

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему: Системний аналіз засобів розробки для
мобільних платформ та розробка гри у жанрі Casual Action

Виконав: студент групи 4144

_____ *Малицький Р.П.*
(підпис)

Керівник роботи:

_____ *старший викладач*
(посада, науковий ступінь вчене звання)

_____ *Павленко А.Ю.*
(підпис)

м. Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра інформаційних управляючих систем та технологій

Спеціальність 124 "Системний аналіз"

Освітня програма "Системний аналіз"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц.НУК Приходько К.С.

" ____ " _____ 2022 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття ступеня вищої освіти "бакалавр"

Студенту Маліцькому Роману Петровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Системний аналіз засобів розробки для мобільних платформ та розробка гри у жанрі Casual Action

Керівник роботи Павленко Альона Юріївна

Затверджені наказом ректора № 269-уч від "18" травня 2022 року.

2. Термін подання роботи: 20 червня 2022 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем.

4. Перелік питань, що належать до розробки (найменування розділів) Розробка концепції інформаційної системи мобільної гри «Astro Survival». Розробка проекту інформаційної системи. Реалізація проекту автоматизованої системи. Охорона праці.

5. Перелік презентаційних матеріалів _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Павленко А.Ю.		
2	Павленко А.Ю.		
3	Павленко А.Ю.		
4	Гурець Н.В.		

7. Дата видачі завдання 14 лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Вивчення і опис предметної області	10.03.2022	
2	Розробка концепції системи	21.03.2022	
3	Розділ з розробки проектних рішень по системі (та її частинах)	29.03.2022	
4	Реалізація проекту та розробка робочої документації	10.04.2022	
5	Вирішення питань охорони праці	18.04.2022	
6	Виконання графічних документів	28.04.2022	
7	Оформлення пояснювальної записки	05.05.2022	
8	Подання ДР на перевірку на плагіат	15.05.2022	
9	Подання ДР рецензенту	20.06.2022	
10	Подання ДР на державний захист	26.06.2022	

Студент

(підпис)

Маліцький Р.П.

(ПІБ)

Керівник роботи

(підпис)

Павленко А.Ю.

(ПІБ)

АНОТАЦІЯ

Тема роботи – «Системний аналіз засобів розробки для мобільних платформ та розробка гри у жанрі Casual Action».

Дипломний проект виконується для покращення навичок аналізу, проектування та розробки інформаційних систем. Розробляється інформаційна система мобільної гри «Astro Survival».

Метою роботи є системний аналіз засобів розробки для мобільних платформ та проходження повного шляху у створенні додатку мобільної гри «Astro Survival» у жанрі Casual, Action.

Актуальність роботи полягає в тому, що у наш час казуальні мобільні ігри стають все більш популярними, а деякі дослідження виявляють що мобільні ігри приносять прибуток набагато більший ніж ігри для персонального комп'ютеру або ігрових консолей. На цю популярність впливає те, що смартфон є у кожної людини в кишені і багато хто проводить свій вільний час використовуючи застосунки у смартфоні. Також більшість мобільних ігор мають дуже велике охоплення аудиторії як різного полу так і різного віку.

В роботі було проведено аналіз предметної області, дослідження існуючих рішень та виявлені існуючі недоліки, аналіз засобів розробки для мобільних платформ. Всі вимоги та обмеження обраної інформаційної системи було затверджено в технічному завданні. Розроблені проектні рішення по системі з різних видів забезпечення, передбачених етапами ескізного та технічного проектування, зокрема загальносистемні рішення, рішення з організаційного забезпечення, рішення з програмного забезпечення. Розроблена робоча документація по спроектованій системі та виконаний спеціальний розділ з охорони праці.

Робота викладена на 71 сторінках друкованого тексту, містить 4 розділа, 20 рисунків, 4 таблиць та 4 додатки, які містять словник, технічне завдання, текст програми та інструкцію з використання.

Робота виконана українською мовою.

DESCRIPTION

The theme of the work is "System analysis of development tools for mobile platforms and game development in the genre of Casual Action".

The thesis project is performed to improve the skills of analysis, design and development of information systems. The information system of the mobile game "Astro Survival" is being developed.

The purpose of the work is a systematic analysis of development tools for mobile platforms and the full path in creating a mobile application "Astro Survival" in the genre of Casual, Action.

The relevance of the work is that nowadays casual mobile games are becoming more popular, and some studies show that mobile games are much more profitable than games for personal computers or game consoles. This popularity is influenced by the fact that everyone has a smartphone in their pocket and many people spend their free time using applications in the smartphone. Also, most mobile games have a very large audience of both gender and age.

The paper analyzes the subject area, studies of existing solutions and identifies existing shortcomings, analysis of development tools for mobile platforms. All requirements and limitations of the selected information system were approved in the terms of reference. Developed design solutions for the system of different types of software, provided by the stages of sketch and technical design, in particular system-wide solutions, solutions for organizational support, software solutions. Working documentation on the designed system is developed and the special section on labor protection is executed.

The work is presented on 71 pages of printed text, contains 4 sections, 20 figures, 4 tables and 4 appendices, which contain a dictionary, terms of reference, program text and instructions for use.

The work is done in Ukrainian.

ВСТУП

РОЗДІЛ I РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

МОБІЛЬНОЇ ГРИ «ASTRO SURVIVAL».....

1.1 Опис об'єкту та аналіз предметної області.....

1.1.1 Аналіз предметної області.....

1.1.2 Аналіз існуючих рішень

1.1.3 Аналіз засобів розробки для мобільних платформ.....

1.2 Вимоги до інформаційної системи.....

1.3 Розробка концепції інформаційної системи.....

РОЗДІЛ II РОЗРОБКА ПРОЕКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

«ASTRO SURVIVAL».....

2.1 Загальносистемні рішення.....

2.1.1 Діаграма варіантів використання.....

2.1.2 Специфікація варіантів використання.....

2.2 Рішення з інформаційного забезпечення.....

2.2.1 Діаграма класів.....

2.2.2 Діаграма переходів станів.....

2.2.3 Діаграма діяльності

2.2.4 Діаграма компонентів.....

2.2.5 Діаграма розгортання.....

2.3 Рішення з програмного забезпечення.....

РОЗДІЛ III РЕАЛІЗАЦІЯ ПРОЕКТУ АВТОМАТИЗОВАНОЇ

СИСТЕМИ.....

РОЗДІЛ IV ОХОРОНА ПРАЦІ.....

3.1 Вступ.....

3.2 Аналіз шкідливих та небезпечних факторів на робочому місці
програміста.....

3.3 Розробка заходів щодо зменшення впливу виробничих факторів на
працюючого.....

ВИСНОВКИ.....

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....

ДОДАТОК А – Словник.....

ДОДАТОК Б – Технічне завдання.....

ДОДАТОК В – Текст програми.....

ДОДАТОК Г – Інструкція з використання.....

ВСТУП

На популярність мобільних ігор впливає те, що смартфон є у кожної людини в кишені і багато хто проводить свій вільний час використовуючи застосунки у смартфоні. Також більшість мобільних ігор направленні мають дуже велике охоплення аудиторії як різного полу так і різного віку.

У ігровій індустрії існують багато різних жанрів. Але на теперішній час жанри ігор виходять за рамки одного жанру комбінуючи у собі особистості різних жанрів. Сам жанр «Casual» полягає у простоті та спонуканні гравця проводити якомога більше часу у грі аби дістатися її кінця. Для більшого інтересу з боку гравця цей жанр комбінують з іншими такими як Action, RPG, Simulator тощо. Тому для гри було вибрано поєднання жанрів Casua; та Action. Жанр Casual відповідає за простоту та настрої поданий у грі а жанр Action відповідає за геймплей.

Саме тому ця комбінація ідеально підходить для цього проекту оскільки після проведеного аналізу мобільної ігрової індустрії було виявлено, що Casual Action на мобільних пристроях користуються великою популярністю.

Аби отримати очікуваний результат використовувався сучасний ігровий двигун Unity та об'єктно орієнтована мова програмування C#.

РОЗДІЛ I РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОБІЛЬНОЇ ГРИ «ASTRO SURVIVAL»

1.1 Опис об'єкту та аналіз предметної області

1.1.1 Аналіз предметної області

Кількість ігор щодня зростає, розробники намагаються здивувати аудиторію різною комбінацією жанрів або ігрових механік аби зацікави користувача провести у грі якомога більше часу та, якщо можливо, змусити його витратити реальні гроші та купити ігрову валюту, доповнення до гри та інший ігровий контент який розробник ввів у гру.

Таким чином користувачі проводять за іграми дуже багато часу, а для когось це спосіб заробити гроші та/або здобути популярність на різних стрімінгових сервісах або у таблиці лідерів. Але усе це проходить не без наслідків – погіршення зору, залежність від комп'ютера, а з ерою розвитку смартфонів ще й залежність від телефону, однак, ігри розвивають такі навички як:

- уважність;
- пам'ять;
- усидливість;
- швидкість реакції;
- спрагу до знань;
- робота у команді;
- та багато інших.

Тому підсумовуючи усі ці фактори можна зробити висновок: від людини залежить – принесе гра задоволення, заробіток, знання, кохання та корисні навички або біль, витрати, погіршення відносин з рідними та/або друзями, притуплення базових навичок та інше.

Гра – це інструмент у руках людини.

А якою буде гра залежить від розробника та його фантазії, так жанрів ігор є багато:

- RPG - гра розважального характеру, в якій учасники приймають певні конкретні ролі та колективно створюють історію або слідуєть уже у вигаданих ситуаціях діючи відповідно до своїх ролей,
- ACTION - жанр відеоігор, який вимагає від гравця певних фізичних зусиль, таких як швидкості реакції та здібності швидко приймати тактичні рішення. Дія таких ігор відбувається дуже динамічно і потребує більшою мірою рефлексів, ніж логіки і планування,
- ROGULIKE - це піджанр рольових відеоігор, визначними особливостями якого є випадкове, процедурне створення рівнів, покроковий ігровий процес, плиткова (тайлова) або ASCII-графіка та перманентна смерть персонажа у випадку поразки,
- SIMULATOR - це жанр відеоігор, характерною особливістю якого є якомога точніше відтворення фізичних законів реального світу, властивостей реальних предметів, процесів та подій,
- SHOOTER - жанр відеоігор, піджанр action, основу ігрового процесу якого складає стрільба зі зброї по цілях, зазвичай вороже налаштованих. Часто основним завданням гравця є знищення усіх цілей на рівні, або/та набір найбільшої кількості очок, а не проходження сюжету,
- PLATFORMER - жанр відеоігор, ігровий процес в якому складається зі стрибків персонажа по різноманітних платформах (звідси і назва) та через перешкоди, збирання предметів, звичайно необхідних для завершення рівня,
- TOP DOWN – це категорія ігор де камера розташована зверху, та дивиться на ігрове поле,
- 2D – це категорія ігор де графіка двовимірна,
- 3D – це категорія ігор де графіка тривимірна,

Ми зупинимося на комбінації декількох жанрів, а саме: Top Down 2D Rogulike Shooter.

1.1.2 Аналіз існуючих рішень

Play Market, App Store та інші – це все магазини цифрових товарів де можна завантажити ігри, фільми, музику, книги та інше. Контент на них є безкоштовним та платним.

Зупинимося на іграх пошук яких є не складною задачею тому що усі грають у мобільні ігри, а якщо хтось говорить, що не грає – це не правда. Нас цікавлять жанри 2D, rogulike, shooter, top down де можна виділити наступні ігри:

- Pixel Dungeon[6] – 2D гра з видом зверху де гравець повинен якомога нижче спуститися у підземелля, на його шляху будуть зустрічатися загадки, сильні вороги та артефакти. Розробник відмовився від прокачки здібностей гравця і замість цього він ввів артефакти які надають персонажу здібностей. Але усе обмежено маленькою ігровою зоною, та малою кількістю ворогів, усе повільно та розмірено.

