX Live, який за щомісячну плату надавав доступ до гри через інтернет і голосового чату. Це швидко стало обов'язковим пунктом багатьох користувачів консолі.

А у 2004 році Blizzard випускає MMORPG World of WarCraft і починає використовувати модель "Щомісячна плата" для доступу до гри. На піку своєї популярності вона досягла майже 14 млн передплатників.



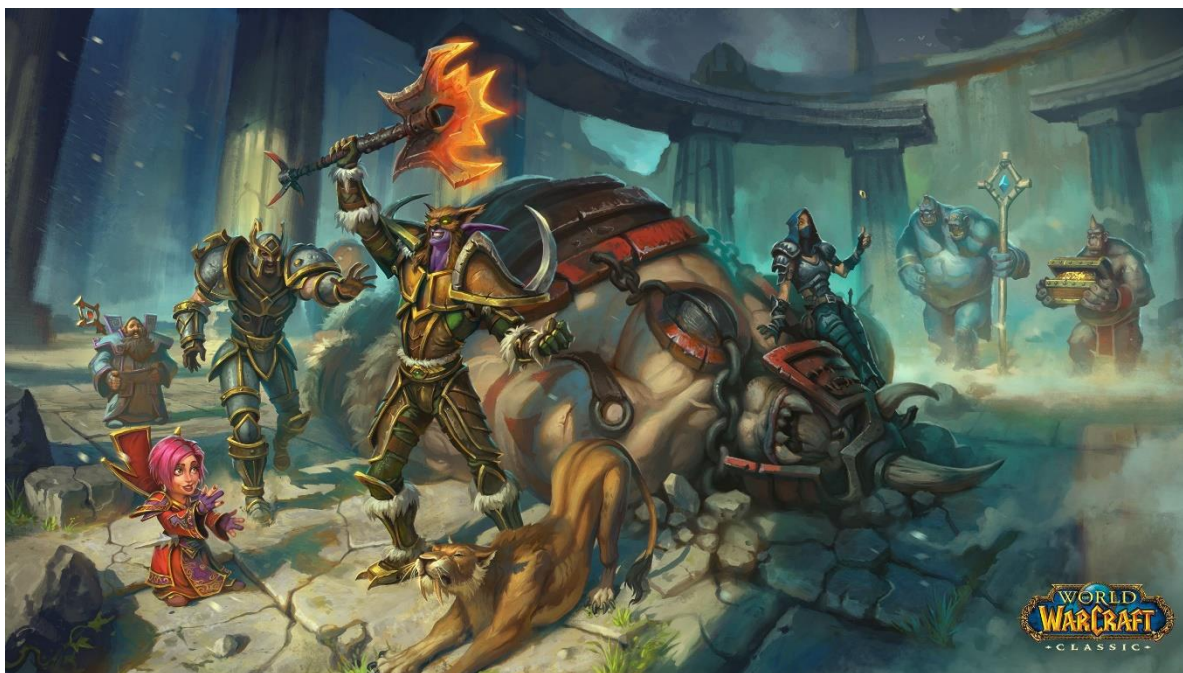


Рис 1.6 – Популярна онлайн гра World of WarCraft

Надалі провідні компанії передбачали потенціал в портативному геймінгу і намагалися його реалізувати. Nintendo зайняла своє місце з лінійкою Game Boy, а Nokia і BlackBerry додавали легкі ігри до своїх телефонів. Однак лідером прориву стала компанія Apple. Магазин App Store від них (незабаром Google теж розробить майданчик для Android) проклав великий шлях для розробників платних, безкоштовних і так званих free-to-play ігор.

Зараз Microsoft і Sony продовжують випускати нові консолі, паралельно розвиваючи сервіси ігор за підпискою.

У 2020 році у всьому світі налічувалось більше 2.7 мільярда гравців, і те, як вони продовжать витратити свої гроші, буде визначати історію відеоігор в тому вигляді, в якому вона зараз є.

Так, на зараз є дуже багато гігантів індустрії зі своїми найпопулярнішими іграми, по яким навіть знімають фільми (наприклад, World of WarCraft).



## 1.2 Змагання у сфері комп'ютерних ігор, як новий етап розвитку

Якщо прослідкувати за іграми на консолях, комп'ютерах та мобільних пристроях, зрозуміло, що ігри, які мають в собі командну гру та систему рейтингу, потребують збільшення проведення ігрового часу для досягнення якихось результатів. Для ігор на мобільному телефоні це актуально у Китаї та в інших країнах Азії, а також в західній частині нашого світу, де мобільні ігри з ранговою системою все частіше потрапляють до переліку дохідних брендів на ринку. Це значною мірою змінило відношення китайських видавців, які почали випускати та продавати свої ігри не на Схід, а на Захід. Велика кількість нових мобільних ігор, розпочатих успіхом PlayerUnknown Battlegrounds на ПК, призвела до зростання попиту ігор жанру Battle Royale на мобільних пристроях на Заході. У той же час, кіберспорт переходить у фазу зрілості.

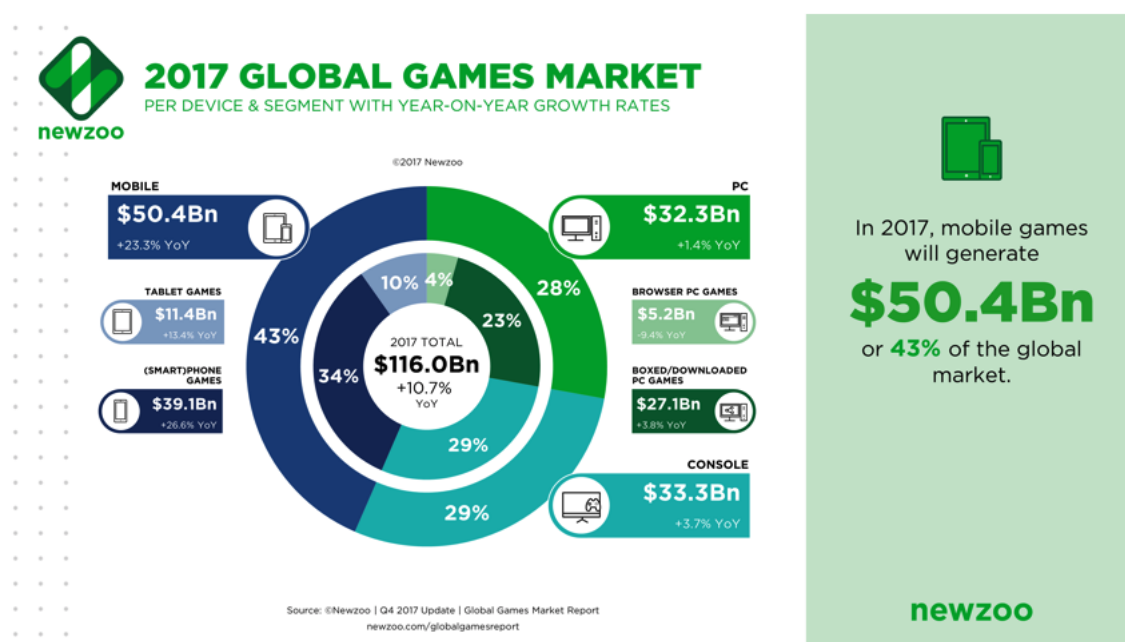


Рис. 1.7 – Підсумки індустрії за 2017р.

Найближчі роки мають вирішальне значення для того, як швидко кіберспорт перетвориться на багатомільярдний бізнес. Ключовими факторами є успіх місцевих

ліг, виконання правил професійними гравцями, розвиток нових форматів ігор та висока конкуренція.

Також значення має рентабельність команди та вплив галузі на розвиток медійної ігрової сфери, збільшення призових фондів та популярності тієї, чи іншої дисципліни. Залежно від того, як ці чинники будуть розвиватися протягом найближчого року чи двох, зростання кіберспорту може прискоритися і досягти десятків млрд доларів за оптимістичним сценарієм, або декількох млрд за базовим сценарієм розвитку цього ринку.

Раніше доходи від ігрового ринку та створення програмного забезпечення порівнювали з продажем квитків у кінотеатрах по всьому світу. Оскільки ігровий процес все більше наближається до традиційних медіа та розваг, є сенс говорити про те, що в найближчі декілька років обсяги грошових надходжень від ігор та інвестицій у них можуть перевищувати показники доходів від прокату фільмів.

Так, призовий фонд чемпіонату світу «The International» 2021 року з гри Dota 2 налічує 40 млн доларів, а команда, яка зайняла перше місце отримала 18 млн.



Рис 1.8 – Переможець чемпіонату «The International» 2021 року

Світовий кіберспортивний бізнес за різними оцінками вже становить від \$130 до \$150 млрд залежно від джерела звітності. Ця сума включає в себе рекламу,

спонсорство, товари та квитки на кіберматчі. За таких темпів зростання, доходи почнуть перевершувати спортивні вливання через декілька років. Якщо ми додамо доходи від консолей (наприклад, близько \$10 млрд у 2017-му) та ігрових персональних комп'ютерів та периферії до них (близько \$23 млрд), то можна буде сказати, що кіберспорт перевершує кіно, й традиційну спортивну індустрію. Фактично можна назвати народження цілком новітнього сектору сучасної економіки.

### **1.3 Жанри відеоігор**

Жанр гри визначається метою та функціональністю гри. Гра може належати як до одного, так і до декількох жанрів відразу, а в унікальних випадках - творити новий. Але в загальному виділяють 8 жанрів ігор, які також можуть поділятися, ще на піджанри.

Action - жанр ігор, успіх гравця в яких по більшому залежить від його реакції і здатності швидко приймати тактичні рішення. Ігровий темп таких ігор дуже динамічний і вимагає неабиякої уваги і швидкої реакції до подій, що відбуваються в грі. При цьому, в якості основного інструменту для забезпечення прогресу в грі, використовують будь-яку зброю. Сам жанр поділяється на такі піджанри: шутери, аркади, файтинги та стелс-ігри. Саме цей жанр є найпопулярнішим та займає близько 70% ринку відеоігор.

3D-шутер - в іграх даного типу гравець, діючи наодинці або з вірними друзями по команді, повинен знищувати ворогів за допомогою зброї ближнього бою і стрілецької зброї, для досягнення цілей на різних рівнях та локаціях, зазвичай, після досягнення поставлених грою цілей, гравець отримує нові завдання та переходить на наступний рівень. Ворогами звісно виступають різні цілі, такі, як бандити та інші «погані хлопці», а також всілякі інопланетяни, мутанти і монстри (напр. Doom, Serious Sam, Duke Nukem 3D, Half- life.).

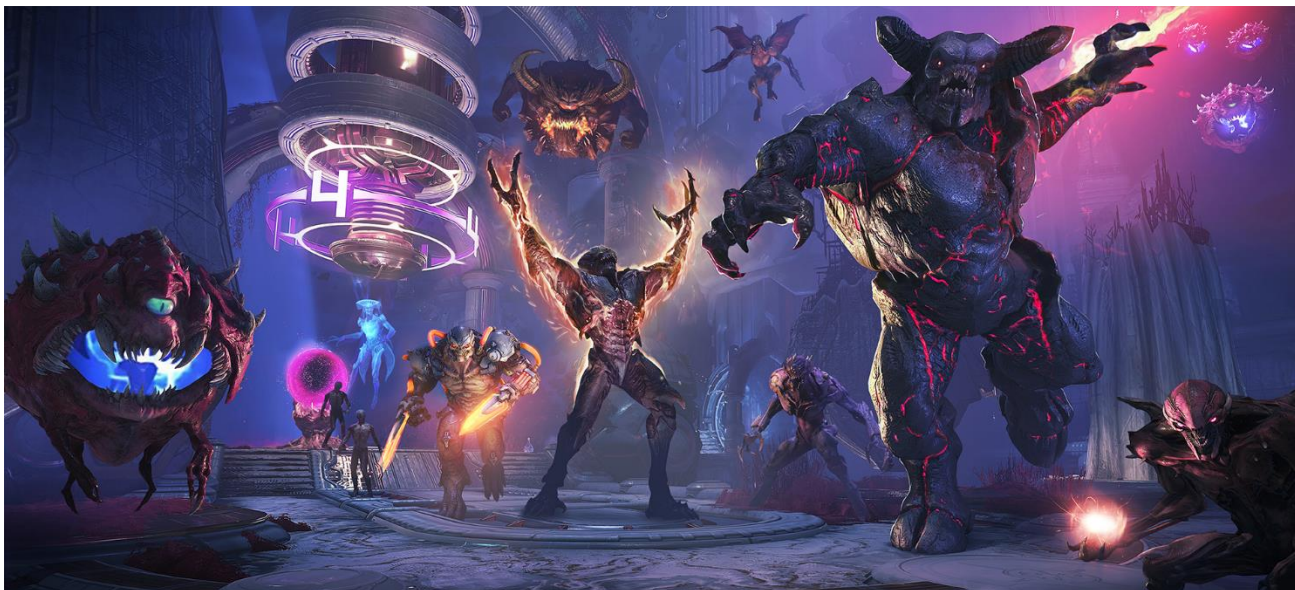


Рис. 1.9 – Популярний 3D-шутер Doom

Слешер - різновид ігор, дуже схожий на файтинги, але спеціалізується на використанні холодної зброї. В основному слешери розробляються для ігрових консолей, так як гра на геймпаді набагато більш підходить для такої гри – в ігровому процесі використовуються не більше 10 кнопок та їх комбінації роблять не можливим використовувати звичну клавіатуру та мишку.

Аркада - гра, в якій гравцеві користувач повинен діяти швидко, використовуючи саме свої рефлексі і реакцію. Ігровий процес простий і не змінний. В аркади часто присутня розвинена система бонусів: нарахування очок , поступово відкриваються певні елементи гри і т.д. Термін «аркада» сам по собі виник у часи ігрових автоматів , які встановлювалися в торгових центрах та ігрових клубах. Ігри були простими в освоєнні для залучення більшої кількості користувачів. Згодом такі ігри передислокувались на ігрові консолі і досі є основним жанром на них.