Скріншот ігрового процесу Pixel Dungeon представлено на рисунку 1.1

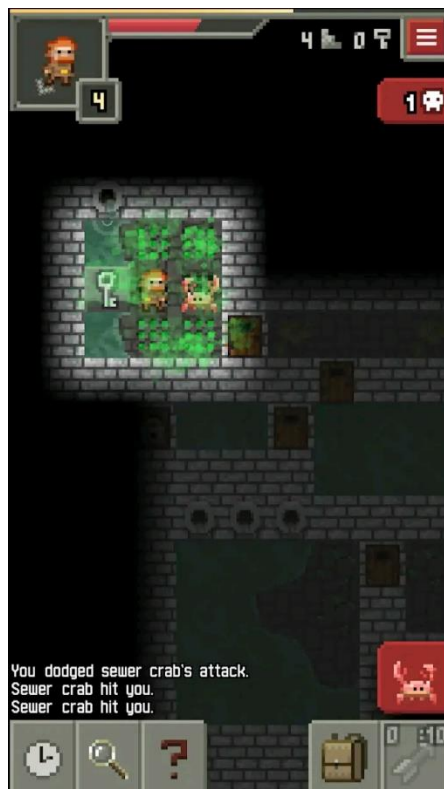


Рис. 1.1 – Скріншот ігрового процесу Pixel Dungeon

- Caves: rogulike[2] – дуже схожа на Pixel Dungeon але має прокачку рівня гравця та вибір здібностей. Також вид зверху, та 2D. Але вже динамічно, інколи немає часу відпочити.

Скріншот ігрового процесу Caves: rogulike представлено на рисунку 1.2



Рис. 1.2 – Скріншот ігрового процесу Caves: rogulike

- Apple knight: action platformer[1] – цю гру виділяє її графіка, розробник багато часу провів над ігровим світом щоб той справив на гравця незабутнє враження. Але стосовно жанра – це гра 2d, rogulike, platformer, зброї немає, здібностей немає. Ігрова зона поділена на рівні які гравцю потрібно проходити один за одним доки не досягне кінця гри.

Скріншот ігрового процесу Apple knight: action platformer представлено на рисунку 1.3



Рис. 1.3 - Скріншот ігрового процесу Apple knight: action platformer

- Space horizon[7] – це шутер з видом зверху де гравцю потрібно досліджувати навколишнє середовище, збирати різні речі щоб зробити

зброю могутніше та свій костюм міцніше та надійніше. Здібності гравця залежать від його зброї, а особливих навичок немає.

Скріншот ігрового процесу Space horizon представлено на рисунку 1.4



Рис. 1.4 – Скріншот ігрового процесу Space horizon

1.1.3 Аналіз засобів розробки для мобільних платформ

Оскільки мобільна розробка з кожним роком все популярніша то й засобів створення мобільних застосунків багато, наприклад:

- Unity[9] – це мультиплатформна програма для розробки ігор, у тому числі й мобільних, використовуючи мову програмування C#. Користується високою популярністю завдяки своїм перевагам:

- * умовно безкоштовна;
- * легка в освоєнні;
- * з низьким рівнем входження;
- * зручний інтерфейс;
- * має велику базу бібліотек та доповнень створених її користувачами.

Але є й недоліки:

- * займає багато місця на комп'ютері;
- * повільність.

- Godot[8] – це відкритий мультиплатформний 2D та 3D ігровий двигун. Загальна архітектура двигуна побудована навколо концепції дерева зі спадкових «сцен». Кожен елемент сцени, будь-якої миті сам може стати повноцінною сценою. Тому при розробці можна легко змінювати всю

архітектуру проекту, розширювати її елементи в будь-який бік і працювати з комплексними сценами на рівні простих абстракцій.

Використовує мову програмування GDScript.

Переваги:

- * має відкритий код;
- * зручна система додатків;
- * мультиплатформність.

Недоліки:

- * високий поріг входження;
- * мала база застосунків від користувачів;
- * не може оброблять 2D та 3D одночасною

- Unreal Engine[12] - це ігровий двигун який має велику кількість вже вмонтованих додатків та ще більшу базу бібліотек та доповнень від користувачів. Використовується переважно для 3D проектів, але, за необхідності можна створити й 2D. Використовує мову програмування C/C++.

Переваги:

- * зручні інструменти;
- * висока оптимізація;
- * найбільша база бібліотек та доповнень від користувачів;
- * великий інструментарій для створення будь яких 3D проектів;
- * користується популярністю серед великих компаній по розробці ігор.

Недоліки:

- * вибагливість до потужностей комп'ютера;
- * над високий рівень входження;
- * складний інтерфейс;
- * займає багато місця на комп'ютері.

- Game Maker Studio[5] – це платний, але легкий у освоєнні двигун для створення 2D проектів, користується популярністю у indie¹ розробників,

має вмонтований графічний редактор та зручний анімаційний редактор, здебільше використовує графічне програмування завдяки так званим нодам².

Переваги:

- * зручний інтерфейс;
- * велика кількість вмонтованих додатків;
- * низький рівень входження.

Недоліки:

- * вартість.

1.2 Вимоги до інформаційної системи

Вибір ігор великий, але інколи недостає тієї самої де є все і усього багато. Аби задовільнити перш за все свої потреби була розпочата розробка цього проекту де було об'єднано жанри ігор які до цього не були об'єднані, але світ мобільних ігор практично безмежний тому, можливо, подібна гра є, але вона не була знайдена.

Саме тому було прийняте рішення створення цього проекту який зібрав у собі жанри які складно об'єднати та від балансувати.

Такий проект передбачає велику кількість ворогів, велике ігрове поле, але, все ж таки обмежене. Велику кількість зброї та здібностей, можливість робити ігрового персонажа сильніше, змагатися з друзями у таблиці лідерів за перше місце та, обов'язково – у кожній грі мати можливість змінити свою тактику, та взяти інші здібності й іншу зброю аби скомбінувати їх і подивитися на результат. Знайти найкращу комбінацію навичок. Але якою б не була комбінація усе залежить від гравця – його реакції та можливості дивитися на декілька кроків вперед.

Тому пересування гравця повинне бути легким, плавним та обов'язково зручним – джостік підпадає під усі ці критерії його й вибираємо[3].

Вороги – повинні бути збалансованими перш за все тому що нікому не захочеться грати у гру де вбивають його персонажа самі перші «найслабші»

вороги. Беручи за увагу ці параметри – перші вороги повинні бути: слабкими та трішки повільнішими за гравця аби дати гравцю фору на деякий час.

З ворогів повинен бути шанс вибиття наступних, обов'язкових предметів: experience crystal, medkit так coin. Без цих предметів гра не буде тією самою яка була задумана розробником. Experience crystal - збільшує відсоток підймання рівня на, наприклад десять відсотків, тому, аби підняти рівень гравця потрібно підняти десять таких предметів. Medkit – повністю лікує гравця. Coin – дає гравцю, наприклад, п'ять монет за які гравець може збільшити свої базові характеристики в меню Upgrades.

Система випадіння предметів – це система в якій після смерті ворога буде відбуватися математична операція по обчисленню відсотка випадіння предмету, для кожного предмету повинен бути програмно заданий відсоток, але у разі необхідності цей відсоток можна легко змінити. Наприклад – гравець знищив базового ворога – після його знищення запускається функція яка обчислює відсоток випадіння предмету, якщо цей відсоток рівний відсотку одного з предметів – то на місці смерті після себе ворог залишає той предмет, відсоток якого дорівнює відсотку який повернувся після обчислення у функції.

Здоров'я гравця – повинне бути відображене наприклад зверху-зліва аби гравці могли слідкувати за своїм здоров'ям та за необхідності тікати від ворогів, або шукати medkit який випав з ворогів[3].

Система ворогів – це система як передбачає появу ворогів навколо гравця. Після появи вороги починають свій рух по заданому шаблону – для більшості ворогів цей шаблон – «пересуватися у напрямку гравця аби знищити його» [3].

Якщо ворог з'явився на ігровому полі і не може наздогнати гравця і з кожною секундою віддаляється від нього – то такого ворога потрібно переміщувати у pool³ усіх ворогів, тим самим він робиться неактивним і не займає собою ігрове середовище, та, що важливіше – ресурси пристрою. Потім, коли системі потрібно буде проявити ворога – вона не буде робити нового ворога – вона візьме одного з тих, хто знаходиться у pool і об'єктів[3].

Аби додати гравцю складності – потрібно додати зграї ворогів які будуть з'являтися по шаблону у формі заповненої геометричної фігури, наприклад коло.

Після появи усій зграї задається напрямок у якому вони швидко пересуваються. Цей напрямок дорівнює різниці між точкою появи ворогів та точці у якій знаходиться ворог. Тим самим ця зграя додає ефект несподіванки для гравця змушуючи гравця активувати одну зі своїх здібностей, або швидко тікати йдучи, іноді, на дуже ризиковані вчинки. Зграя, також як і усі інші вороги після віддалення від гравця переміщується у pool об'єктів звідки вона буде взята у наступному разі, але, якщо у зграї не вистачає ворогів – тоді, відсутні вороги будуть створені по новому.

Настав час системи підняття рівня гравцем. У цій системі повинно бути реалізована генерація трьох випадкових здібностей, можливість пере-генерувати здібності. Коли здібності згенеровано – їх буде відображено у спеціальному меню яке з'явиться одразу як гравець підніме свій рівень. Для гравця буде виведено картинка цієї здібності, короткий опис роботи здібності(наприклад пасивна здібність Critical hit – збільшує шанс нанесення подвійного пошкодження ворогу), та позначка, яка буде візуально говорити гравцю чи вибирає він цю здібність у перший раз, чи ні.

Таким чином коли гравець піднімає свій рівень гра зупиняється, з'являється відповідне вікно у якому гравцю потрібно вибрати одну з трьох здібностей.

Усі здібності які гравець вибрав під час гри буде відображено зверху-зліва біля здоров'я персонажа. Кількість здібностей повинна бути обмежена оскільки тоді, гравцям потрібно буде думати над комбінацією здібностей аби протриматися у грі якомога довше.

Якщо гравець вибрав здібність, та підтвердив свій вибір і здібність виявилася типа АКТИВНА – то повинне бути відображене вікно з описом роботи здібності. Наприклад гравець вибрав здібність Dash – ця здібність активного типа, аби її активувати потрібно натиснути кнопку яка з'явиться автоматично знизу-зліва. По її натисканню гравець приблизно за одну десяту секунди пересувається у перед на коротку відстань тим самим уникаючи зустрічі з ворогом. Якщо у напрямку в якому гравець активує Dash знаходяться вороги – вони ігноруються, окрім тих ворогів які знаходяться у точці в якій гравець зупиниться. Після

активації здібності вона піде на перезарядку. Таким чином здібність відбалансована – вона не дозволить постійно використовувати себе аби гравець тікав від ворогів постійно і інколи атакував заробляючи очки. Для кожної активної здібності передбачливо зроблено обмеження аби уникнути overpowered skill⁴.

Аби у гравця був азарт заходити у гру знову і знову потрібно додати складності, а саме боса. Це дуже сильні вороги які мають багато здоров'я та броні, але вони дуже повільні. Зроблено це для того аби збалансувати їх і не зробити занадто сильними. Бос має особливі здібності а також на відміну від інших ворогів він може стріляти по гравцю, та створювати собі приспівників. Авжеж у кожного боса будуть свої навички та здібності, хоча іноді вони й будуть повторюватись. Незважаючи на це - їх вигляд, та спосіб пересування завжди буде унікальним. Наприклад, бос - літаюча тарілка, вона може «літати» - тобто пересувати поверх інших ворогів аби наблизитися до гравця. Також вона буде засмоктувати в себе ворогів та завдяки особливій зброї буде «кидати» їх у гравця аби знищити його таким чином. Також вона здатна вражати гравця здалеку завдяки лазеру, та випускати маленьких автоматичних дронів які будуть літати навколо неї захищаючи летючу тарілку своїм тілом.

Система досягнень – проста система яка зчитує данні гри, наприклад кількість знищених ворогів або час який гравець провів у одному заході. Це все в кінці гри буде виведено завдяки рор-уп⁵ повідомленням які проінформують гравця завдяки чому він отримав досягнення.

Дизайн гри – дизайн повинен бути простим, не навантажений аби користувачу було приємно дивитися на нього. Потрібні анімації, яка ж гра без анімацій. Квадрат який обертається навколо своєї осі завжди привертає уваги більше ніж статичний квадрат. Тому анімації - наше все.

Particles⁶ – система частинок. Повинна бути, оскільки зможе додати гри динаміки без додаткових анімацій завдяки VFX⁷ ефектам.

Музика та звуки – музика повинна бути динамічна, швидка, аби при простому переміщенні персонажа гравець отримував адреналін та бажання знищити усе, що бачить.

Звіт про гру представлений у двох екземплярах – один у вікні лідерів, а другий більш докладніший у вікні кінця гри – де буде розписано детальна інформація про кінцеві характеристики гравця, його досягнення у кількості знищених ворогів, піднятих монет, заробленого рівня та кількість часу який гравець протримався живим. Уся ця інформація буде зберігатися на пристрою гравця аби можна було побачити у локальній таблиці лідерів з якою комбінацією здібностей гравець добився цього рекорду.

На основі проведеного аналізу сформовано наступні вимоги за функціоналом.

- формування таблиці лідерів;
- перегляд таблиці лідерів;
- зручний та приємний інтерфейс;
- інтуїтивне керування гравцем;
- підвищення рівня гравця;
- набуття гравцем здібностей.

Для виконання функціональних вимог необхідні наступні програмні засоби:

- операційна система;
- мова програмування для розробки;
- система управління базою даних.

Технічне завдання на розробку гри представлено у додатку Б.

1.3 Розробка концепції

За результатом сформованих вимог розроблено концепції інформаційної системи та її модель відповідно до функціональних вимог (рисунок 1.5)

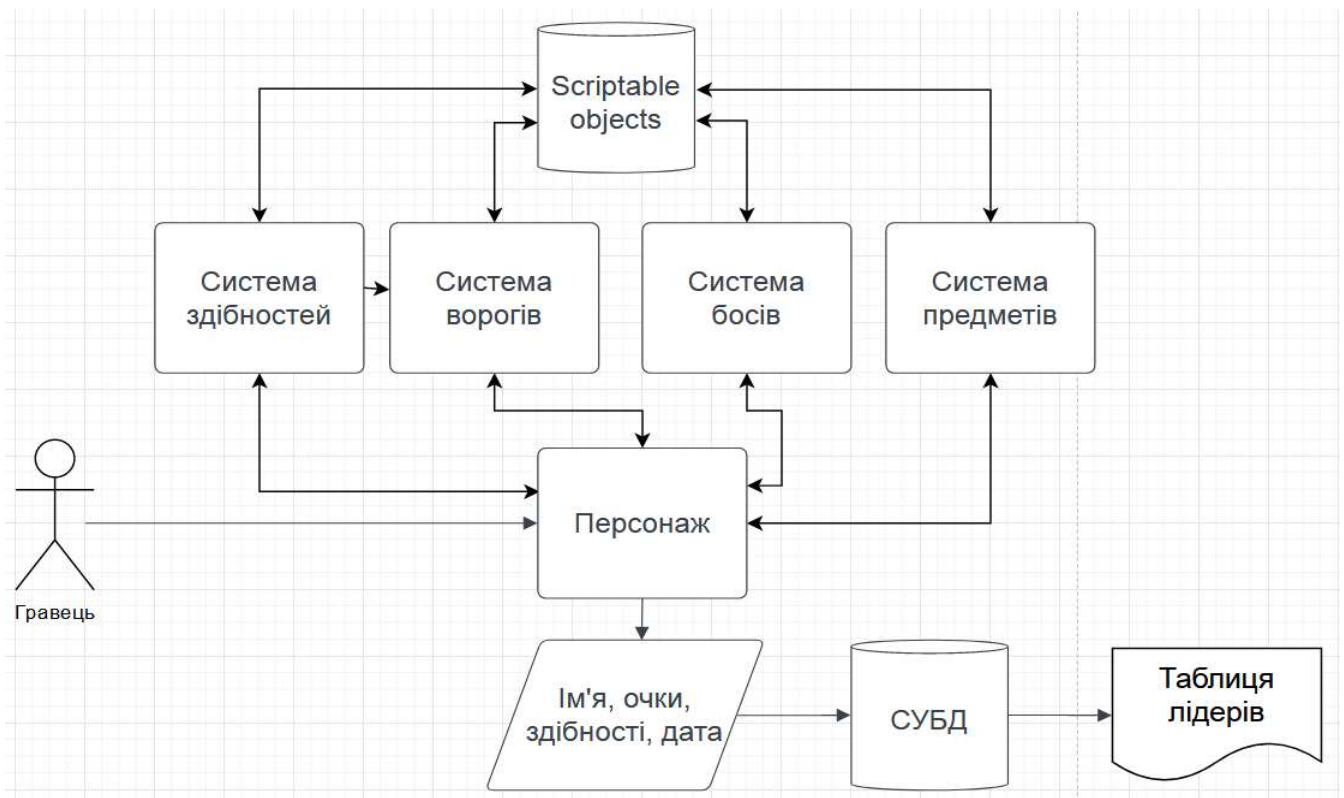


Рис. 1.5 – Схема запропонованої концепції інформаційної системи «Astro Survival»

Концепція гри полягає у тому, що є гравець та його персонаж – керуючи персонажем гравець повинен знищувати ворогів та ставати сильніше піднімаючи свій рівень та набуваючи могутніх здібностей. Гра теоретично не має кінця оскільки гравець ніяк не обмежений часом, лише ігровим полем, але при розумному використанні своїх здібностей персонаж гравця може жити вічно набиваючи собі очки та підіймаючись все вище у таблиці лідерів.

Гра має декілька систем завдяки яким вона функціонує:

- система здібностей (відповідає за усі здібності які гравець може отримати, а також за їх функціонал під час гри)
- система ворогів (відповідає за усіх ворогів, за їх пересування, поведінку під час зіткненням з гравцем та їх спосіб знищення гравця)
- система предметів (відповідає за предмети які можуть випасти з ворога, та функціонал цих предметів після підймання гравцем)
- система босів (відповідає за босів, їх пересуванням, їх логікою нанесення гравцю пошкодження, поведінкою при різній кількості здоров'я)

Кожен ворог, та бос оцінюється у різну кількість очків:

- базовий (1)
- живучий (2)
- швидкий (3)
- літаюча міна (2)
- троянське коло (спочатку 2, потім по 1)
- пила (3)
- боси (кожен бос має різну кількість очків, залежить від прожитого часу гравцем)

Кількість ворогів обмежена, але незважаючи на це неможливо постійно від них тікати оскільки вороги які опинилися занадто далеко від гравця автоматично переносяться у pool об'єктів звідки потім вони знову з'являться у новому місці поблизу гравця.

Уся інформація про здібності гравця, ворогів, босів та предметів знаходиться у Scriptable objects завдяки цьому зменшується навантаження на базу даних.

На основі представленої концепції інформаційної системи розроблюється технічний проект.

РОЗДІЛ II РОЗРОБКА ПРОЕКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ «ASTRO SURVIVAL»

2.1 Загальносистемні рішення

2.1.1 Діаграма варіантів використання

Одна гра, десятки, сотні, тисячі користувачів і усі використовують її по різному. Авжеж, розробник заклав у свою гру лише одну ціль – грати. Але не усі «грають» у ігри як прості користувачі. Багато людей шукають натхнення, методи використання тієї або іншої ігрової механіки або просто копіюють проект змінивши лише графіку та/або музикальний супровід.

Однак не зважаючи на ці фактори усі завантажують ігри та хоча б раз їх запускають. Розробники намагаються створити візуально зрозумілі відносини між гравцем, ворогами, об'єктами та, особливо, ігровим світом. Але на професійному рівні ці відносини описують завдяки «Діаграмі варіантів використання».

Діаграма варіантів використання (сценаріїв поведінки, прецедентів) є вихідним концептуальним поданням системи в процесі її проектування і розробки. Дана діаграма складається з акторів, варіантів використання і відносин між ними.

Діаграма варіантів використання представлена на рисунку 2.1

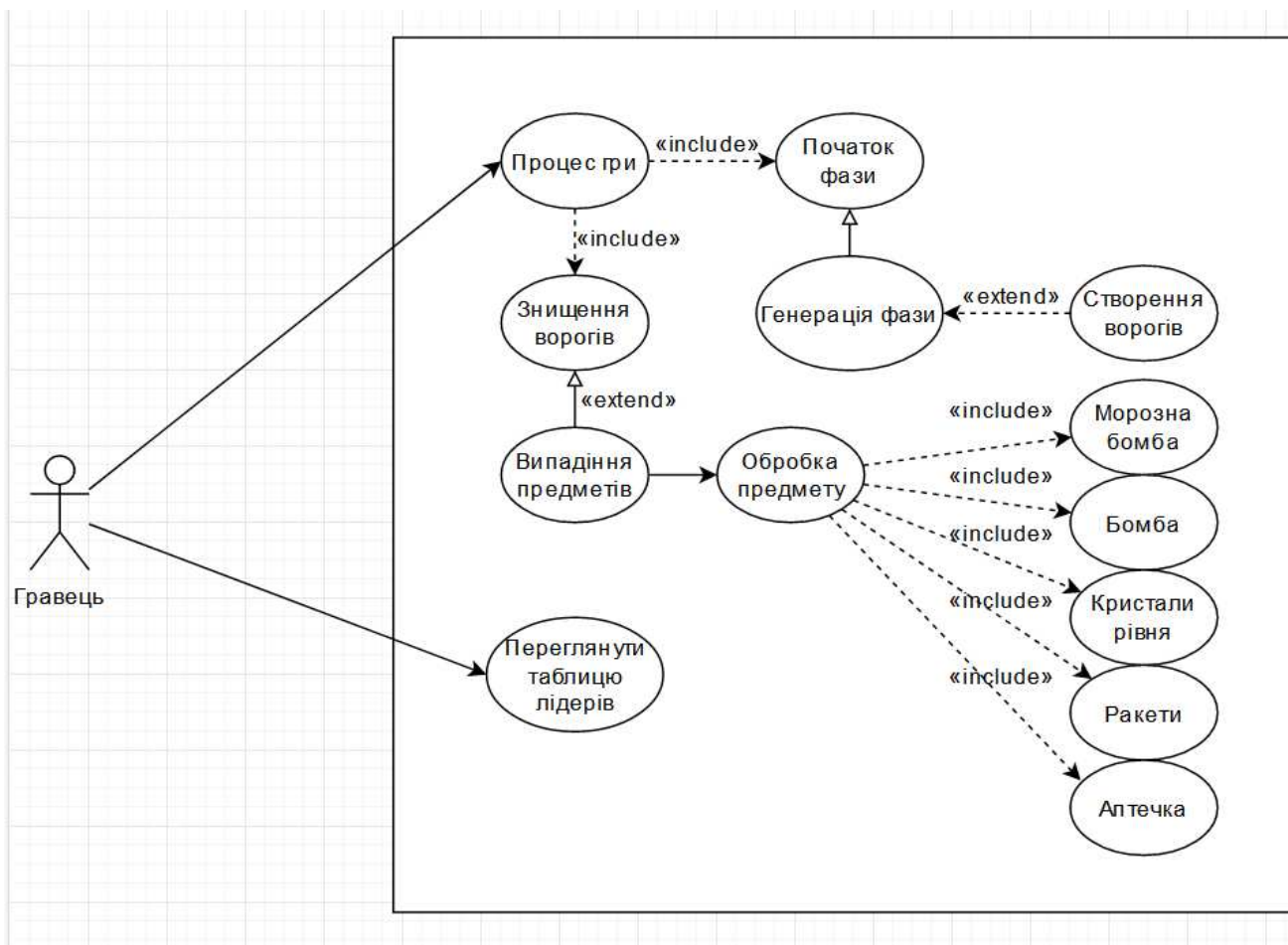


Рис. 2.1 – діаграма варіантів використання

Після запуску гри гравець опиняється у головному меню. Після натискання кнопки «Start» гравець переходить до самої гри. Персонаж з’являється на ігровому полі, водночас з гравцем починається фаза і вороги з’являються навколо гравця за межами камери. Гравцю потрібно їх знищувати, а після їх знищення може випасти один з предметів які гравцю потрібно піднімати аби стати сильніше. Коли кількість знищених ворогів досягає певного рівня або кількість ворогів які повинні були з’явитися – з’явилися: настає нова фаза.