Файтинг - жанр ігор, основною механікою, яких є рукопашний бій персонажів в межах невеликого простору, в рамках так званої арени. Файтинг близький до жанра «побий їх усіх», проте між ними існують певні відмінності. Так, в більшості файтингів гравець перебуває в одній невеликій арені і не може вийти за її межі, а бій складається з 3-х або 5-ти раундів і має певну межу в часі. Також у



користувача в інтерфейсі при веденні бою присутні шкали для зображення життєво важливих показників персонажів.



Рис 1.10 – Популярний файтинг Mortal Kombat

Симулятор – ігри, що імітують управління різними процесами або транспортним засобом. Найчастіше слово «симулятор» використовується до комп'ютерних програм. За допомогою програмної реалізації, відтворюють інтер'єр кабіни апарату. Саме в таких іграх тренуються пілоти, космонавти, машиністи поїздів. Симулятор, це програмні та апаратні засоби, що створюють враження реальності, відображаючи справжні явища і властивості у віртуальному середовищі.

Комп'ютерна рольова гра або RPG – популярний жанр ігор, заснований на елементах ігрового процесу для традиційних рольових ігор. На сьогодні майже у кожній популярній та великій грі (так звані AAA ігри) присутні елементи RPG, для збільшення ігрового часу та цікавості гри, на рис. 1.8 приклад такої гри. У рольовій грі гравцю відводиться управління одним або декількома персонажами, кожен з яких має свої певні характеристики, здібностями і вміннями. У ході гри вони можуть змінюватися. Однією з важливих характерних характеристик ігрового процесу є підвищення здібностей персонажів за рахунок покращення параметрів та вивчення нових здібностей.



Рис. 1.11 – Популярна польська гра The Witcher 3 з елементами RPG

Стратегії - ігри, які вимагають планування і розробки власної стратегії для досягнення певної мети, наприклад, досягти перемоги у військовій операції. Гравець здійснює управління не тільки одним персонажем, а цілим підрозділом, або групою бійців. Стратегії діляться на лінійні або покрокові ігри, де гравці роблять хід по черзі, і кожному гравцеві відводиться необмежений або обмежений час на свій хід, і стратегічні лінійні ігри в реальному часі, в яких всі гравці виконують свої дії одночасно.

Пригода - гра-розповідь у якій герой просувається по сюжету і взаємодіє з ігровим світом з використанням різних предметів і спілкуванням з квестовими персонажами, а також можливим рішенням різного роду задач.

#### **1.4 Огляд аналогів**

Звісно виконуючи проект, потрібно розуміти чого хоче гравець та які переваги та недоліки є вже у представлених на ринку додатків або ігор. Так для аналізу поточної ситуації було обрано декілька ігрових проектів виконаних у жанрі Tower Defense, це Plants vs Zombies 2, Swamp Attack, Kingdom Rush Frontiers, Defense Zone 3.

Роблячи аналіз ігор в цьому жанрі можна зрозуміти, що популярною і основною платформою на якій виконані майже всі перелічені ігри є Android та iOS. І це не дивно тому, що загалом мобільний геймінг розвивається дуже швидко і це є дуже зручно, бо легше взяти з собою телефон до кишені та пограти в будь-якому зручному місці, ніж грати на ПК, чи на приставці. Але недооцінювати стаціонарні платформи не можна, адже ними користуються дуже багато людей, які так само люблять геймінг на стаціонарних платформах та не хочуть замінювати його на мобільні. Але, це не завадить для того, щоб оцінити переваги та недоліки цих ігор, і використати їх при розробці власного проекту на ПК.

### **Plants vs Zombies**

Перша гра, Plants vs Zombies, є дуже популярною у всьому світі, вона дарує радість майже кожному гравцю, незалежно від віку та статі. Звісно, є і хейтери, які скаржаться на гру через її простоту, але це є і тим пунктом, який влаштовує більшість.

Як такого сюжету в грі немає. Гравець просто повинен облаштувати свій захист використовуючи рослини та гриби, які мають певні здібності та характеристики, що допоможуть від хвиль зомбі, які намагаються потрапити в звичайний одноповерховий будинок фермера. Але не все так просто, як здавалося б, залежно від типу рослин, гравець повинен розумно розмістити різні захисні рослини, щоб противник не досяг своєї цілі.

Звісно і противник в вигляді зомбі має певні здібності, деякі можуть пригати, плавати, літати і враховуючи це потрібно правильно все зробити. Тому геймплей є цікавим, але не простим.



Рис. 1.12 – Ігровий процес Plants vs Zombies

І звісно розробники не обмежились проходженням однієї кампанії і додали багато різних міні ігор та головоломок, які допомагають забезпечити більший час ігрового досвіду гравця. Але це використовується не тільки для забезпечення позитиву цього гравця, а й для отримання більшого заробітку компанії розробника. Так у безкоштовній версії гри часто можна побачити надокучливу рекламу, яку можливо прибрати тільки після внесення коштів, для отримання платної версії гри. Але після оплати можуть бути доступні і нові ігрові режими та різні покращення, які звісно роблять неправильний баланс сил між гравцями та можуть псувати враження від гри. Гравець який грає на безкоштовній версії гри буде значно слабший та не зможе швидко досягати певних цілей, для цього йому потрібно буде витрати набагато більше власного часу.

Проаналізувавши гру, можна зрозуміти, що в цілому це дуже гарний проект, який виконаний на високому рівні, але як і всюди, в нього є свої недоліки та переваги:

#### **Недоліки:**

- Залежність гри від внесення додаткових коштів, для отримання більшої кількості контенту та досягання певних цілей;



- Гра незважаючи на свою простоту, може здатись комусь складною, адже потрібно правильно побудувати свій захист, з захисних споруд різного виду, під різного ворога;
- Велика кампанія;
- Відсутність локалізації українською.

### Переваги:

- Геймплей, який приносить радість гравцю;
- Цікаві ігрові механіки та режими;
- Оригінальний погляд на жанр;
- Можливість грати не одну годину.

### Swamp Attack

Далі було розглянуто гру Swamp Attack, яка виконана по такому самому принципу, як і попередня описана гра. В центрі сюжету ситуація, в котру потрапив головний персонаж. Звичайна людина в білій майці та червоній кепці, яка опинилась наодинці і своєму будинку на болоті і його потрібно захистити від противників, які бажають захопити цей будинок. Гравцю потрібно буде захищати будинок від різних тварин, які були заражені вірусом та стали вести себе агресивно.

Сама гра виконана з новими оригінальними функціями, які на перший погляд не досить схожі на звичайний Tower Defense, але це не так, адже основною задачею будь-якого Tower Defense, є захистити певну точку, або споруду.



Рис. 1.13 - Swamp Attack

Для успішного відбиття атак гравцю потрібно клацати по екрану, після чого ігровий персонаж буде стріляти по ворогу, який йде на нього з правого кутка. Щоб захиститись у гравця є три види озброєння, це зброя, вибухівка та захисні споруди, які вбивають противника самостійно, без допомоги гравця, вони звісно не такі сильні, як інші види озброєння, але значно покращають виконання основного завдання. Звісно в будь-якій грі в якій виникає потреба в покупці нового озброєння, є недоліки, це потреба в витраті великого обсягу часу, для покращення свого озброєння та досягання поставленої мети. Також проблемним місцем гри є те, що для проходження завдань потрібна енергія, яка додається з часом, але її звісно можна купити за реальні кошти (Рис. 1.14).



Рис. 1.14 – Ігровий магазин з озброєнням

Отримавши досвід у цій грі, можна сказати, що вона виконана якісно та приносить задоволення за проведеним в ній часом. І звісно, як у кожній грі в ній є свої недоліки та переваги:

**Недоліки:**

- Залежність гри від внесення реальних грошей;
- Потреба у енергії для того, щоб грати, яка займає досить багато часу;
- Складність гри;

**Переваги:**

- Великий вибір різного озброєння для захисту;
- Цікавий ігровий процес;
- Гарна система покращення озброєння;
- Різноманітність противників та босів, які покращують ігровий досвід.

**Kingdom Rush**

Kingdom Rush Frontiers, гра створена в середньовіковому стилі, яка працює на мобільних та стаціонарних платформах. В основі гри є сюжет, який розділений на карти, з різними скульптурами, будівлями та рослинністю зі зміною погоди та пори року.



Рис. 1.15 – Ігрова мапа одного з рівнів гри Kingdom Rush Frontiers

На кожній карті є спеціальні місця з яких виходять вороги та дорога, якою вони йдуть до кінцевої точки. Біля їх шляху також є спеціальні місця в яких можна

розмістити захисні споруди з воїнами, всього їх чотири виду – лучники, маги, воїни та інженери. Вони мають свої характеристики та певні можливості. Також у захисних споруд є покращення, які відкривають нові можливості та покращують воїнів.

Сам геймплей базується на розумному розміщенні захисних споруд та правильній взаємодії певних воїнів, саме вони повинні вбивати ворогів, які йдуть по дорозі до кінцевої точки. Складність полягає в тому, що вороги йдуть не тільки з однієї точки, а з декількох і ресурсів, які потрібні для побудови споруд не вистачає, тому потрібно розумно їх використовувати.

Також позитивною функцією та перевагою цієї гри, є те, що на карті є герой, яким гравець може управляти. Він має певні здібності, які гарно допоможуть у відбитті атак. Героїв в грі достатньо і можна підібрати потрібного для проходження певного складного завдання.

В грі є три режиму – простий, героїчний та хардкор. Вони відрізняються складністю та різними модифікаціями такими, як завдання без героїв і з одним життям, або безкінечною кількістю противників.

Так, отримавши ігровий досвід можна зрозуміти, що гра є цілком повноцінним проектом, з забезпеченням цікавих ігрових механік та функцій, вона має попит серед багатьох гравців світу, але є декілька недоліків:

**Недоліки:**

- Складність гри;
- Багато механік, які можуть здатись важкими та перенавантаженими;

**Переваги:**

- Великий вибір захисних споруд;
- Цікавий ігровий процес;
- Гарна система покращення озброєння;
- Різноманітність противників та босів;
- Наявність героїв, які допоможуть під час гри.



### Defense Zone 3

Defense Zone 3, гра з типовими правилами для жанру Tower Defense, де хвилями ворог наступає по певному маршруту, а гравець повинен правильно розмістити захист.

Гра виконана в стилі майбутнього. З самого початку гравцю будуть доступні турелі різного виду та з різними режимами вогню, звісно для них є покращення, використовуючи їх можна краще виконувати поставлене завдання з захисту. Також для кращого захисту гравець може використовувати міни, повітряну підтримку і навіть ядерну зброю, але це все з'являється з часом і досвідом. Всього доступно 8 видів зброї та 5 варіантів покращення для них.



Рис. 1.16 - Гра Defense Zone 3

Гра є складно, але любителям жанру Tower Defense вона подобається. Загалом одне бойове завдання йде до 20 хвилин, а перед ним з'являється реклама, яка в цілому не дуже заважає.

#### Недоліки:

- Складність гри;
- Малий вибір озброєння;
- Стилістика.

**Переваги:**

- Цікавий ігровий процес;
- Гарна система покращення зброї;
- Протяжність ігрової сесії.

**Захист вежі для дітей чи новачків: Bloons TD 6**

Практично у всіх частинах цієї гри, головними ворогами є солдати, лицарі, монстри, але в 6-й частині потрібно захищатися від повітряних куль. Звісно повітряні кулі не дуже відрізняються від іншого виду нападників, оскільки мають практично ті ж здібності, що й інші вороги в різних іграх. Гравець повинен розставити мавп-захисників, які за правильної побудованої тактики зможуть відбити атаки противників.

В грі є доступ до великої кількості рівнів, а також до різних доповнень. Перевагою гри є кооперативний режим та спеціальні завдання. Все це розроблено для збільшення ігрового часу користувача.



Рис. 1.17 – Гра Bloons TD 6

### **Петро захист вежі: Dungeon Defense**

Ця гра нагадує вище названу Plants vs. Zombies. Гравцю належить побудувати оборону з одного боку, а з іншого на нього будуть йти вороги. Відмінною рисою гри є те, що від гравця залежить набагато більше, ніж звичайна установка захисних споруд.

В грі є можливість мати різні види озброєнь та покращень для неї, що значно покращує оборонну спроможність та цікавість гри. Dungeon Defense виконана в ретро-стилі, яка розроблюється лише 1-м розробником.

Так отримавши досвід в таких проектах можна буде правильно скласти план та вимоги до майбутнього проекту, враховуючи недоліки та переваги від цих ігор.

### **1.5 Інструменти розробки ігор**

Для того, щоб створити успішний проект, який буде користуватися попитом у багатомільйонної аудиторії, потрібні фахові розробники різного типу, такі, як програмісти, дизайнери, художники, аніматори, продюсери та інші. Саме за їх участю проект проходить усі стадії розробки. При залученні такої великої кількості розробників звісно потрібно забезпечити їх належним програмним забезпеченням, яке буде задовольняти потреби сучасного світу розробок ігор.

#### **Ігровий двигун**

Розробка кожної гри залежить від ігрового двигуна. Наприклад, якщо говорити про мобільні ігри з 2D-графікою, то, можливо розробити власний двигун буде навіть краще. Але якщо починати розробку проекту у повному 3D, то сьогодні вже краще використовувати готові розроблені графічних двигунів.

Так одним із найпопулярніших ігрових двигунів є Unity. Створений у 2004 році групою розробників, але на зараз він перетворився на справжнього монстра ігрової розробки, який підтримує 25 апаратних платформ, включаючи розробку для персональних комп'ютерів, ігрових приставок, смартфонів та навіть smart телевізорів.

Unity має дуже велику кількість переваг. Він є безкоштовним для особистого користування. Студія розробників може отримати його безкоштовно, якщо офіційно зареєструється, як розробник ігор для приставок - PlayStation, Xbox або Nintendo.

Unreal Engine – потужний набір інструментів для створення 3D програм та ігор AAA-класу. Додатки розробляються за допомогою мови програмування C++. Мова дає гарні можливості для створення ігрових світів та оптимізації. З 2015 року Unreal Engine перейшов до безкоштовних додатків, за умови, що прибуток від додатків, створених за його допомогою не буде перевищувати 3000\$ за квартал. Ігровий двигун підтримує розробку для Windows, Linux, Mac OS, PlayStation та Xbox.