2.1.2 Специфікація варіантів використання

Загальноприйнята структура специфікації (опису) варіанта використання включає в себе наступні елементи:

- Ім'я;
- Ідентифікатор;
- Короткий опис;

- Актори (головні і другорядні);
- Передумови;
- Основний потік;
- Альтернативні потоки;
- Післяумови;
- Спеціальні вимоги.

Беручи в увагу усі елементи структури специфікації можна розробити наступні специфікації варіантів використання (VV):

А) Специфікація VV «показати данні таблиці лідерів»:

Ім'я: Показати данні таблиці лідерів

ID: 0

Короткий опис: система виведе на екран таблиці лідерів.

Основна діюча особа: гравець

Другорядні дійові особи: ні

Передумови:

1) Повинен бути зв'язок з інтернетом

Основний потік:

1) VV починається коли гравець натисне кнопку «Leaderboard»

2) Система виведе гравцю локальну систему лідерів

Постумови:

1) Гравець натиснув кнопку «Global»

2) Система завантажила з БД глобальну таблицю лідерів

Альтернативні потоки:

1) Гравець повернувся у головне меню

Б) Специфікація VV «у змагальних цілях»:

Ім'я: у змагальних цілях

ID: 1

Короткий опис: система дасть гравцям змогу змагатися між собою

Основна діюча особа: гравець

Другорядні дійові особи: інші гравці

Передумови:

1) Повинен бути зв'язок з інтернетом

Основний потік:

1) ВВ почнеться коли гравець програє та введе свій нікнейм у спеціальне поле

2) Натиснувши кнопку «Back to menu» запуститься ВВ з ID - 0

Постумови:

1) Гравець почне нову гру без перевірки свого місця у рейтингу

Альтернативні потоки:

1) Гравець не зможе змагатися з іншими якщо не введе свій нікнейм.

В) Специфікація ВВ «скорочення часу»:

Ім'я: скорочення часу

ID: 2

Короткий опис: гравець скоротає час за грою аби відпочити від домашніх діл або роботи

Основна діюча особа: гравець

Другорядні дійові особи: ні

Передумови: ні

Основний

потік:

1) ВВ почнеться коли гравець запустить систему

Постумови:

1) Гравець програв та вимкнув систему

Альтернативні потоки: ні

Г) Специфікація ВВ «грати у гру»:

Ім'я: грати у гру

ID: 3

Короткий опис: знайомство з грою, отримання ігрового досвіду

Основна діюча особа: гравець

Другорядні дійові особи: ігровий персонаж, вороги, ігрові об'єкти

Основний потік:

1) ВВ почнеться коли гравець натисне кнопку «Start»

2) Система почне відробляти свою функцію

Постумови:

- 1) Гравець керує персонажем та знищує ворогів підіймаючи предмети та прокачуючи ігрового персонажа

Альтернативні потоки:

- 1) Гравець повернеться у головне меню

Д) Специфікація ВВ «максимальна прокачка персонажа»:

Ім'я: максимальна прокачка персонажа

ID: 4

Короткий опис: догнення максимальної прокачки гравця

Основна діюча особа: гравець

Другорядні дійові особи: ігровий персонаж, вороги, предмети

Основний потік:

- 1) ВВ почнеться коли запусниться ВВ з ID – 3.
- 2) Гравець керуючи ігровим персонажем – підіймає кристали рівня тим самим збільшує свій рівень та прокачуючи себе завдяки здібностям

Постумови:

- 1) Гравець досягне максимального рівня усіх свої здібностей

Альтернативні потоки:

- 1) Гравець програє

Аби будь-який з ВВ виконував свою роботу гравець повинен упевнитись у достатньому заряді свого пристрою аби під час виконання одного з ВВ пристрій не вимкнувся та дозволив гравцю отримати максимальне задоволення від конкретного ВВ.

2.2 Рішення з інформаційного забезпечення

2.2.1 Діаграма класів

Діаграма класів – це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, Такі як: класи, типи даних, їх зміст та їх відношення.

Діаграма класів представлена на рисунку 2.2

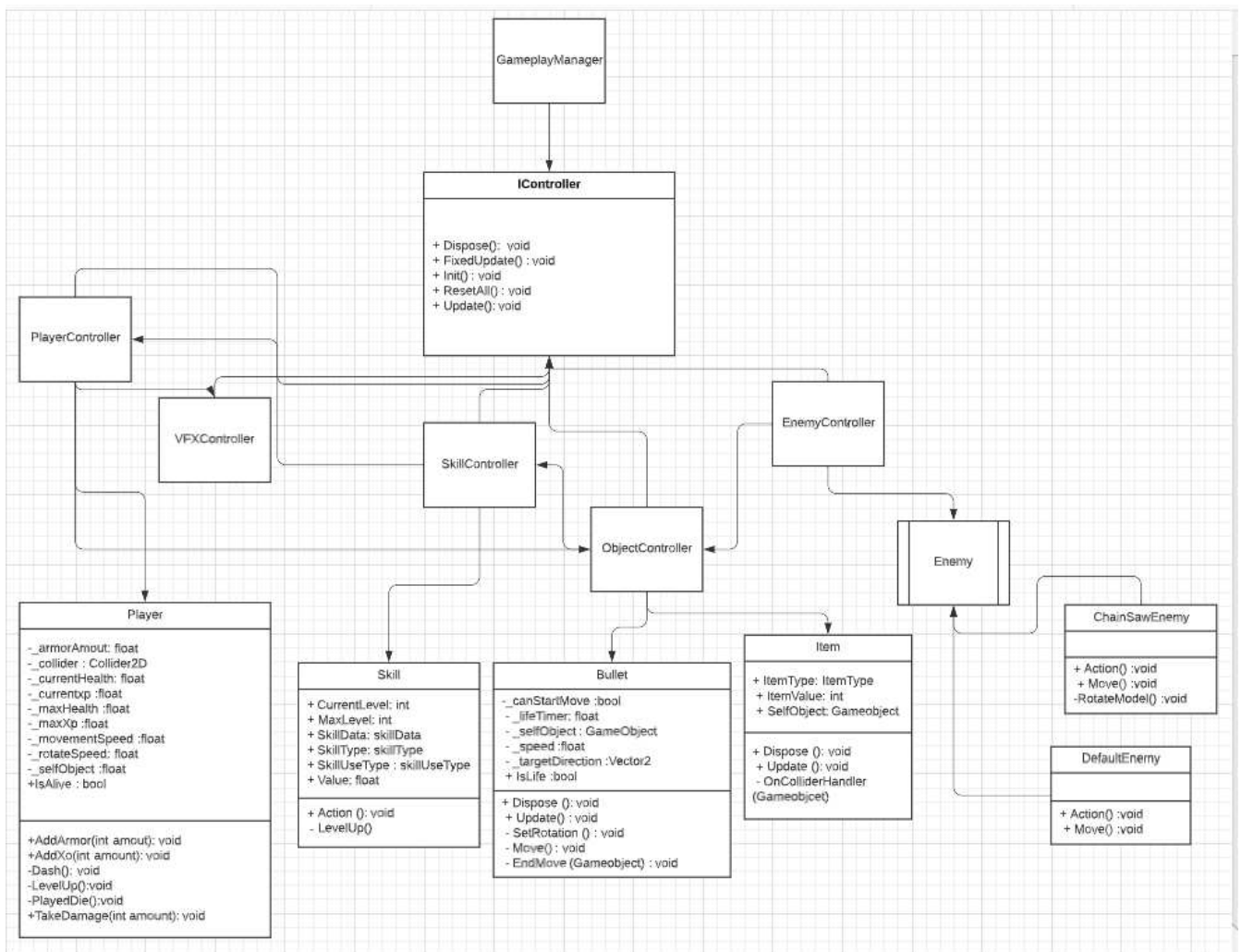


Рис. 2.2 – діаграма класів

2.2.2 Діаграма переходів станів

Діаграма переходів станів – це UML діаграма, що демонструє поведінку розроблюваної програмної системи при отриманні керуючих впливів (ззовні). За допомогою діаграм переходів станів можна моделювати подальше функціонування системи на основі її попереднього і поточного функціонування.