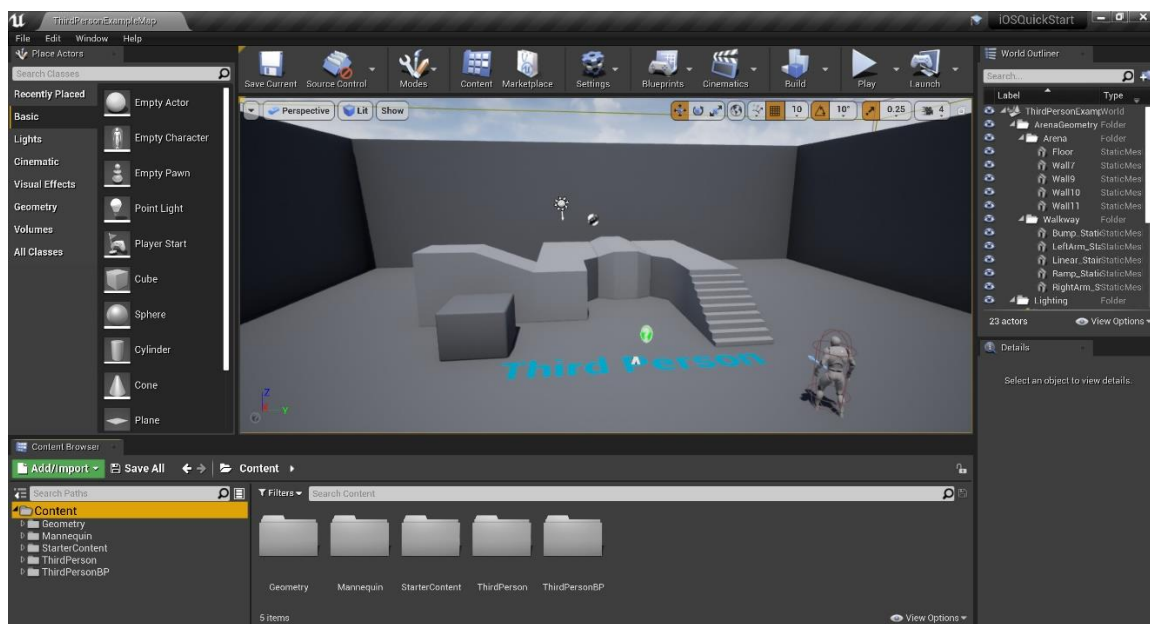


Рис 1.18 – Інтерфейс ігрового двигуна Unreal Engine

Було описано тільки два доступних для використання ігрових двигуна. Для того, щоб обрати кращий двигун для розробки власної гри, потрібно опиратись від наступних факторів двигунів: їх можливості, умови використання, ціна, підтримка сторонніх інструментів розробки, простота використання.



## Середовище 3D-моделювання

Напевно, одним з головних інструментів гейм-дизайнера є середовище 3D-моделювання. Навіть для ігор, які розробляються в 2D, сцени для неї в основному проєктуються саме у 3D. Одним з найвідоміших інструментів для такого моделювання є 3ds Max. Професійні дизайнери з навичками його використання дуже затребувані і дуже добре продаються на ринку праці. 3ds Max має велике значення, як інструмент ще тому, що для нього створено велика кількість плагінів, що дозволяють швидко створювати різні візуальні ефекти.

Також популярним 3D-редактором є і Мауа, який також відмінно підходить для створення кінематографічних сюжетів, включаючи розташування джерел світла, розміщення віртуальних камер і т.д. Навчитись розробці з Мауа не дуже просто, але результат того вартий. Для підтвердження цього є сотні голівудський студій, які використовують Мауа для створення спецефектів в кіно.



Рис. 1.19 - Інтерфейс середовища 3D-моделювання 3ds Max

## Засоби анімації

Навіть 2D-моделі, створені для анімації, треба "оживити", особливо коли гра має гарну, велику сюжетну лінію. Для того щоб це зробити, дизайнери переносять намальовані ними моделі в спеціальне програмне забезпечення, яке використовує

технологію skeletal. Фігурам персонажів та різних об'єктам додають «кісті», рухаючи якими, можна швидко створити анімації.

### **Додаткові графічні засоби**

Також додаткові графічні засоби, які можуть допомогти при створенні гри, мають великий список додатків. Так, наприклад може знадобитись інструмент для одягання на 3D-моделі текстур, що нададуть реалістичність. Для цього є чудовий додаток - Substance Painter.

Якщо проект потребує великої кількості рослинності, то заощадити величезну кількість ручної праці та витраченого часу допоможе додаток GrowFX, який налічує велику кількість вже готових шаблонів, засоби кастомізації та механізм перенесення готових моделей до інших редакторів.

Моделювати та малювати як у 2D, так і в 3D можна за допомогою додатку Zbrush, яка може бути основним або додатковим засобом обробки. Основною перевагою додатку є створення моделі, майже ліплячи її з глини.

Також не обійтись без популярного в усьому світі Photoshop, який забезпечує комфортну обробку вже багато років. Він допоможе майже на кожній стадії розробки ігор.

### **Засоби керування версіями**

Також, зрозуміло, що для забезпечення надійності зберігання проробленої роботи потрібно мати можливість контролювати версії та зберігати проект віддалено. Протягом розробки проект може змінюватись декілька разів і це потрібно контролювати. Для таких потреб також є спеціальні додатки, сервіси, наприклад, Perforce, яке часто використовується і в розробці ігор, і при створенні анімацій або GitHub.

## **1.6 Розробка ігор за допомогою двигуна Unity**

### **1.6.1 Ігровий двигун Unity**

На початку Unity призначався для розробки на операційних системах Mac, вже пізніше з'явилась можливість розробки під ОС Windows.

У 2008 році Unity вийшло оновлення з можливістю розробки під iOS, а у 2010 для Android пристроїв, тільки після цього Unity було оновлено під розробку для Xbox і Playstation.

Це повноцінний ігровий двигун, який передбачає, що процес розробки ігор буде відбуватись в редакторі, який вже влаштований в додаток. Багато популярних ігрових продуктів створені саме за допомогою цього двигуна: Hearthstone: Heroes of Warcraft, Age of Magic, Royal Blood та інші.

Це простий в роботі додаток, але в той же час дуже потужний, що дозволяє створювати та випускати 2D та 3D-ігри. Unity відкриває перед розробниками великі можливості своїх платформи для підтримки та монетизації розроблених ігор.

## 1.6.2 Можливості ігрового двигуна

### Робота з ресурсами

Для роботи з ресурсами Unity інтерфейс має власний редактор, до якого належать багато різних вікон, які можна розташувати на свій смак. Завдяки такому інтерфейсу можна розробку гри чи застосунка прямо в редакторі. До головних вікон відносяться: інспектор об'єкту, вікно попереднього перегляду, оглядач ресурсів проекту, огляд сцени та огляд ієрархії ресурсів.

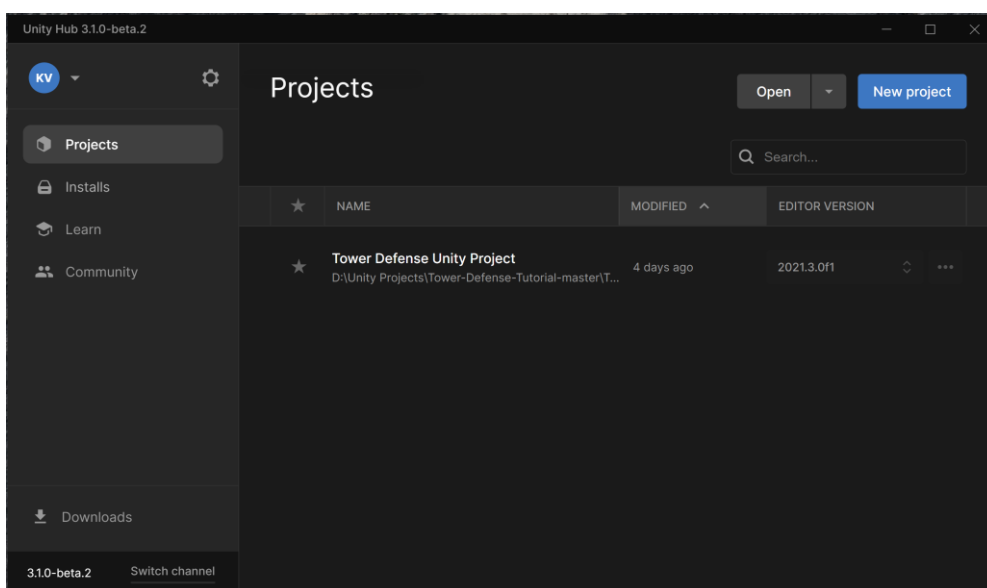


Рис. 1.20 – Інтерфейс перегляду проектів та версій Unity Hub

Проект в Unity розробляється поділом на сцени, це окремі файли, в яких розроблено свій ігровий світ з набором певних об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі, як деталі ігрового світу такі, як: персонажі, предмети довкілля, ландшафт, тощо. Так і ігрові об'єкти, які не мають фізичної оболонки, але змінюють поведінку інших об'єктів. Такі об'єкти можна рухати, змінювати розмір, обертати, використовувати до них скрипти. Так, кожен предмет на сцені має компонент Transform — він зберігає в собі координати предмету, повороту і розмірів по всіх трьох осях. У об'єктів з геометрією також присутній компонент Mesh Renderer, що робить модель видимою. Моделі можна об'єднувати в набори (ассети) для швидкого доступу до них.

В Unity існує підтримка фізики твердих тіл і тканин та фізики типу Ragdoll (ганчіркова лялька). Редактор має систему успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни батьківського об'єкта. До об'єктів також прикріплюються скрипти у вигляді окремих компонентів.

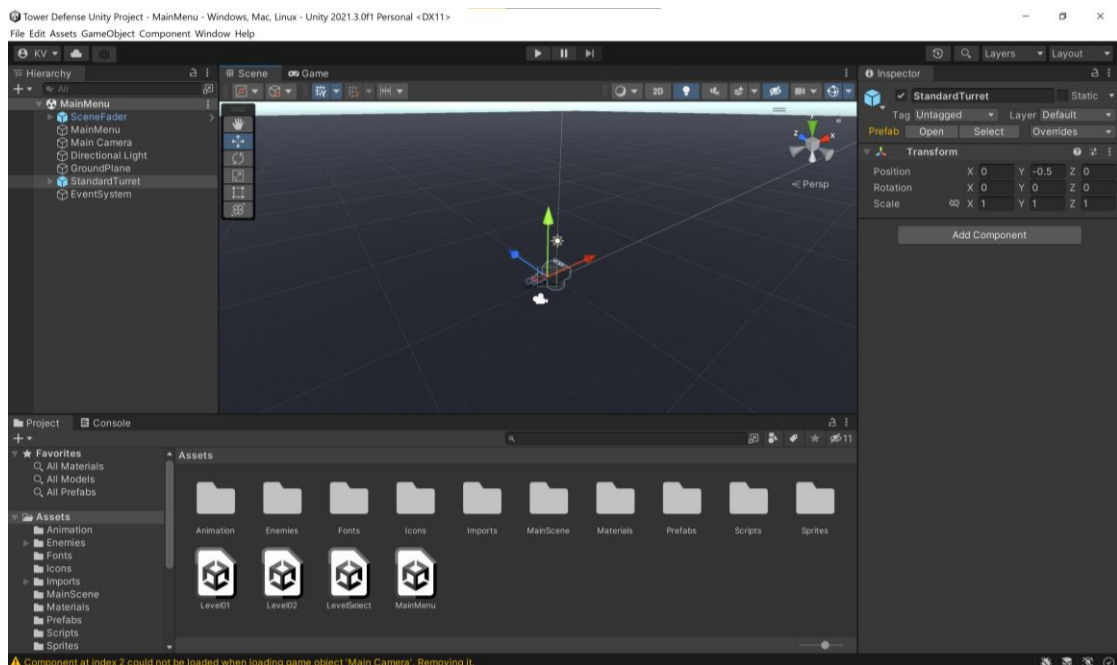


Рис. 1.21 – Інтерфейс редактору Unity

Для розробки 2D ігор в Unity зазвичай використовують спрайти. В 3D іграх використовуються тривимірні моделі, на які накладаються текстури та шейдери

(скрипти, за якими вираховується зміна кольору пікселів згідно заданих параметрів). В обох видах розробки використовуються системи часток для відображення елементів, таких як рідини чи дим.

Unity може забезпечити стиснення текстур і різні налаштування роздільності екрана під кожен платформу; забезпечує мапінг відображень, паралакс-мапінг, динамічні тіні за картами, рендер у текстуру та можливість обробки зображення, для забезпечення глибини чіткості, розмиття в русі, зернистості, відблиску віртуальних лінз.

### **Рендеринг**

Рендеринг зображення відбувається за допомогою віртуальної камери огляду. В області редактора сцена може розміщуватися, як завгодно, а при рендерингу — відносно того, як виставлено об'єкт камери. Сцена може мати декілька камер, які будуть рухатись за персонажем або за потрібною траєкторією. Вигляд з камери може подаватись, як двовимірно, так і тривимірно. Зазвичай на фоні зображено небо, яке утворюється за допомогою скайбоксу, але звісно можна розмістити і довільні зображення.

Графічний двигун використовує DirectX для Windows або OpenGL, як для Mac, так і Windows і Linux, OpenGL ES для Android та iOS. Також підтримуються reflection mapping, parallax mapping, bump mapping, screen space ambient occlusion (SSAO), динамічні тіні, які використовують shadow maps, render-to-texture та ефекти post-processing.

Unity підтримує великий список середовищ для 3D моделювання. В ігровий проект можна імпортувати об'єкти з цих додаків та виконувати налаштування за допомогою графічного інтерфейсу.

Для створення шейдерів можна використовувати ShaderLab, що підтримує шейдерні програми написані на GLSL або Cg. Для шейдеру можна написати декілька варіантів налаштувань під різні відеокарти, а Unity визначить найкращий варіант. Unity також підтримує фізичний двигун Nvidia PhysX.