Діаграму переходів стану поразки представлено на рисунку 2.3

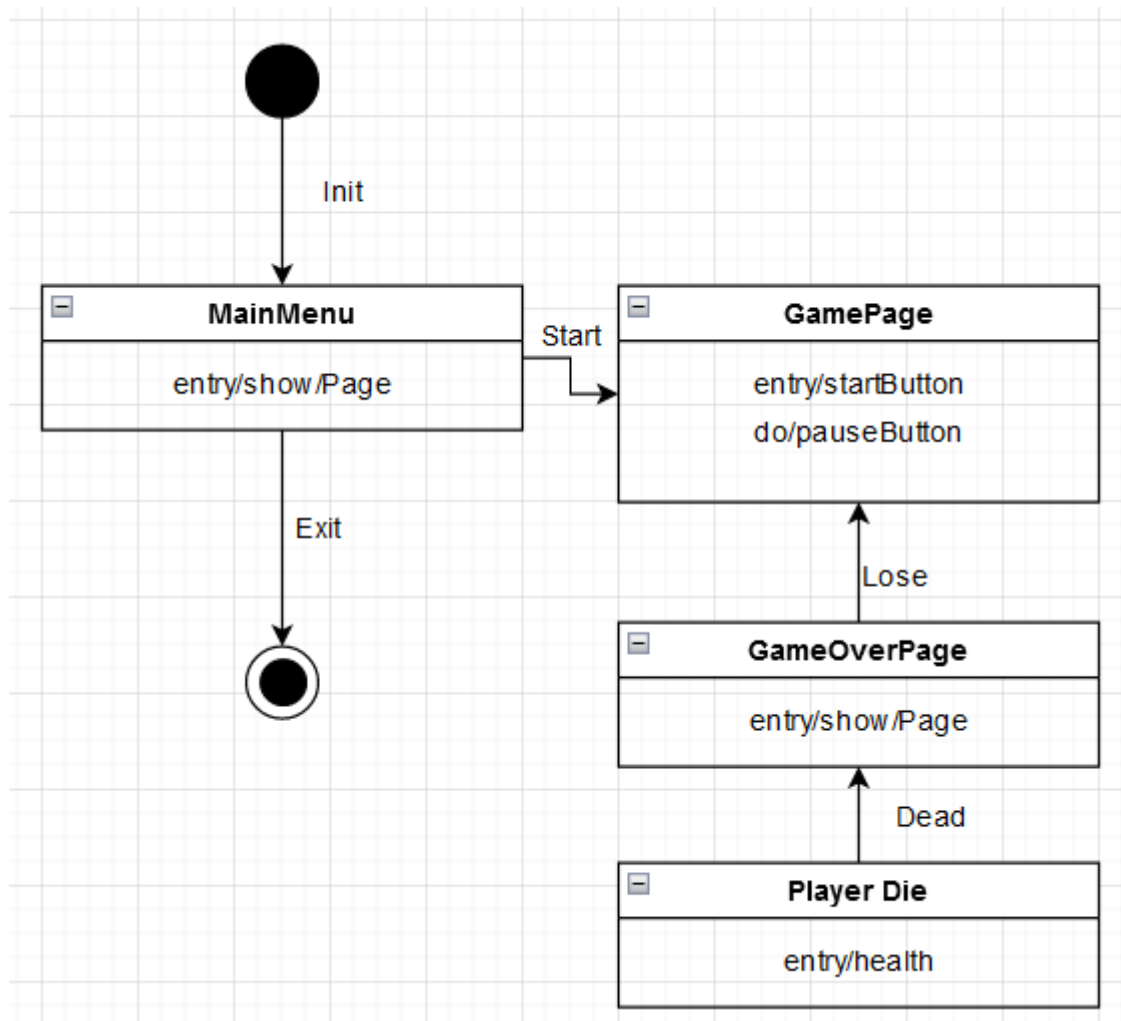


Рис. 2.3 – діаграма переходів стану поразки

На діаграмі переходів станів поразки показано процес гри, в якому після смерті героя відкривається спливаюче вікно програшу зі статистикою за матч і гравець має змогу перейти у головне меню.

Діаграму переходів стану активності гравця представлено на рисунку 2.4

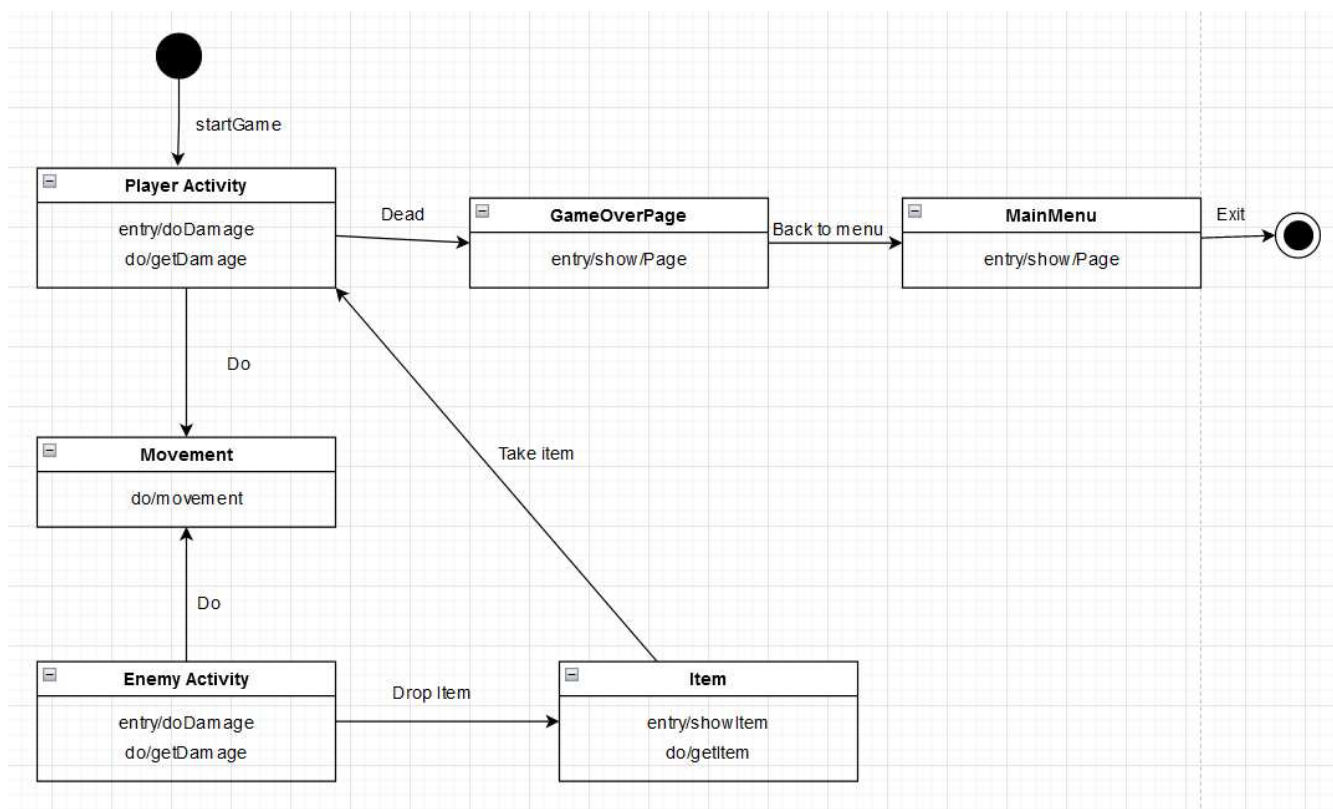


Рис. 2.4 – діаграма переходів станів активності гравця

На діаграмі переходів станів активності гравця. Для всіх персонажів у грі для кожної дії існує свій метод який обробляє цю дію. Після смерті головного героя відбувається поразка і гра закінчується.

Діаграма переходів станів процесу гри представлена на рисунку 2.5

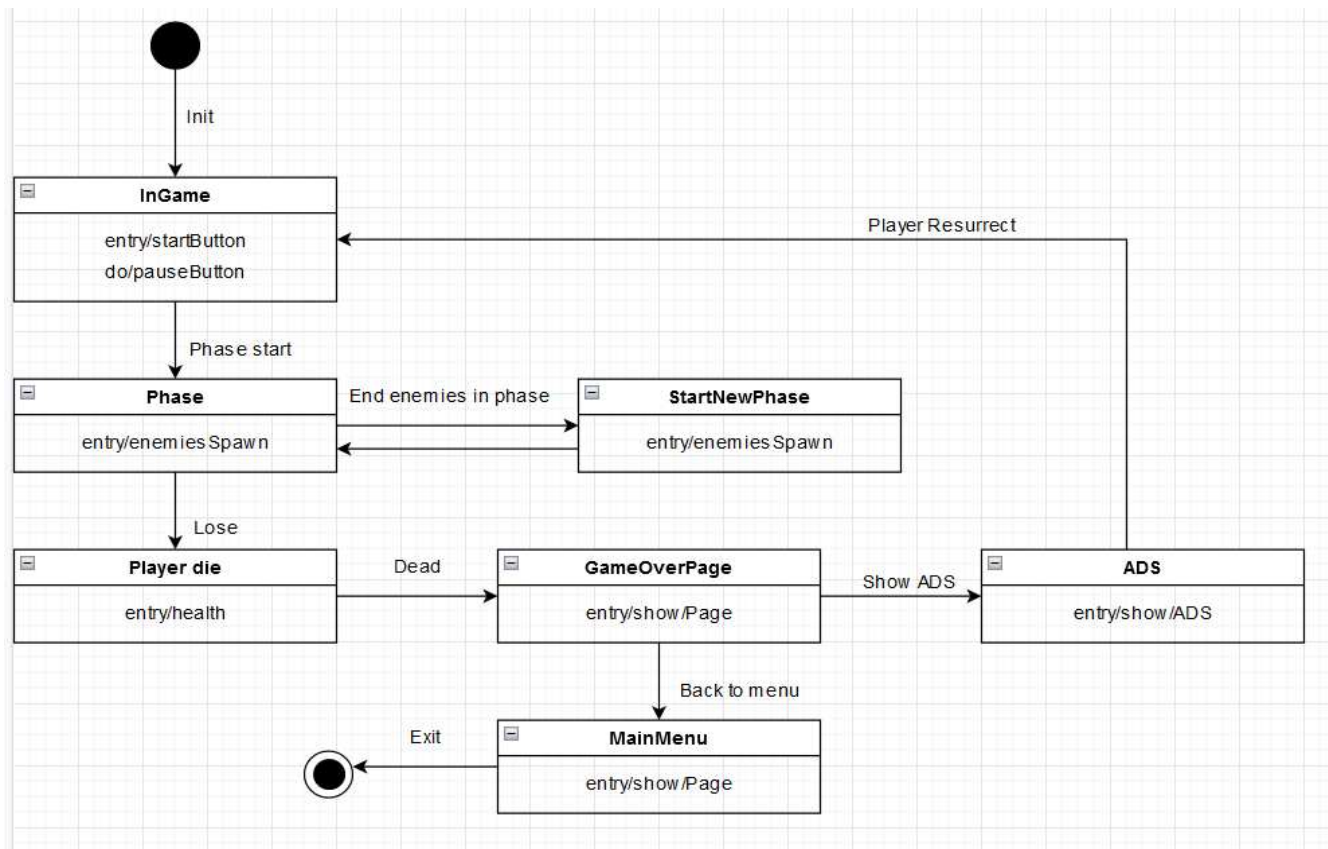


Рис. 2.5 – діаграма переходів станів процесу гри

На діаграмі переходів станів процесу гри показано процес гри. Після старту починається генерація фаз, на сцені з'являються вороги, з них випадають предмети, після поразки гра закінчується. Але можна переглянути реклами і продовжити гру.

2.2.3 Діаграма діяльності

Діаграма діяльності представлена на рисунку 2.6

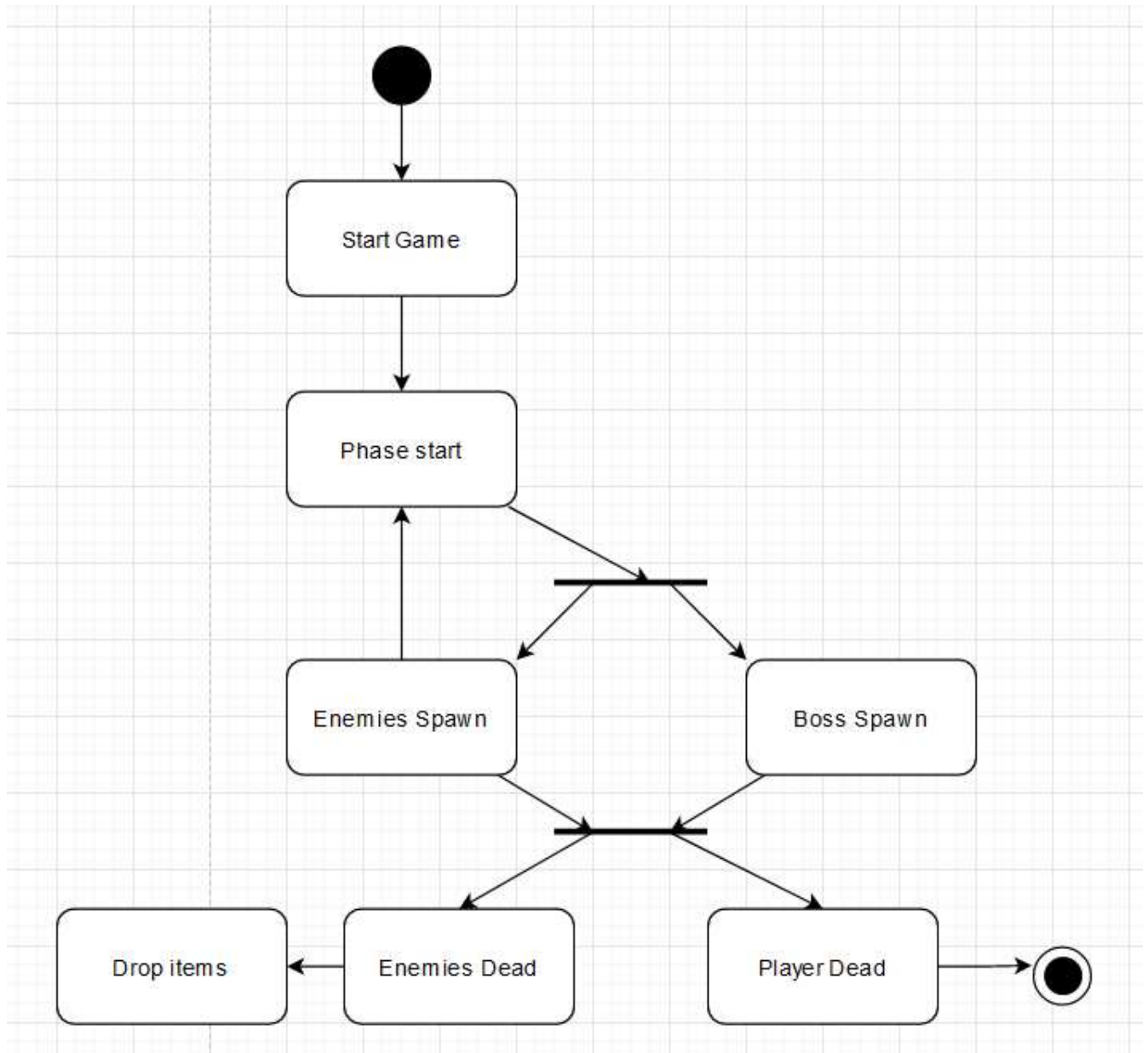


Рис. 2.6 – діаграма діяльності

Діаграма діяльності – це UML-діаграма, на якій показані дії, специфікація виконуваної поведінки у вигляді координованого послідовного і паралельного виконання підлеглих елементів - вкладених видів діяльності і окремих дій, з'єднаних між собою потоками, які йдуть від виходів одного вузла до входів

іншого. Діаграми діяльності складаються з обмеженої кількості фігур, з'єднаних стрілками.

2.2.4 Діаграма компонентів

Діаграма компонентів(Component diagram) - показує взаємозв'язки між модулями (логічними або фізичними), з яких складається система, що моделюється.

Діаграма компонентів представлена на рисунку 2.7

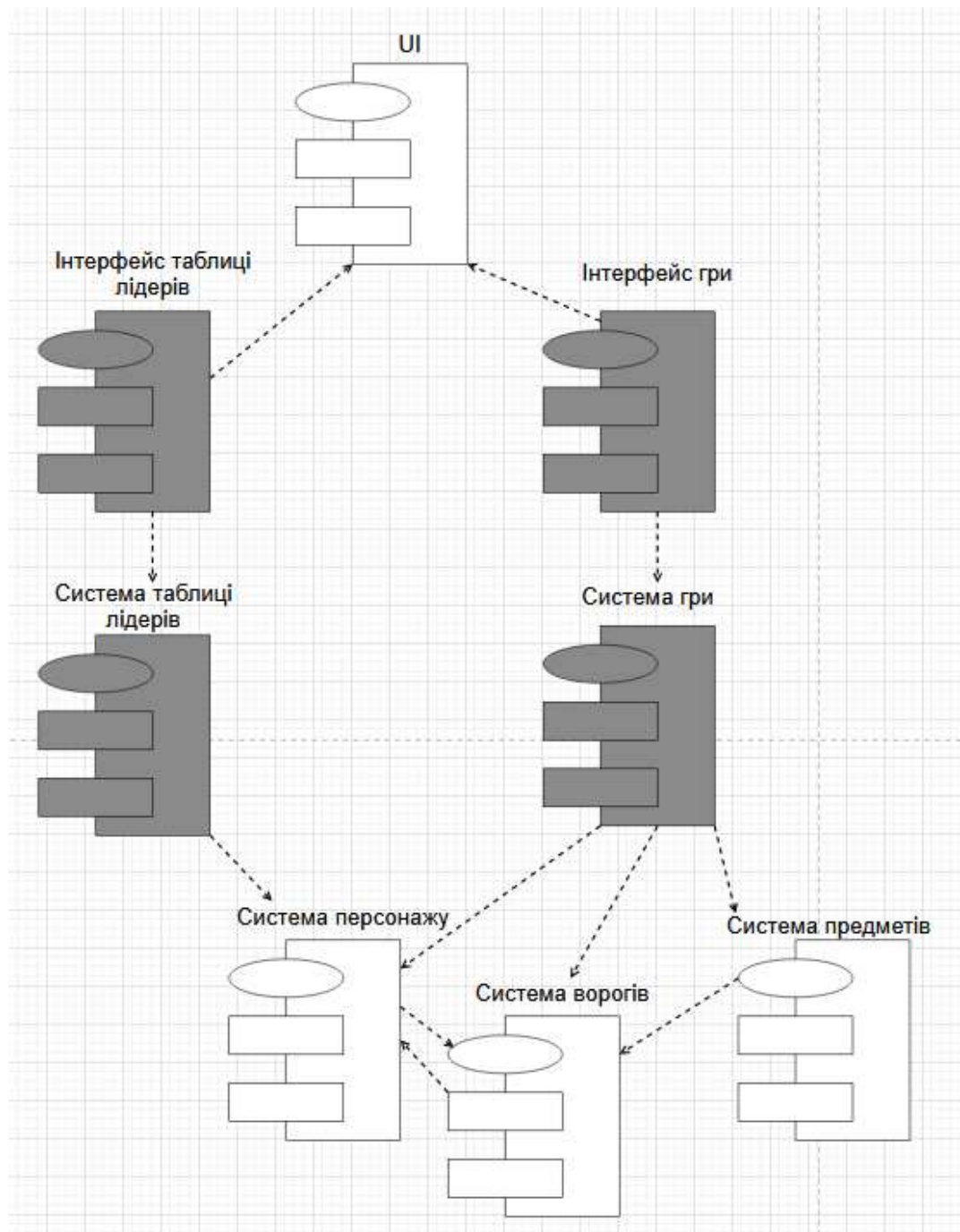


Рис. 2.7 – діаграма компонентів.

2.2.5 Діаграма розгортання

Діаграма розгортання — діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах.

Діаграма розвертання представлена на рисунку 2.8

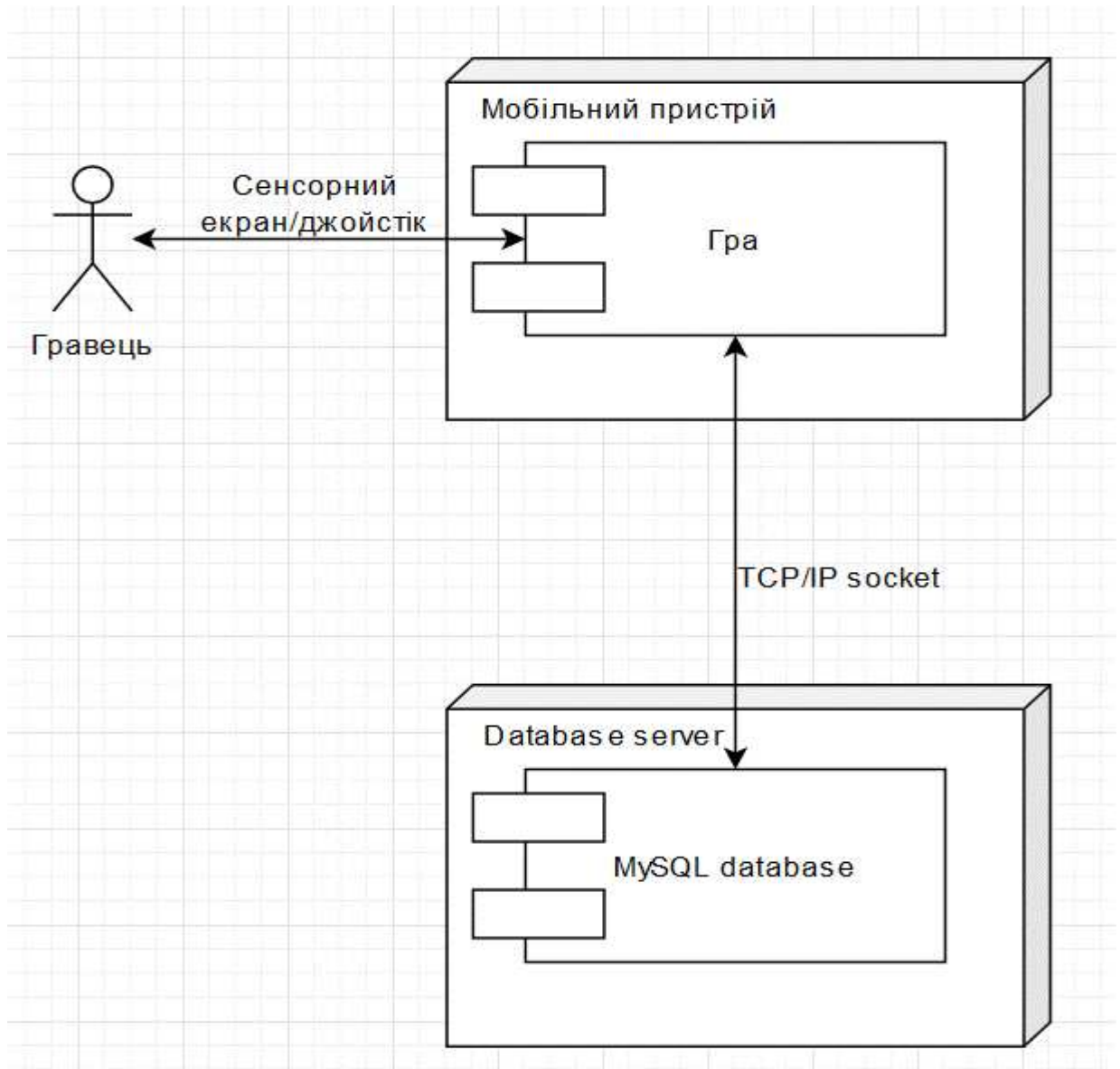


Рис. 2.8 - діаграма розвертання

2.3 Рішення з програмного забезпечення

Беручи до уваги переваги та недоліки різних програм для розробки ігор, можна зробити висновок, що Unity є найоптимальнішим засобом для створення мобільної гри, оскільки він володіє дуже великою бібліотекою, де можливо

знайти різні матеріали для гри: графіку, допоміжні інструменти та музику. На Unity можливо розробити гру, не маючи навиків в створенні 3D та 2D графіки. Також у проекті використовується ще одна особливість Unity ScriptableObject. Це контейнер даних який можна використовувати для збереження великих об'ємів даних, незалежно від екземпляру класу. Один із варіантів зменшити використання пам'яті у проекті, уникаючи копіювання значень. Це дуже корисно, коли у проекті є Prefab⁸ який зберігає незмінні дані в прикріплених скриптах.