## Скрипти

Реалізація скриптів побудована на Mono, це вільний відкритий проект з реалізації .NET Framework. Програмісти можуть використовувати також власну мову програмування Unity – UnityScript, яка подібна до JavaScript та ECMAScript, або C#.

З виходом версії 5.2 у 2015 році з'явилась підтримка редагування скриптів у середовищі Visual Studio.

Також Unity має власний магазин готових ассетів та плагінів. За допомогою нього можна розробляти власні проекти набагато швидше та з меншими витратами, адже ассети в Unity Asset Store може додавати будь який розробник.

Тому Unity підходить розробникам, які не мають великої команди, але вже готові взяти на себе більшість процесів розробки.

### 1.6.3 Плюси та мінуси двигуна Unity

#### Плюси двигуна Unity:

- Зрозумілий редактор та інструментарій: дуже легкий для освоєння, навчитися працювати та розробляти ігри може кожен новачок. А коли з'являться якісь складні питання або задачі, завжди можна знайти відповіді тому, що спілка розробників дуже велика та дружня. Існує багато форумів, відеороликів, книжок, уроків та курсів, які допоможуть навчитись розробці навіть школяру.
- Сучасний рівень графіки, який здатний конкурувати з платними двигунами. Звісно в Unity є набагато сильніші конкуренти, але як для першого та безкоштовного двигуна, новачку сподобаються такі можливості, як deferred освітлення, стандартний набір постпроцессингових ефектів, SSAO, прискорене опрацювання лайтмапів.
- Ігровий двигун Unity є умовно безкоштовним.
- Велике спілка розробників, багато випущених ігор.

- Unity Asset Store, де можна купити готові фрагменти коду, спрайти, асети та звуки.
- Unity дозволяє легко розробляти ігри та змінювати їх під різні ОС, такі, як Windows, Linux, OS X, Android, iOS, на консолі PlayStation, Xbox, Nintendo, на VR- та AR-пристрої.

### **Мінуси середовища розробки Unity:**

- Для розробки ігор на Unity потрібно мати хоча б якісь навички програмування, адже це не конструктор ігор, а середовище з підтримкою мов програмування, тому новачку потрібно бути готовим до освоєння мов програмування.
- Багато різних компонентів можуть здатись складними для освоєння, а розроблений продукт об'ємним. Це може бути проблемою, адже не кожен гравець любить скачувати великі за розміром ігри і не кожен має потужний ПК, якщо гра буде з дуже детальною графікою.
- Розробник не може отримати вихідний код власної гри, для цього потрібно звертатись до інженерів Unity.
- Немає інтеграції із зовнішніми сервісами та бібліотеками (наприклад, Facebook, Twitter), розробникам потрібно налаштувати це власноруч. С# для Unity

#### 1.6.4 Розробка ігор за допомогою мови програмування С#

Процес створення 2D та 3D ігор на Unity поділяється на два основні етапи:

- Побудова візуальної частини через Unity;
- Написання коду за допомогою мови програмування С#.

Мова програмування С# розроблена компанією Microsoft, це одна з найпопулярніших мов програмування.

Сам ігровий двигун Unity бере участь у розробці фізики, графіки, багатьох інших технічних елементів, щоб розробник гри міг зосередитись на розробці

концепції гри. Але без реалізації скриптів за допомогою їх програмування, гра буде не повноцінною, тому професійні розробники рекомендують для цього двигуна вибрати саме мову програмування C#.

C# вважається мовою програмування, яка дуже потужно розвивається. Для неї написано дуже багато документації, яка допоможе розібратись зі складними питаннями. Також фактично це єдина мова програмування, яку краще використовувати при розробці ігор на Unity.

Так, як Unity використовує інтегроване середовище розробки Mono, яке є кроссплатформеною реалізацією Microsoft .NET, тому C# виступає, як основна мова програмування, на якій написана платформа .NET і також всі бібліотеки Unity.

А згідно рейтингів DOU можна констатувати, що мова програмування C# в Україні на 3-му місці серед програмістів, що програмують в Україні. А у світових чартах на 4 місці по Popularity of Programming Language — це рейтинг який використовує Google Trends. Тому можна сказати, що C# є однією з популярніших мов програмування і вона буде тільки далі розвиватись далі і ставати ще більш потрібнішою.

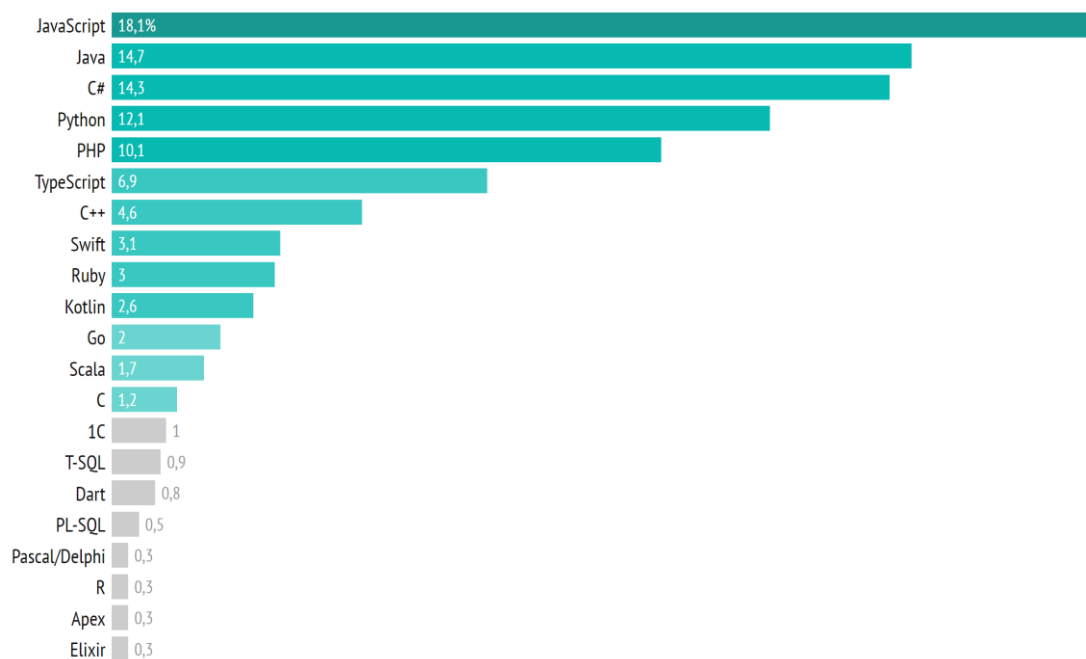


Рис. 1.22 – Рейтинг мов програмування згідно сайту DOU



## 1.7 Постановка завдань дослідження

Під час розробки концепції гри було визначено, що перед проектуванням та вибором інструментів та засобів реалізації, потрібно дослідити ігрову сферу та ознайомитись з додатками, які виконані у такому ж жанрі та мають подібний функціонал, до майбутнього проекту.

Так для ефективності розробки проекту, було поставлено перелік основних завдань з дослідження технологій розробки та інструментів, що в майбутньому дозволять реалізувати даний проект, а саме:

- Дослідити ігри конкурентів та зробити висновки, щодо переваг та недоліків і визначити вимоги до розробки власного проекту;
- Дослідити варіанти реалізації та перелік наявних технологій з розробки ігор, мов програмування, які використовуються для вирішення цього питання;
- Дослідити та вирішити, як реалізувати функції;
- Дослідити структуру проекту, необхідність реалізації певних скриптів на мові програмування C#;
- Дослідити можливість тестування основних сценарії за допомогою певних технологій, або з використанням тестувальників.

## 2. РОЗРОБКА СТРУКТУРИ ГРИ В ЖАНРІ TOWER DEFENSE

### 2.1 Завдання для розробки гри

На сьогодні можна сказати, що ринок комп'ютерних ігор має сильний підйом, так обсяг ігрової індустрії за 2020 рік склав 190 мільярдів доларів, враховуючи цю цифру можна зрозуміти, що обсяг ігрової індустрії сильно перевищив обсяг кіноіндустрії (100 мільярдів доларів) та обсяг індустрії спорту (75 мільярдів доларів).

Аналізуючи ці дані, що ігри вкрай затребувані серед різних груп гравців, і навіть те, що ігри стали звичною частиною життя багатьох людей, також створює велику потребу в створенні ігор різних жанрів для забезпечення потреб цих груп. Звісно розробка більшої кількості ігор створює деякі проблеми, які можуть впливати на розробників ігор. Такими проблемами, є:

- Перенасиченість ігрового ринку;
- Складність створення та реалізації нового функціоналу
- Ускладнення в процесі продумування нової ідеї, концепції гри.

Але, незважаючи на ці проблеми ігрові фанати потребують нових ігор, які зможуть вразити їх кращою графікою, реалізацією тих, чи інших функцій відносно конкурентів, створенням стилістично приємних інтерфейсів та обраних сеттингів гри та в цілому можливістю кожному гравцю обирати гру, опираючись на свої потреби та вподобання.

**Мета роботи** – підвищення зацікавленості гравця в ігровому процесі гри жанру Tower Defense за рахунок впровадження нових противників та покращення захисних споруд.

## 2.2 Моделювання об'єкту програмування

### 2.2.1 Діаграма прецедентів для гри

Основна мета створення для будь-якої системи, додатку або гри, це створення такого продукту, який буде забезпечувати основні потреби користувачів та допомагати виконувати ті, чи інші завдання. Для створення додатку, який задовольнятиме такі потреби, визначаються певні вимоги. Розуміючи всю складність питання, звісно можна було б просто зібрати всі потреби користувачів з різних форумів, соціальних мереж та створити «ідеальний» додаток, який допоможе у вирішенні їх проблем. Але зрозуміло, що такий проект буде тяжким в освоєнні та незрозуміло чи він взагалі допоможе користувачеві, чи гравцю і гри. Тому потрібно мати список певних, основних вимог та за допомогою відомих технологій їх опрацювання, створювати проект.

Так, зрозумілою та популярною технологією опису функціональних вимог можна назвати опис функціональності гри через варіанти використання (Use Case або прецеденти). Use Case діаграма, це опис послідовності дій, які буде використовувати додаток, система, чи гра та давати відповідь на певні дії користувача. Use Case допомагають визначити основний перелік функцій, які забезпечують відчутний результат при взаємодії користувача з додатком.

Також, Use Case використовуються для визначення основних функціональних вимог проекту та для зручності керування процесу розробки. Загалом варіанти використання описують всі види діяльності, які пов'язані з розробкою проекту. Так під час аналізу та проектування додатку, гри вони допомагають зрозуміти, що в кінцевому випадку хоче отримати користувач і як повинні працювати окремі елементи системи, для забезпечення певної функціональності.

Так, наприклад, при тестуванні, це дозволяє оцінити правильність реалізації вимог і допомагають провести їх перевірку.

При розробці діаграми варіантів використання є певні правила та основні її елементи, вона складається з акторів, для яких ця програма і розробляється і власне

Use Case, де описано, що актор отримує від певного варіанту використання. Актор позначається фігуркою чоловічка, а Use Case – овалом, а вони в свою чергою з'єднуються стрілочкою.

Метою даного проекту є розробка гри, яка буде працювати за всіма правилами жанру Tower Defense та мати типовий функціонал для цього жанру. Основним завданням цієї гри є виконання типових функцій розроблених самостійно для забезпечення гарного ігрового досвіду гравця.

Користувачами гри будуть звичайні геймери, зацікавлені жанром Tower Defense, які хочуть грати в такі ігри, але їх не влаштовують вже існуючі проекти. На рисунку 2.1 описано основні дії, які будуть доступні гравцю.

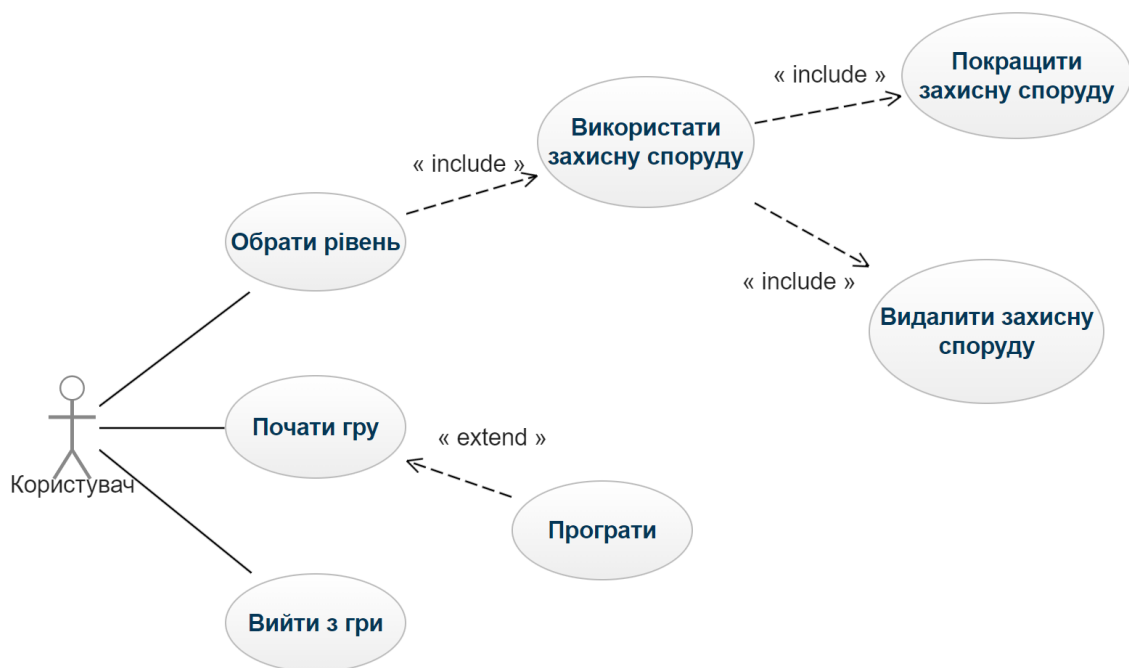


Рис. 2.1 – UML діаграма прецедентів проекту

Діаграма класів (class diagram) – UML діаграма, на якій представлені елементи моделі, класи з атрибутами і операціями, а також відносини між ними.

Діаграма класів створюється для представлення структури моделі проекту в термінології класів об'єктно орієнтованого програмування. Так до діаграми класів відносяться, також інтерфейси та відносини інші елементи.

Class – абстрактний опис однорідних об'єктів, що мають одні і ті ж самі операції, атрибути та відносини з об'єктами інших класів. Графічно клас при розробці діаграми класів малюється у вигляді прямокутника, який розділяється лініями на окремі секції. В них зазначаються спочатку ім'я класу, потім атрибути і операції.

При початку розробки діаграми, перші класи можуть позначатись прямокутником, де описано ім'я класу, а після продовження розробки проекту до нього можуть додаватись атрибути та операції класу.

Самі класи взаємодіють між собою за допомогою методів, а вже методи утворюють інтерфейс системи.

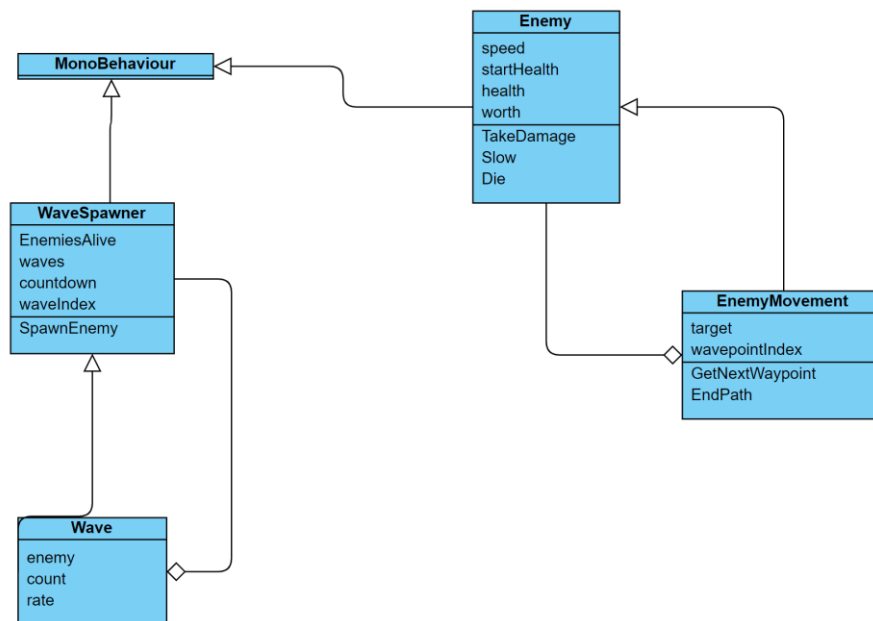


Рис. 2.2 – Діаграма класів

### 2.2.2 Структура даних проекту

Потрібно розуміти, що правильний підбір структури даних є надзвичайно важливим для ефективного роботи алгоритмів обробки даних. Добре побудована структура даних дозволяє оптимізувати час та пам'ять системи, для виконання найкритичніших операцій.

Оскільки проектом є розробка гри, яка буде використовуватись на ПК, як такої структури даних не буде. Всі дані будуть зберігатися в файлах гри. Так, наприклад, для зберігання шляху по якому будуть йти вороги використовується клас waypoints, в якому записано точки до яких вороги прямують і алгоритм, який знаходить 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 – Опис класу Waypoint, який реалізує пошук нових точок шляху

Для забезпечення інформації відносно самого противника було описано клас Enemy, в якому знаходяться дані про його стартову швидкість, описано інформацію про здоров'я, гроші отримані за його знищення та методи, які описують його стан отримання шкоди та смерті після знешкодження всього здоров'я.

```

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. 2.4 – Клас Enemy, з описом інформації про противника

Так для опису механізму руху противника написаний клас EnemyMovement, де написано, як саме та з якою швидкістю буде переміщуватись противник та описано метод, який знаходить на ігровій карті наступний waypoint та фінальну точку куди противник має дійти. Звісно для зручності створено різні класи, які містять інформацію про типові ігрові механіки, де певний елемент гри описаний згідно логіки його роботи. На рис. 2.5 представлено загальна кількість написаних класів, які описують функціонал та окремі елементи гри.

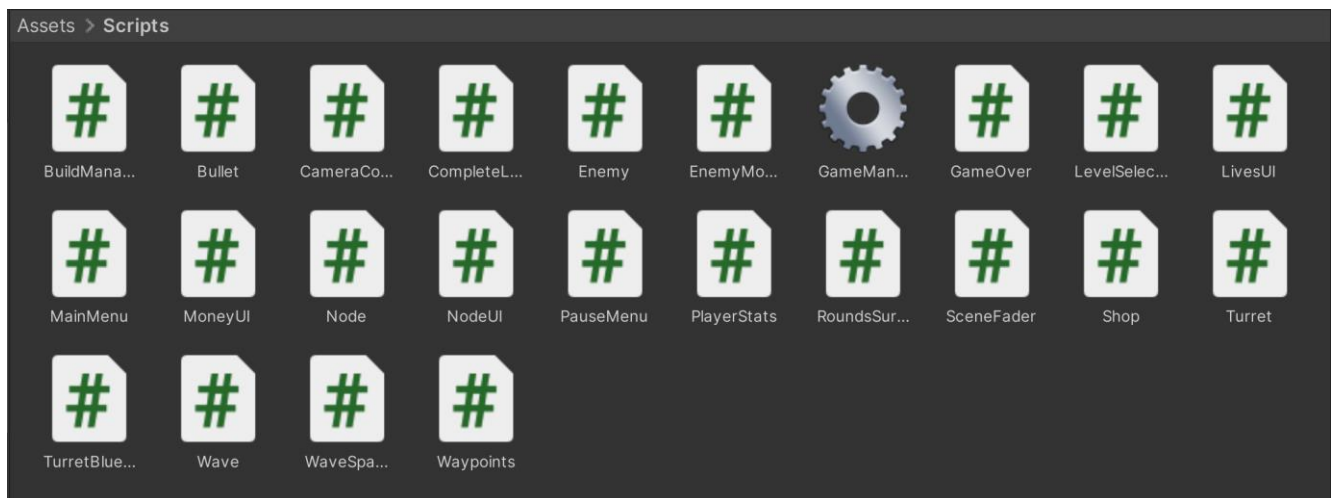


Рис. 2.5 – Всі класи в яких описаний функціонал гри



### 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ

#### 3.1 Оцінка, планування та зберігання гри

Згодом, після визначення основних вимог, архітектури системи, постало питання формування списку завдань, які звісно потрібно реалізувати. Метою розробки проекту було забезпечити мінімально необхідний функціонал та декілька рівнів, які продемонструють його. Звернувши увагу на те, що на розробку проекту були досить обмежені часові рамки, потрібно було використати цей час, для розробки дійсно гарно працюючого проекту, який покаже типовий функціонал та задовольнить основні вимоги та елементи, які будуть використані в грі.

Тому, було потрібно створити перелік першочергових задач, після виконання яких можна сказати, що проект задовольняє потреби MVP (Minimum Viable Product).

<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. 3.1- Функціональні вимоги

Задачі, які були описані, для зручності були перенесені до системи відстеження розробки Trello, ця система допомагає краще відстежувати статус розробки певного завдання (Рис. 3.2).

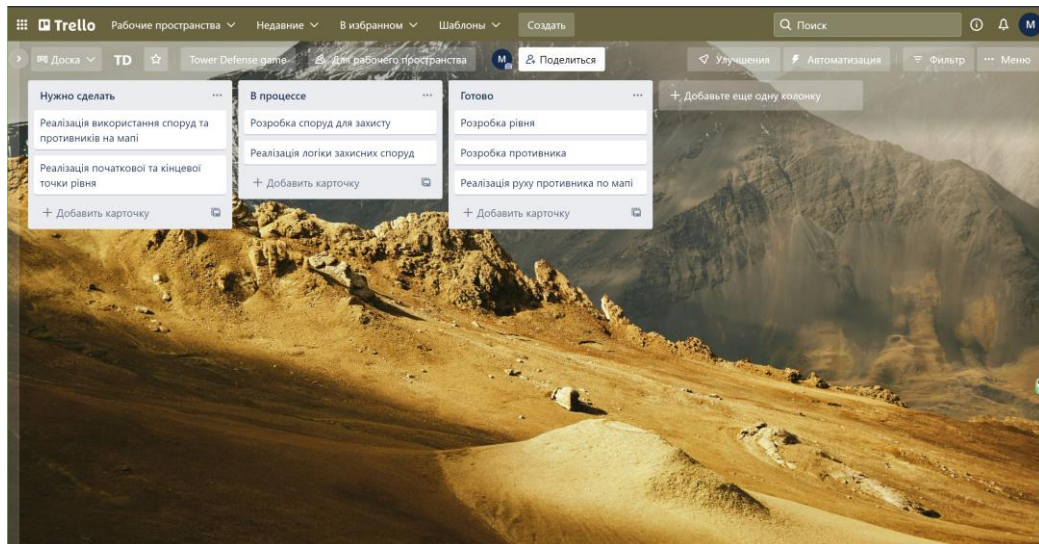


Рис. 3.2 – Відстеження статусу розробки продукту в Trello

Для зручності кожен задачу було описано в форматі, того, які задачі буде виконувати функція та, як вона буде взаємодіяти з гравцем (Рис. 3.3).

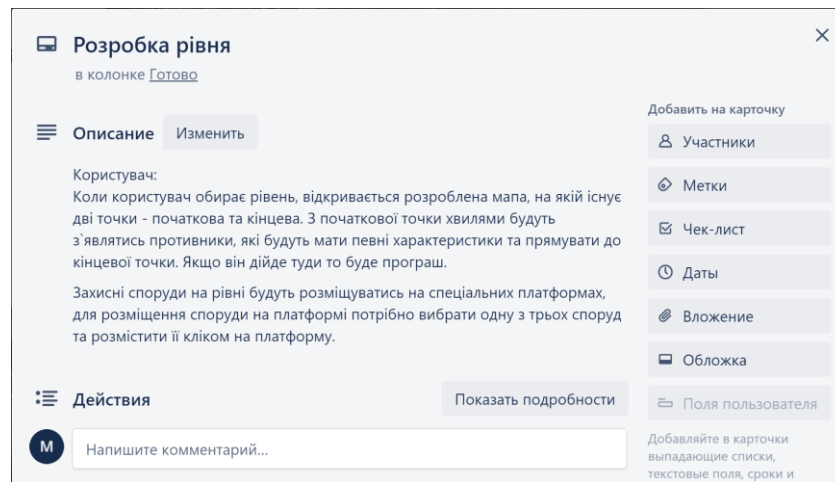


Рис. 3.3 – Приклад описання задачі

### 3.2 Набір використаних інструментів при розробці гри

Звісно при розробці ігор виникає питання, який ігровий двигун обрати, але опираючись на тему роботи, потрібно розробити гру за допомогою використання одного з найпопулярніших ігрових двигунів Unity та мови програмування C#.

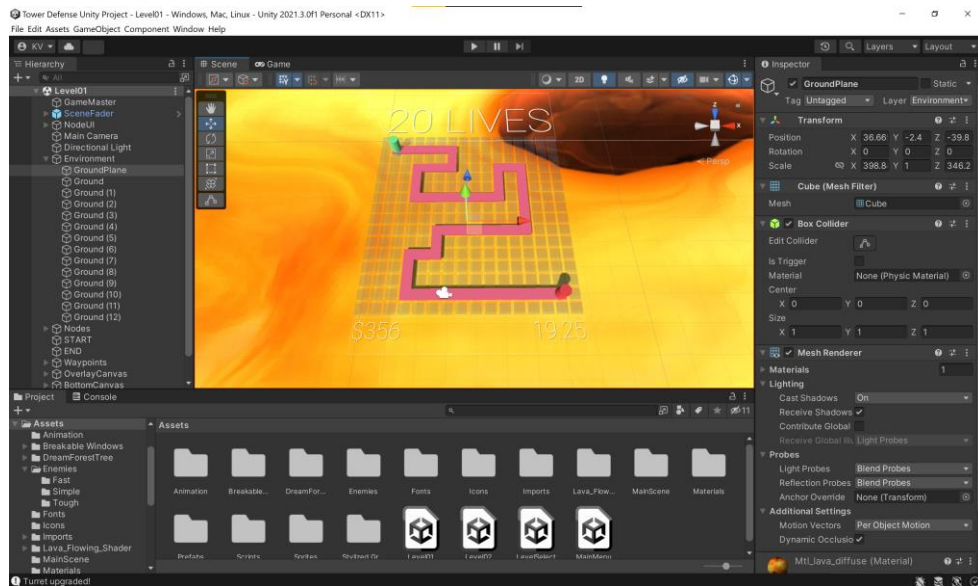


Рис. 3.4 – Unity

Оскільки Unity рекомендується для використання саме під мову C#, то звісно можна зрозуміти, що це є потужним середовищем для розробки та має переваги серед представлених аналогів, а саме:

- Зручний інтерфейс;
- Гарна сумісність з мовою програмування C#;
- Великий вибір асетів та широкий спектр сайтів, довідників та книг;
- Інтеграція з системою контролю версій GIT.

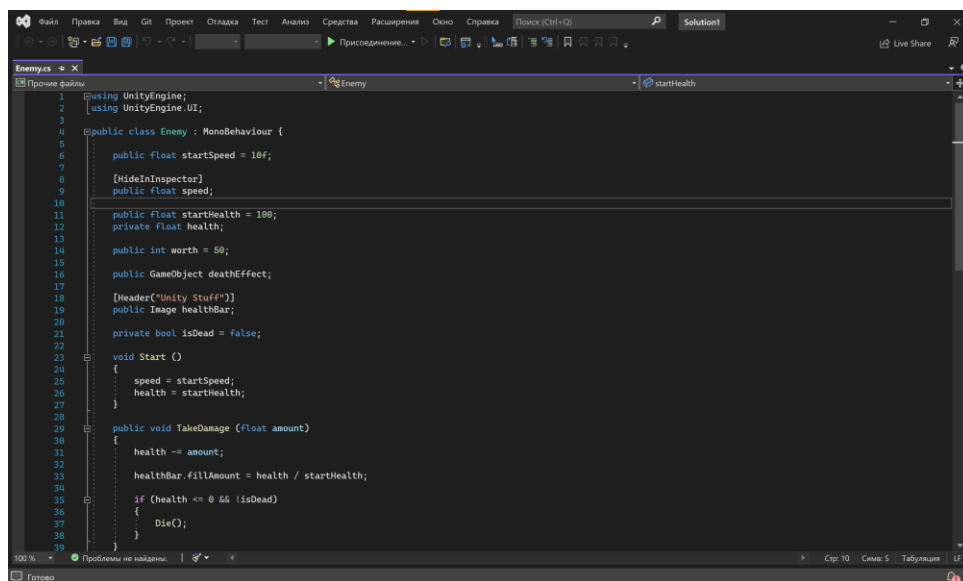


Рис. 3.5 – середовище розробки Microsoft Visual Studio

В якості редактору коду, було обрано Microsoft Visual Studio, оскільки вона добре інтегрована в Unity та має такі самі переваги, як і сам ігровий двигун Unity.

Для відстеження статусу виконання окремих задач, які потрібні для успішного виконання проекту, планування та документування успішності розробки було використано інструмент Trello, який є дуже простим та зручним у використанні та забезпечує комфортний менеджмент з розробки проекту (Рис. 3.6).

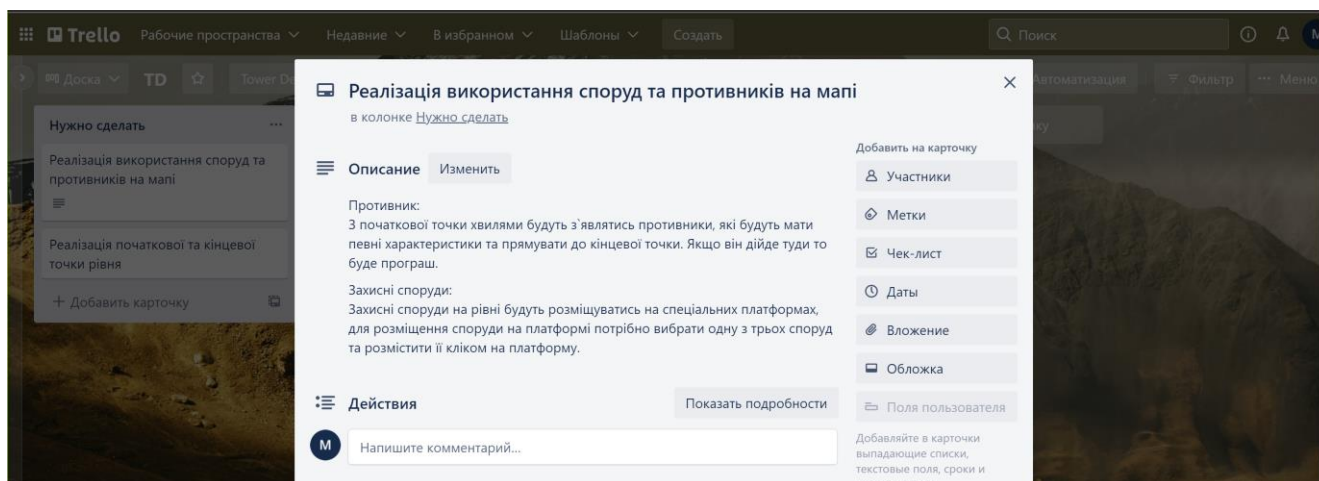


Рис. 3.6 – інструмент для менеджменту розробки проекту Trello

### 3.3 Функціонал гри

Описуючи функціонал гри, можна виділити декілька основних функцій:

- Поява противника на карті (Wave Spawner);
- Реалізація точок шляху (Waypoints);
- Перехід на наступний рівень (Level Selector);
- Закінчення рівня при виконанні певної умови (Complete Level);
- Реалізація статистики на рівні (Player Stats);
- Рахунок раундів хвиль противника (Rounds Survived).

Всі ці функції реалізовані за допомогою написаних скриптів, які мають власні поля та методи, що реалізують логіку функціонування.

Так, спавн нової хвилі противників після знищення попередніх буде відбуватись за допомогою скрипту Wave Spawner, в ньому описаний метод, який

відслідковує скільки залишилось противників на карті, якщо хвиля зачищена, то він запускає нову за допомогою інтерфейсу `SpawnWave()`, а в ньому ж описана логіка спавну нових противників (Рис. 3.7). Коли кількість хвиль стане рівною числу хвиль, яке задане для кожного рівня, то буде відкриватись нове меню, з вибором для переходу на наступний рівень, або до головного меню гри.