C#[16] - це об'єктивно-орієнтовна мова програмування яка була створена у 1998 році інженерами компанії Microsoft. Ця мову програмування була створення для використання у платформі .NET. C# відноситься до сім'ї мови C та має схожий синтаксис. Мова має статеву типізацію, підтримує поліморфізм, перегрузку операторів, делегати, атрибути, івенти та мову запитів LINQ. У порівнянні з C++, C# набагато краще із-за того що має керовану пам'ять, тобто сам керує розподіленням пам'яті та запобігає витоку тощо.

Вибір мови пав на C# оскільки це:

- об'єктно-орієнтована, проста і водночас потужна мова програмування, яка дозволяє розробникам створювати багатофункціональні програми.
- C# ставитися до мов компілюваного типу, тому він має всі переваги таких мов.
- C# поєднує найкращі ідеї сучасних мов програмування Java, C++, Visual Basic тощо.
- Через велику різноманітність синтаксичних конструкцій та можливості працювати з платформою .Net, C# дозволяє швидше, ніж будь-яку іншу мову, розробляти програмні рішення.
- C# відрізняється надійністю та елегантністю.

Для реалізації таблиці лідерів потрібно створити базу даних де й будуть зберігатися усі рекорди гравців. Проаналізувавши різні СУБД вибір пав на phpMyAdmin оскільки вона має низку переваг над іншими СУБД:

- доступна;
- має простий та інтуїтивний інтерфейс;

- великий функціонал;
- швидкість;
- зручність у маніпуляції даними.

РОЗДІЛ III РЕАЛІЗАЦІЯ ПРОЕКТУ АВТОМАТИЗОВАНОЇ СИСТЕМИ

Було розроблено технічне завдання на розробку та описано в додатку Б. Текст програми представлено в додатку В.

В інформаційній системі реалізовано: запит до СУБД та отримання таблиці лідерів – відбувається завдяки спеціального вікна у якому таблиця відображена (рис 3.3), пересування гравця – виконується завдяки джойстику (рис 3.2), знищення ворогів – гравець має автоматичну зброю яка вистрілює у напрямку руху гравця коли попереду знаходиться ворог, вибивання предметів – виконано завдяки псевдо випадковості яка обчислюється кожного разу після смерті ворога, збільшення рівня гравця та отримання могутніх здібностей – гравець отримує відсоток підймання рівня підіймаючи *experience crystal's* та після досягнення 100% гравець підіймає свій рівень та має змогу отримати одну з трьох здібностей на вибір. Можливість перегляду реклами заради відродження персонажу – виконано завдяки *Unity Ads* , обмеження ігрової зони – на кінцях ігрової зони присутній пустотній простір в якому неможливо знаходитись, якщо гравець не покине цю зону протягом 5 секунд – гравця поверне до центру ігрової зони.

Графічний інтерфейс – UI⁹ дуже важлива складова будь-якої гри адже перше на що гравець звертає увагу це на інтерфейс. Тому до його розробки відносяться дуже серйозно. Інтерфейс повинен бути у першу чергу зручним, адже якщо, наприклад, зробити кнопку гарною, але маленькою то людині з великими пальцями буде складно на неї натиснути. Тому аби зробити інтерфейс зручним кнопки повинні бути достатньо великими аби будь-якій людині було зручно натискати їх. Кожна сторінка та спливаюче вікно інтерфейсу повинно бути представлене завдяки великій назві тим самим допомагаючи гравцю орієнтуватися в інтерфейсі.

У проєкті буде реалізовано три ігрові сторінки, та чотири спливаючих вікна. Інтерфейс сторінки головно меню представлено на рисунку 3.1[4]



Рис. 3.1 – інтерфейс сторінки головного меню

Назва гри, велика та приваблива. Кнопки, достатньо великі для зручного натискання. Шрифт зручний до читання. Кнопки розташовані таким чином за для зручності натискання.

- «Start» - переходить до ігрового вікна.
- «Upgrades» - переходить до вікна прокачки гравця за ігрові монети(не реалізовано)
- «Leaderboard» - переходить до вікна таблиці лідерів
- «Credits» - переходить до вікна інформації про розробників (не реалізовано)
- «Look Ads to support developers» - вмикає рекламу від Unity Ads. Тим самим ви підтримуєте розробників.

Інтерфейс ігрового вікна представлено на рисунку 3.2[4]

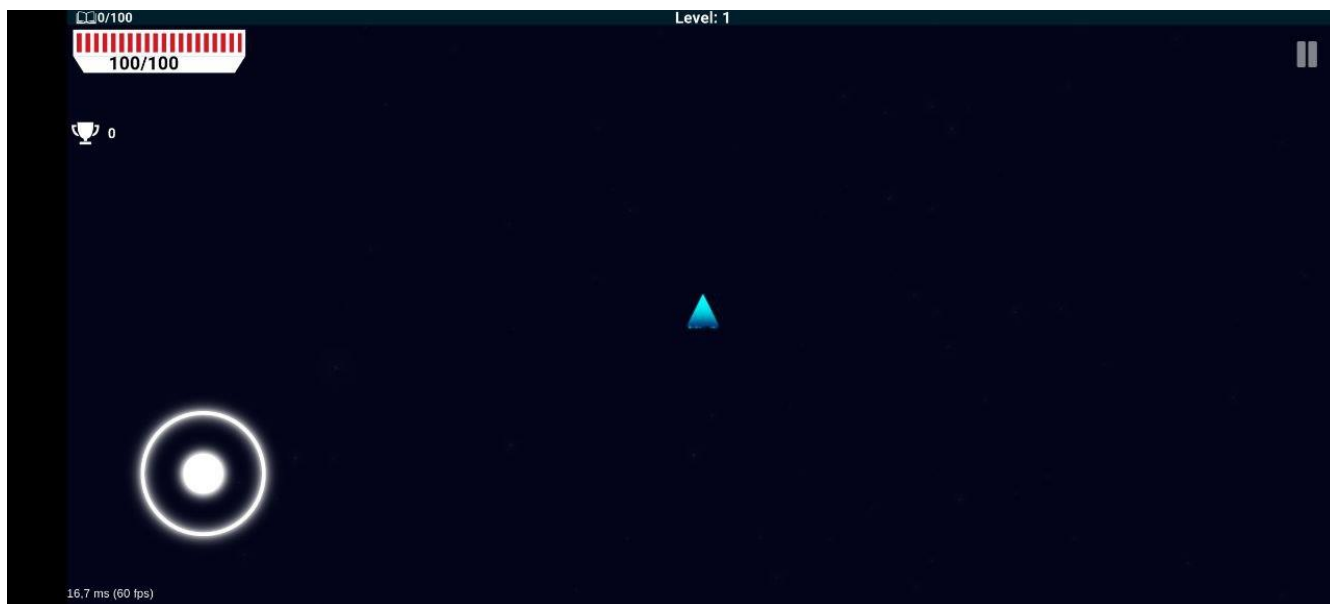


Рис. 3.2 – інтерфейс сторінки ігрового вікна

Керування персонажем виконується завдяки джойстику, який спеціально зроблений великим. Кнопка паузи справа зверху аби призупини гру на необмежену кількість часу. Графічне відображення зроблене для наступних пунктів інтерфейсу:

- Здоров'я гравця – зверху зліва червона смуга.
- Очки гравця – відображене у вигляді кубка з лівої сторони екрану.
- Прогрес рівня гравця – смуга із заповненням зверху, розтянута на усю ширину екрана.

Текстове відображення наступних елементів інтерфейсу:

- Очки гравця
- Прогрес рівня гравця (€/має бути)
- Здоров'я гравця

Інтерфейс сторінки лідерів представлено на рисунку 3.3

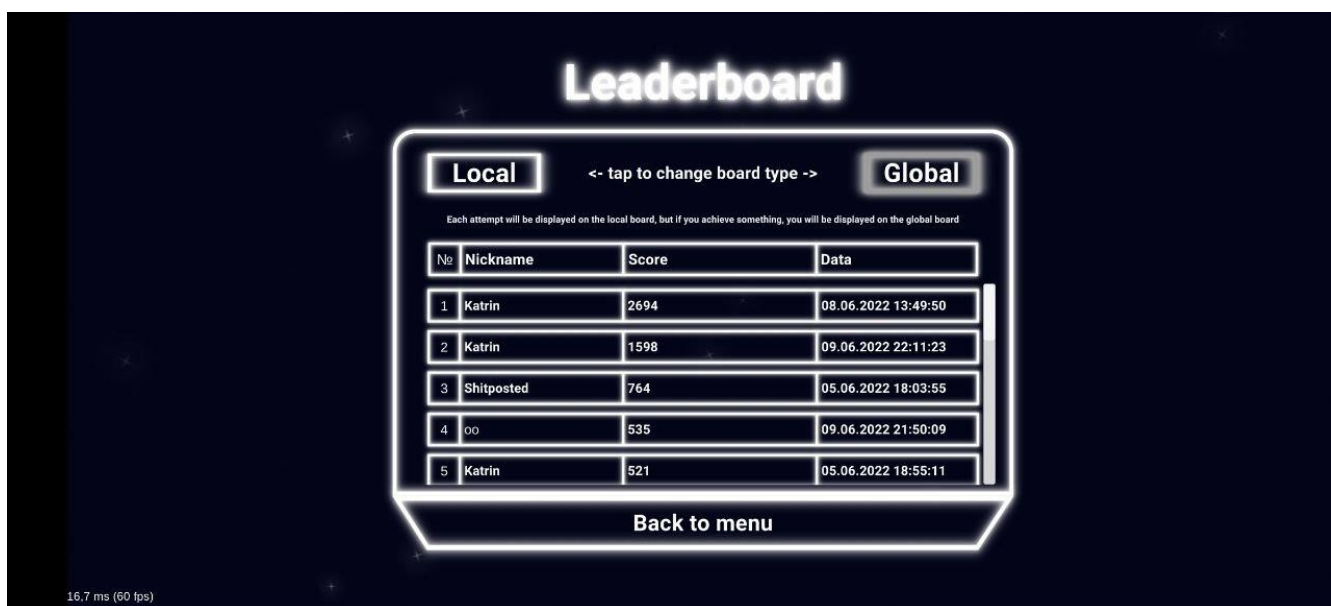


Рис. 3.3 – інтерфейс сторінки лідерів

Перш за все гравці у таблиці представляють собою список, а кожен об’єкт у списку має у собі наступні текстові поля:

- Номер
- Нікнейм гравця
- Кількість очків
- Дата занесення рекорда у таблицю

Також присутня кнопка повернення у головне меню аби гравець міг переміщуватись між сторінками. Аби розділити локальну та глобальну таблиці лідерів було запроваджено дві кнопки, кожна з яких активує свою таблицю лідерів. Адже якщо відображати одразу обидві таблиці – це буде навантажувати інтерфейс що буде запобігати легкому читанню інформації.