```
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. 3.7 – Реалізація спавну нових хвиль ворогів на мапі

Пошук шляху, реалізовано за допомогою так званих точок, до яких прямують противники, скрипт отримує нову точку до якої будуть прямувати противники та після досягнення її знаходити нову та прямувати до неї.

```
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. 3.8 – Реалізація пошуку нових точок для шляху противника

Так функція переходу на наступний рівень реалізується, коли кількість хвиль противника буде рівна кількості хвиль, яка задана для цього рівня та коли будуть знищені всі противники. Звісно за такою є логікою реалізовано перехід на наступний рівень.



Рис. 3.9 – Статистика, яка відображається в грі

Також для візуального відображення певної статистики, такої, як гроші, які має користувач для розміщення захисних споруд на мапі, кількість хвиль та часу, який є у гравця для підготовки перед новою хвилею противника реалізовано за допомогою певного скрипта, який має всі ці характеристики. Всі ці характеристики виведені на головний екран для кожного рівня, вони виконані, як інтерфейс, але стилістично вони відображаються наче прямий елемент в грі.



## 4. ПРИКЛАДИ ВИКОРИСТАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ

### 4.1 Опис інтерфейсу гри та ігрового процесу

Першим, що бачить кожен користувач будь-якої гри, звісно є меню та його інтерфейс. Тому потрібно було створити простий та приємний інтерфейс, який буде виконаний згідно одного стилю. Так після запуску гри, користувач бачить меню на якому є всього дві кнопки, це «Play» та «Quit», вони розміщуються на двох захисних спорудах, які є в грі, тому гравець може зразу побачити та оцінити, що його чекає в грі.

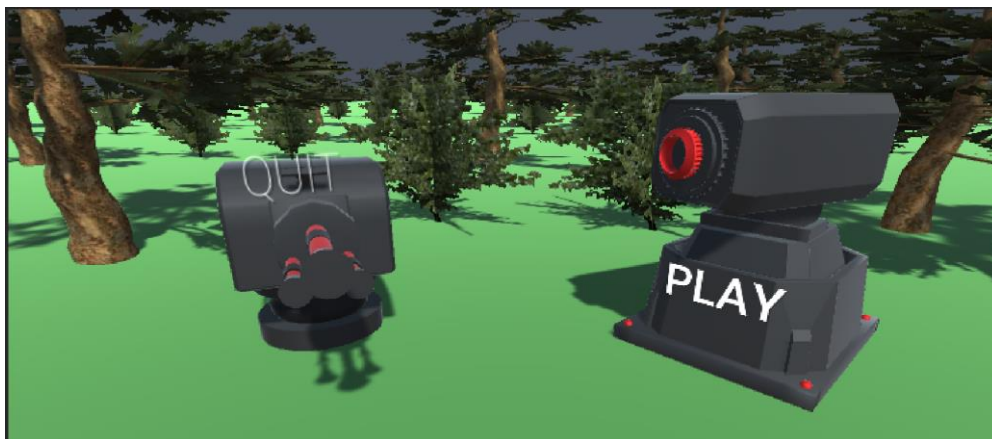


Рис. 4.1 – Стартове меню гри

Звісно після натискання кнопки «Quit», гра буде вимкнена. А кнопки «Play» відбудеться перехід до нового меню з вибором рівня для гри. В цьому меню буде відкрито доступ до певної кількості рівнів, які доступні гравцю (Рис. 4.2).

Надалі після вибору рівня, вже почнеться сама гра. Гравець буде бачити поле битви, яке має дві точки, це початкова точка виходу противника та кінцева точка, після досягання ворогом, гра буде провалена. Саме з першої точки будуть виходити різні вороги, які будуть відрізнятись між собою своїм зовнішнім виглядом та характеристиками.



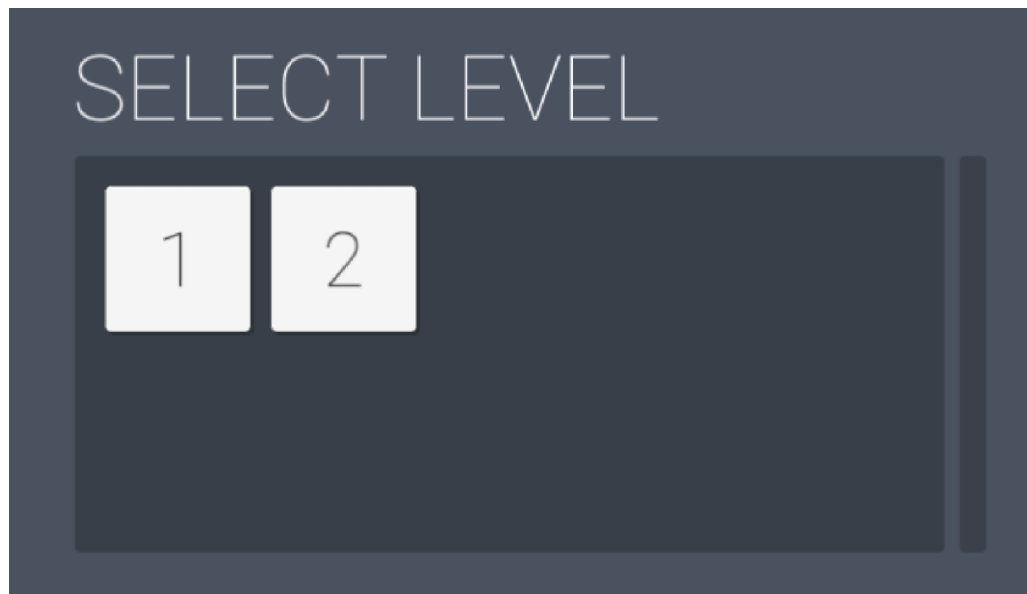


Рис. 4.2 – Меню вибору рівня

До інтерфейсу гри також можна віднести статистику, яка показується на ігровому полі. Це кількість допустимих ворогів, які можуть пройти до кінцевої точки і гра не буде провалена (для першого рівня, якщо хоча б один ворог дійде до кінця, гра провалена). Знизу під ігровим полем пишеться кількість коштів доступних для розміщення захисних споруд та час, який є до наступної хвили противника. Також на головному інтерфейсі знизу є захисні споруди, які можна розмістити на ігровій мапі.



Рис. 4.3 – Інтерфейс гри

Для того, щоб розмістити захисну споруду на ігровий простір потрібно натиснути на одну з запропонованих та натиском на панельку, поставити її для захисту. Також при повторному натиску на вже поставлену споруду її можна продати, або покращити (Рис. 4.4).

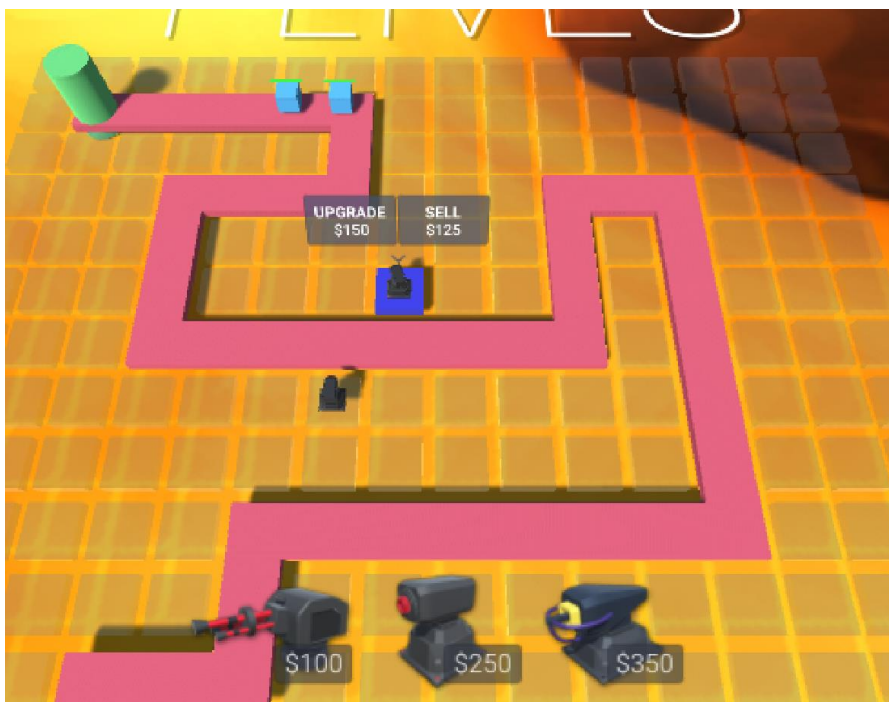


Рис. 4.4 – Можливості взаємодії з вибраною захисною спорудою

Для того, щоб гравець міг розуміти, яка захисна споруда вже покращена, вона має інший вид відносно попередньої. При покращенні вона стає більшою та кулі, якими споруди стріляють змінюють свій вигляд. Звісно при покращенні змінюються і характеристики цієї зброї, вона буде наносити більшу шкоду для певного типу противника (Рис. 4.5).

Після того, як рівень буде успішно завершений, відкриється меню з двома кнопками на вибір, «Continue» щоб продовжити гру і перейти на наступний рівень, «Menu» для того, щоб вийти до основного меню (Рис. 4.6).



Рис. 4.5 – Ігровий процес

Також, якщо рівень було провалено, відкриється відповідне меню з якого можна буде спробувати пройти рівень ще раз, або вийти до головного меню.

Переходячи на другий рівень гравця зустрічає вже звичний інтерфейс, який повинен бути оформлений в одному стилі, але для привабливості потрібно кожен, або кожні декілька рівнів змінювати фон та кольорову палітру під цей фон.

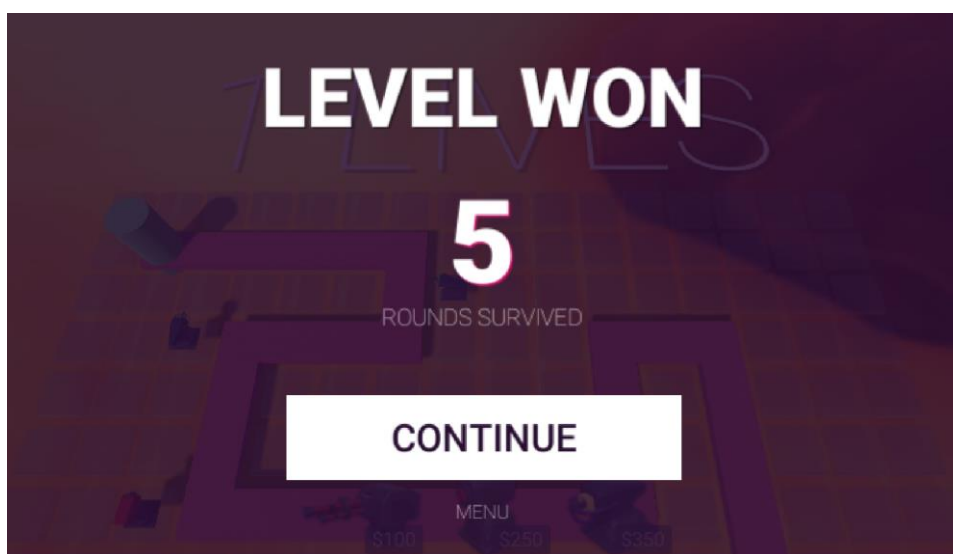


Рис. 4.6 – Меню після успішно закінченого рівня.

## 4.2 Результати апробації та подальший розвиток гри

Для підтвердження того, що продукт готовий до використання та для знаходження певних недоліків, які могли бути упущені при розробці, потрібно було провести апробацію проекту, або його тестування серед певного кола користувачів.

Так для тестування було проведено випробування серед трьох користувачів, які в свою чергу зазначили, що продукт має сильні сторони і звісно, ж деякі недоліки.

Основними перевагами були виявлені:

- Простота у використанні;
- Стабільна робота гри та відтворення потрібних функцій;
- Приємний інтерфейс та підібрані кольори.

Але були і недоліки, такі як:

- Невелика кількість рівнів;
- Варіативність розроблених противників і захисних споруд;
- Неможливість обрати складність ігрового процесу.

Отримавши цю інформацію, яка підтвердила, що продукт задовольняє мінімальні вимоги та показує гарну якість використання, був сформований подальший план розвитку гри та її покращень:

- Збільшення кількості ігрових рівнів, у нових стилях;
- Можливість обирати більшу кількість захисних споруд;
- Покращення в інтерфейсі гри;
- Розширення функціоналу;
- Додавання більшої кількості противників.

Цей план розроблений згідно того, щоб усунути вище зазначені недоліки, так і додавання покращень, які зроблять ігровий процес цікавішим, а використання гри більш приємним.

## ВИСНОВКИ

Підводячи підсумки роботи, можна констатувати, що потреба в розробці нових ігор звісно зростає. Гравці постійно потребують якихось новинок, від яких вони зможуть отримувати нові позитивні емоції та котрі будуть забезпечувати ігри новим функціоналом, сюжетом і так далі.

Дана робота було спрямована на розробку нової гри у жанрі Tower Defense за допомогою ігрового двигуна Unity та мови програмування C#, що допоможе новачкам ігрової індустрії ознайомитись з нею та з жанром Tower Defense.

1. Досліджено актуальність розробленого проекту та визначено, що для отримання першого позитивного досвіду ігровий світ потребує в розробці простих ігор, що дадуть таку можливість новачкам. Було підтверджено, що ігрова індустрія постійно потребує нових ігор.
2. Досліджено типові технології, які використовуються при розробці ігор в сучасному світі. Вибрано інструменти для побудови проекту, такі як Unity, C#, Trello, Unity Asset Store, їх використання та ефективність було обгрунтовано зрівнявши їх з популярними аналогами.
3. Також для забезпечення розуміння концепції майбутньої гри було проведено аналіз типових існуючих проектів, що надало змогу визначити основні недоліки існуючих проектів та визначити мету розробки. Після розбору аналогів було описано мінімальні вимоги, які допомогли виконати проект вчасно та з досягненням поставленої мети. За допомогою опису вимог, було створено функції та методи, які допомогли реалізувати логіку появи противника та роботи захисних споруд, які є основою гри.
4. Описано програмні засоби та інструменти, які були застосовані для розробки гри. Так стало зрозумілим, що ігровий двигун Unity забезпечить простоту розробки та гарну роботу додатку на різних ПК, за допомогою якої користувач зможе запустити гру навіть на дуже слабкій платформі. Підібрано мову програмування C#, адже використовуючи її можна забезпечити правильність розробки методів та функції, які повинні бути в

даному проєкті. Також для забезпечення менеджменту та відстеження статусу виконання проєкту, було використано Trello. В ньому просто можна описати вимоги, з детальними характеристиками та при виконанні певної вимоги, відмітити це.

5. Якість та відповідність роботи проєкту було апробовано за допомогою проведення практичних тестувань серед 3 користувачів, які мають навички в тестуванні ігор і було описано переваги та недоліки проєкту, які допомогли визначитись з майбутнім планом розробки та покращенням проєкту. Виявлено, що основними перевагами є:
- Простота у використанні;
  - Стабільна робота гри та відтворення потрібних функцій;
  - Приємний інтерфейс та підібрані кольори.

Результати досліджень бакалаврської роботи апробовані на всеукраїнських науково-технічних конференціях: Науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаційних технолоніях», м.Київ: ДУТ, 20.04.2022 року та Міжнародна студентська наукова конференція «Діджиталізація науки як виклик сьогодення», 03.06.2022р., м. Львів.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Аналіз динаміки розвитку ринку відеоігор, джерел його фінансування та особливостей монетизації продукції в даній сфері. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: [http://www.economy.nayka.com.ua/pdf/5\\_2021/4.pdf](http://www.economy.nayka.com.ua/pdf/5_2021/4.pdf) - Дата звернення: 20.04.2022
2. Історія розвитку комп'ютерних і відео ігор. Історія виникнення комп'ютерних ігор. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://fondeco.ru/uk/istoriya-razvitiya-kompyuternyh-i-video-igr-istoriya-vozniknoveniya/> - Дата звернення: 20.04.2022
3. Ігрова індустрія до 2020 року — мільярдні прибутки, епоха мобільних ігор та спортивний бізнес. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://nachasi.com/creative/2018/08/20/igrova-industriya-do-2020/> - Дата звернення: 20.04.2022
4. 50 років ігрової індустрії. Основні етапи розвитку бізнесу ігор. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://webdevandseo.com.ua/50-years-of-the-gaming-industry-stages-of-game-development/> - Дата звернення: 20.04.2022
5. Класифікація за жанрами. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://sites.google.com/site/wwwbossfighttvua/klasifikacia-igor/klasifikacia-za-zanrami> - Дата звернення: 21.04.2022
6. Ігри на ПК в стилі баштового захисту. Ігри Tower Defence (Захист Башти) онлайн. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://tokyo161.ru/uk/warming/igry-na-pk-v-stile-bashennaya-zashchita-igry-tower-defence-zashchita-bashni-onlain/> - Дата звернення: 22.04.2022



7. Вибираємо програму для створення гри. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://uk.soringpcrepair.com/programs-for-make-games/> - Дата звернення: 22.04.2022
8. Інструментальні засоби для розробки комп'ютерних ігор. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://conf.ztu.edu.ua/wp-content/uploads/2020/05/19.pdf> - Дата звернення: 23.04.2022
9. Використання двигуна Unity в мобільному геймдеві: можливості, переваги та недоліки. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://vokigames.com/ua/vikoristannya-rushiya-unity-u-mobilnomu-gejmdevi-mozhливosti-perevagi-ta-nedoliki/> - Дата звернення: 23.04.2022
10. Unity (рушій гри). [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: [https://uk.wikipedia.org/wiki/Unity\\_\(%D1%80%D1%83%D1%88%D1%96%D0%B9\\_%D0%B3%D1%80%D0%B8\)](https://uk.wikipedia.org/wiki/Unity_(%D1%80%D1%83%D1%88%D1%96%D0%B9_%D0%B3%D1%80%D0%B8)) - Дата звернення: 24.04.2022
11. C# Tutorial). [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://www.w3schools.com/cs/index.php> - Дата звернення: 24.04.2022
12. C# documentation). [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://docs.microsoft.com/en-us/dotnet/csharp/> - Дата звернення: 25.04.202

13. Діаграма варіантів використання (use case diagram). [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: [http://mmsa.kpi.ua/sites/default/files/disciplines/didkovska\\_m\\_v\\_testing\\_lecture\\_3.pdf](http://mmsa.kpi.ua/sites/default/files/disciplines/didkovska_m_v_testing_lecture_3.pdf) - Дата звернення: 26.04.2022
14. Microsoft Visual Studio). [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: [https://uk.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio) - Дата звернення: 26.04.2022
15. Чому варто вивчати мову програмування C#. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://www.quality-assurance-group.com/chomu-varto-vyvchaty-movu-programuvannya-c-c-sharp/> - Дата звернення: 28.04.2022
16. Тренди у програмуванні. Ігри. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://freelancehunt.com/blog/triendi-u-proghramuvanni-ighri/> - Дата звернення: 28.04.2022
17. Defense Zone 3 Ultra HD on Steam. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: [https://store.steampowered.com/app/324520/Defense\\_Zone\\_3\\_Ultra\\_HD](https://store.steampowered.com/app/324520/Defense_Zone_3_Ultra_HD) - Дата звернення: 23.04.2022
18. Plants vs. Zombies 2 Безкоштовна гра для мобільних пристроїв. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://www.ea.com/ru-ru/games/plants-va-zombies/plants-vs-zombies-2> - Дата звернення: 23.04.2022
19. Kingdom Rush – Tower Defense on Steam. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: [https://store.steampowered.com/app/246420/Kingdom\\_Rush\\_Tower\\_Defense](https://store.steampowered.com/app/246420/Kingdom_Rush_Tower_Defense) - Дата звернення: 23.04.2022
20. Swamp Attack Tower Defense. [Електронний ресурс]: [Веб-сайт] – електронні дані. – Режим доступу: <https://pdalife.to/swamp-attack--android-a10270.html> - Дата звернення: 23.04.2022

# ДОДАТОК А

## Демонстраційні матеріали



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



## РОЗРОБКА ГРИ У ЖАНРІ TOWER DEFENSE НА UNITY ЗА ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ C#

Виконавець: студент 4 курсу,  
групи ПД-41  
Коваленко Володимир Миколайович  
Керівник роботи: Трінтіна Наталія  
Альбертівна, доцент кафедри ІПЗ,  
кандидат технічних наук

Київ 2022

### Аналіз аналогів

Розглянуті ігри	Plants vs Zombies 2	Swamp Attack	Kingdom Rush	Defense Zone 3
Платформи	PC, iOS, Android, Xbox Live Arcade, PlayStation Network	iOS, Android, PC	iOS, Android, Nintendo Switch, PC	Android, Linux
Складність ігрового процесу	Середньої складності	Середньої складності	Дуже складна	Складна
Доступність	Безкоштовна	безкоштовна	безкоштовна	2.49\$
Візуальна складова	+	-	+	-
Переваги	Оригінальний погляд на жанр; Велика кількість захисних споруд та ворогів	Різноманітність озброєння та його покращення	В грі доступний окремий герой, що є не дуже типовим для цього жанру	Можливість покращити озброєння до 5 різних типів та декількох десятків рівнів
Недоліки	Для отримання всього контенту гри, потрібно платити реальні кошти	Гравець не може безперервно грати, йому дається енергія тільки на 5 битв, далі потрібно чекати поки вона відновиться	Складність гри через велику кількість незрозумілих новачку механік та кількість хвиль противника	4 види складності гри від просто складної до неможливої

## Мета, об'єкт та предмет роботи

- **Мета роботи** – підвищення зацікавленості гравця в ігровому процесі гри жанру Tower Defense за рахунок впровадження нових противників та покращення захисних споруд.
- **Об'єкт дослідження** – процес гри в жанрі Tower Defense.
- **Предмет дослідження** – програмний продукт в жанрі Tower Defense.

3

## Технічне завдання

- Реалізувати декілька рівнів для демонстрації основних функцій.
- Розробити противників та забезпечити їх характеристиками.
- Реалізувати логіку проходження шляху по мапі від початкової точки до кінцевої.
- Розробити засоби, які будуть зберігати результати виконаного завдання.
- Реалізувати показ статистики гравця.
- Реалізувати захисні споруди, з можливістю їх покращення та видалення.

4

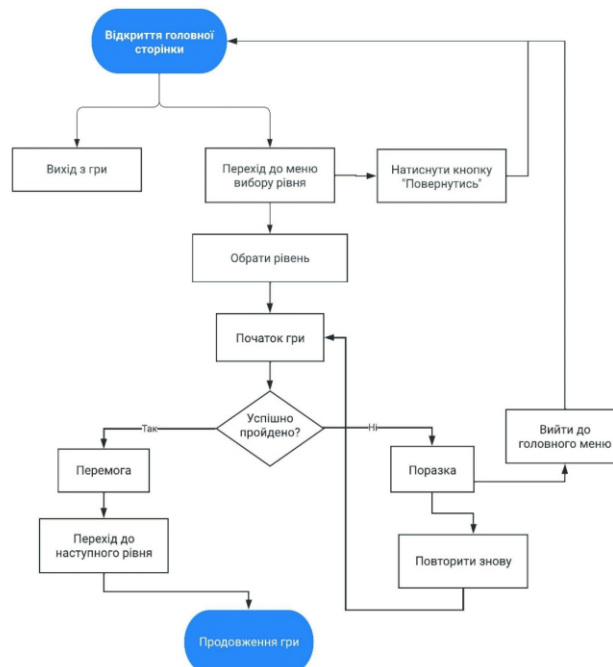
## Програмні засоби та інструменти реалізації

- **Unity** – був використаний, як ігровий двигун для розробки гри.
- **Visual Studio 2019** – використовувався для написання коду, потрібного для реалізації функцій, характеристик та логіки окремих елементів гри.
- **Unity Asset Store** – використано для отримання певних текстур, шейдерів, окремих елементів.
- **Trello** – використано для зручності опису окремих вимог та відслідковування процесу їх реалізації.



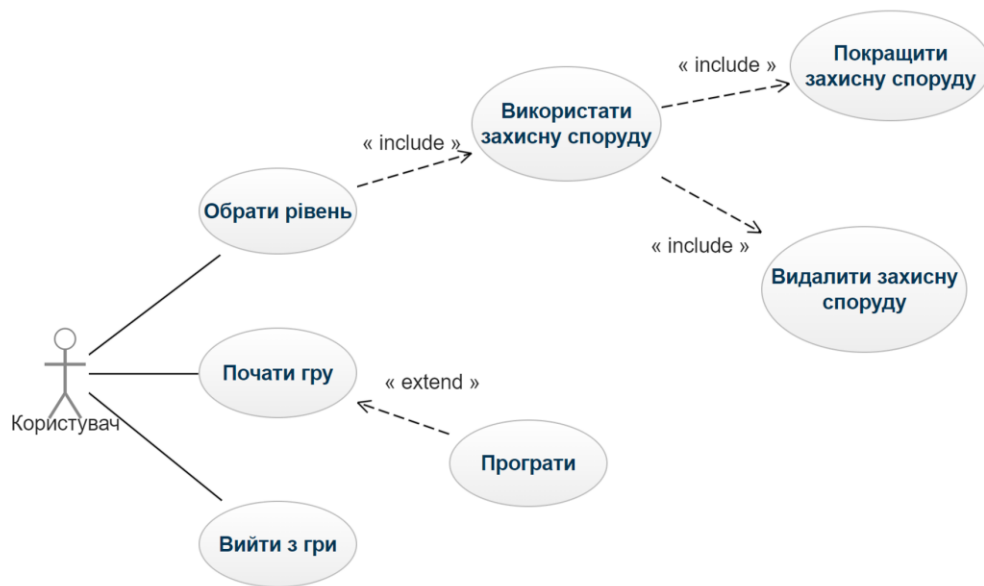
5

## Загальний алгоритм гри



6

## Діаграма використання користувача

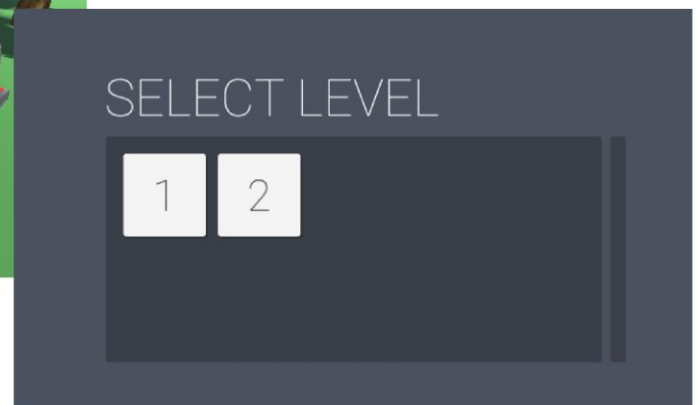


7

## Елементи інтерфейсу



Початкове меню



Меню вибору рівня

8

## Екранні форми



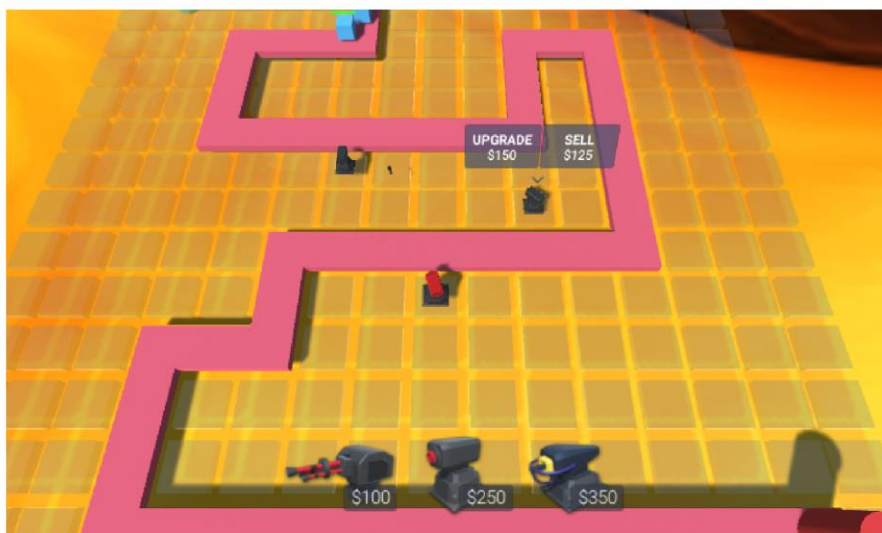
Ігрове мапа з противниками



Розміщення захисних споруд

9

## Взаємодія з захисною спорудою

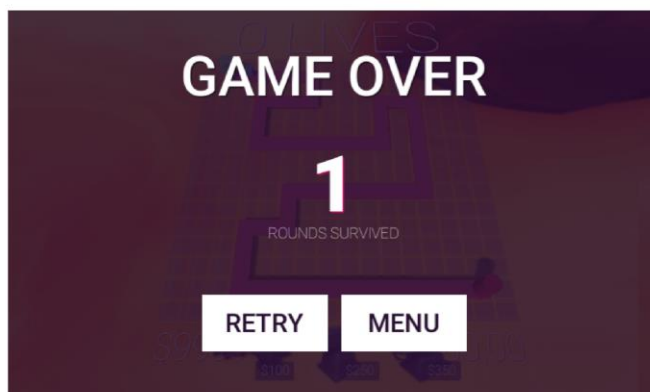


Кнопки для взаємодії зі спорудою

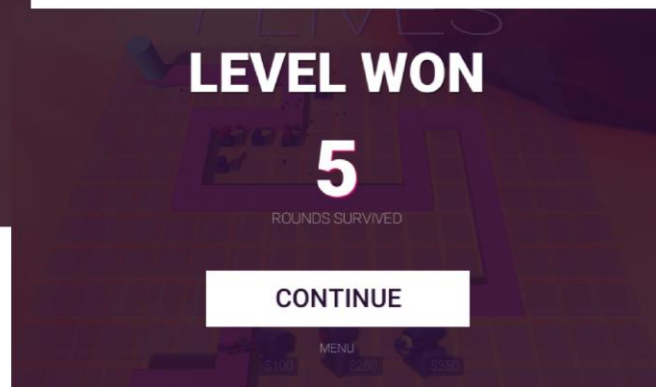
10



## Меню результатів рівня



Меню програшу



Меню перемоги

11

## Апробація результатів дослідження

- Коваленко В.М. UI/UX дизайн в розробці ігор: Науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаційних технологіях», ДУТ, 20.04.2022р., м. Київ, с. 27-28.
- Коваленко В.М. Розробка гри у жанрі Tower Defense на Unity за допомогою мови програмування С#: III Міжнародна студентська наукова конференція «Діджиталізація науки як виклик сьогодення», 03.06.2022р., м. Львів, с. 187-188

12

## Висновки

1. Проаналізовано та обгрунтовано 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 підвищують зацікавленість до ігрової індустрії та жанру Tower Defense.
2. Досліджено програмні засоби та інструменти, які можуть бути використані при розробці та найкраще впораються з створенням гри. Визначено, що взаємодія таких інструментів, як Unity та Visual Studio для написання коду на мові програмування C# надає змогу швидко та просто розробити гру та написати скрипти, що будуть інтегровані до гри.
3. Проаналізовано аналоги в жанрі Tower Defense, визначено недоліки та переваги. При аналізі аналогів виявлено, що сучасна індустрія потребує новинок, які спростять перше ознайомлення новачка з жанром Tower Defense.
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.

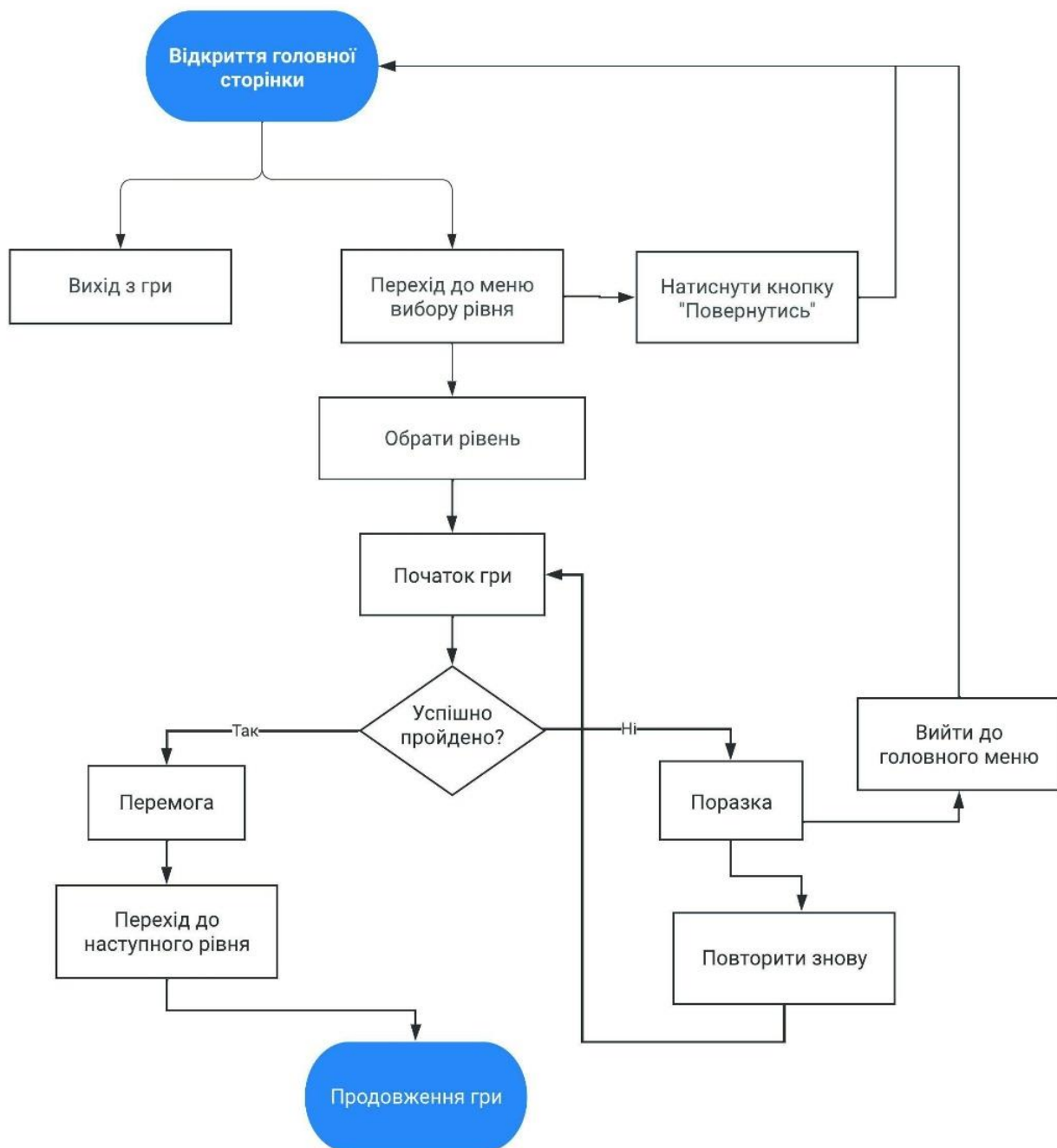
13

Дякую за увагу!

14

## ДОДАТОК Б

### Загальний алгоритм гри



## ДОДАТОК В

### Дизайн документ

### Гра в жанрі Tower Defense

**Платформи:** Гра розробляється на платформу ПК, з подальшим переносом на мобільні платформи iOS та Android.

**Цільова аудиторія:** Гра розрахована на будь-яку вікову категорію, але саме на новачків жанру на ігрової індустрії.

#### Ігрові механіки:

- Розміщення захисної споруди;
- Поява противника на мапі;
- Прохід противника по шляху;
- Знищення противника за допомогою захисних споруд.

**Сюжет та головні моменти гри:** Після початку ігрового рівня у гравця буде деякий час для розміщення захисних споруд, потім з початкової точки на мапі виходять противники, після чого, якщо гравець правильно та в достатній кількості розмістив захисні споруди то противники будуть знищені.

**Противники:** Три різних типу противника, які відрізняються за кольором та характеристиками.

- Противник, який має середню швидкість та кількість життів.
- Противник швидкий та з маленькою кількістю життів.
- Противник, який має дуже багато життів, але невелику швидкість.

#### Захисні споруди:

- Захисна споруда, яка знищує противника за допомогою лазера.
- Захисна споруда, яка знищує противника за допомогою снарядів.
- Захисна споруда, яка знищує противника за допомогою ракет.

**Контент:** Гра буде забезпечена контентом на 15 ігрових годин, середня ігрова сесія користувача очікується приблизно 30-50 хвилин. Роль користувача в грі, це розміщення захисних споруд купуючи їх за ігрову валюту та покращення захисною споруди. Для успішного захисту кінцевої точки з кожним рівнем від гравця буде потребуватись все більше розміщених споруд.

**Монетизація:** Гра буде повністю безкоштовна, але вона буде монетизована за рахунок реклами, яка буде з`являться у гравця після успішного закінчення трьох рівнів та після провалу хоча б одного рівня.