Спливаюче вікно програшу представлене на рисунку 3.4[4]

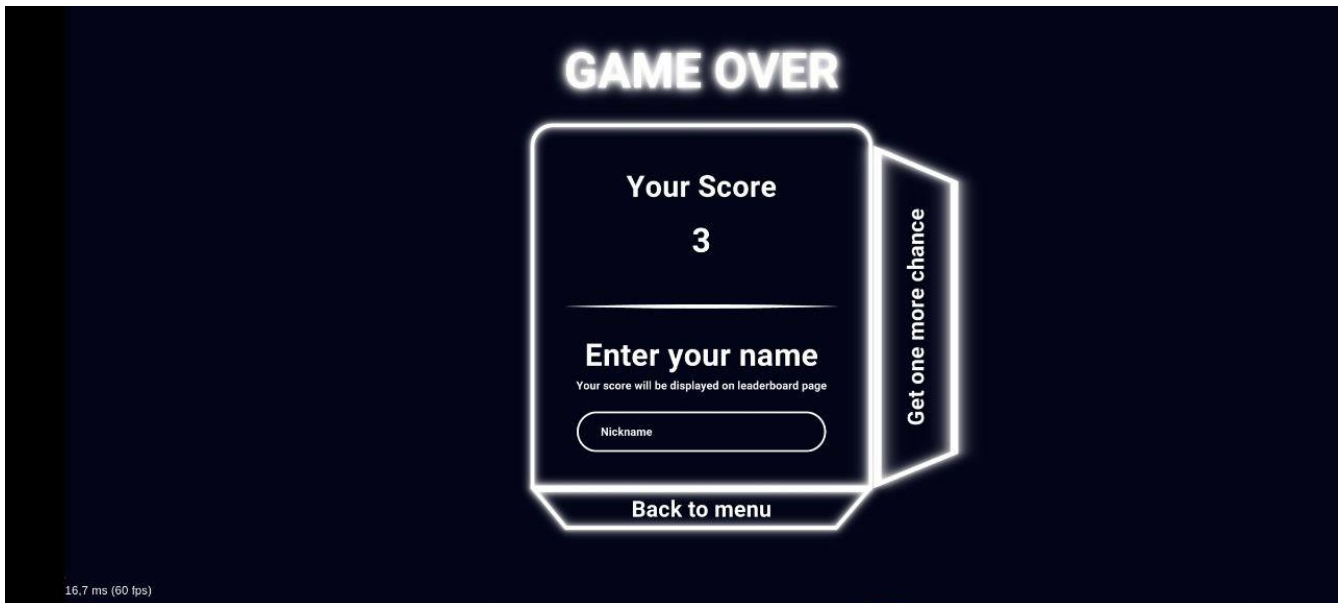


Рис. 3.4 – інтерфейс спливаючого вікна програшу

Має кнопку повернення у головне меню. Кнопку перегляду реклами задля відродження гравця з повним здоров'ям. Текстове відображення зароблених гравцем очків. Поле для введення никнейма гравця аби той попав у таблицю лідерів (якщо никнейм не ввести у таблиці лідерів гравець не відобразиться).

Спливаюче вікно паузи представлено на рисунку 3.5[4]

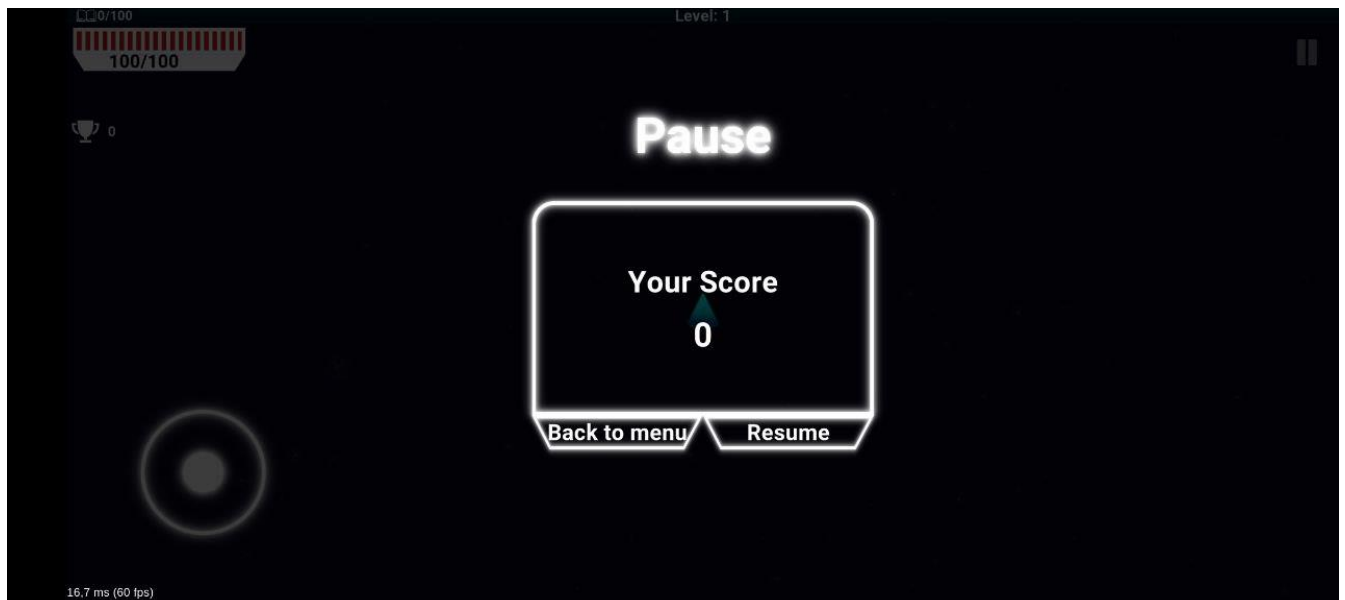


Рис. 3.5 – інтерфейс спливаючого вікна паузи

Гра зупиняється аби гравець міг відпочити. У текстовому форматі відображено кількість зароблених гравцем очків. Також наявні дві кнопки – повернення до гри та повернення у головне меню.

Інтерфейс спливаючого вікна підняття рівня представлено на рисунку 3.6[4]

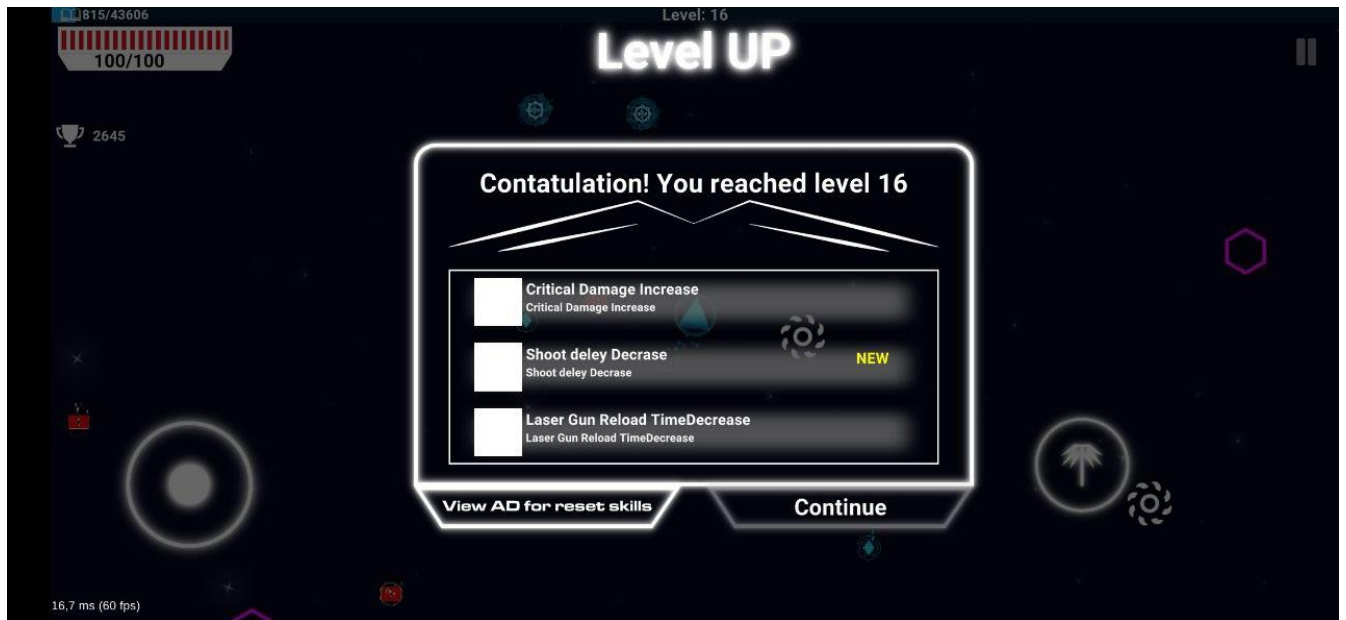


Рис. 3.6 – інтерфейс спливаючого вікна підняття рівня

Гра зупиняється аби гравець міг подумати та вибрати одну з трьох здібностей які представлені трьома кнопками. У текстовому форматі представлені:

- Назва здібності
- Короткий опис здібності
- Позначка нової здібності(з'являється лише для здібностей які гравець жодного разу не вибирав). Після вибору однієї із здібностей кнопка підтвердження, водночас представляє собою кнопку продовження гри – стає інтерактивною й її можна натиснути. У той же час кнопка перегляду реклами задля зміни здібностей інтерактивна доки гравець на неї не натисне(перегляд реклами можливий лише один раз).

Інтерфейс спливаючого вікна відкриття скриньки представлений на рисунку 3.7[4]

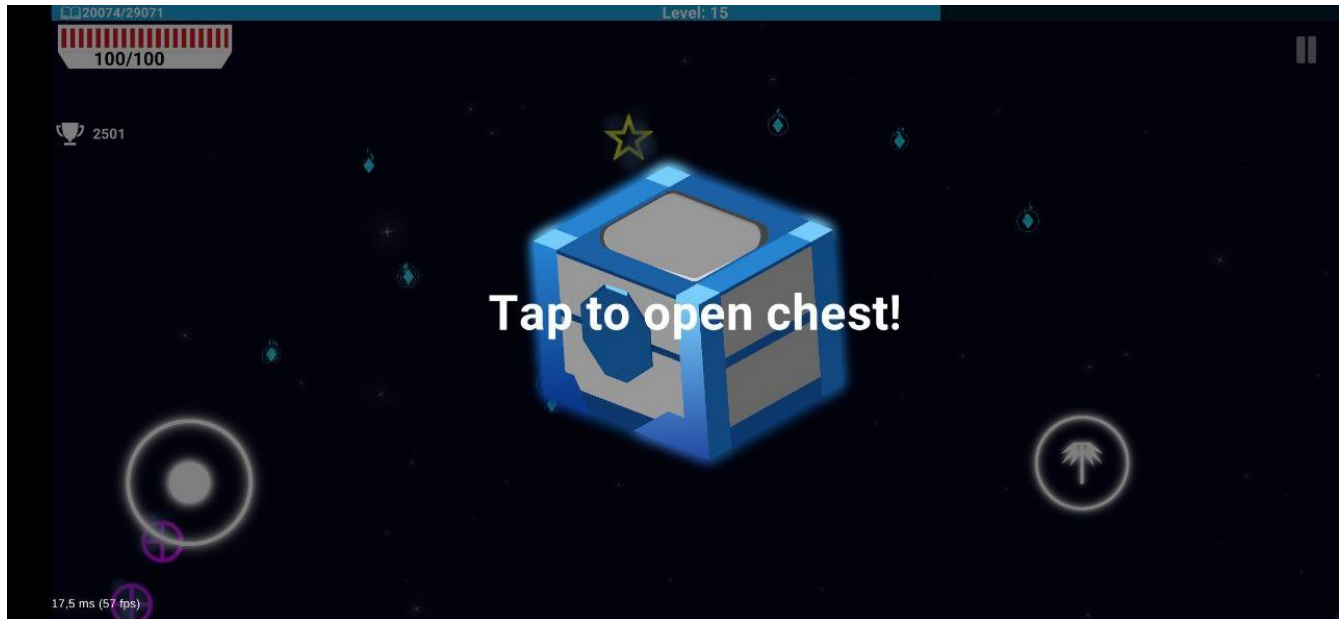


Рис. 3.7 – інтерфейс спливаючого вікна відкриття скриньки

Спливаюче вікно з'являється одразу після підняття скриньки. Гра у цей час зупиняється.

Інтерфейс спливаючого вікна відкритої скриньки представлений на рисунку 3.8[4]

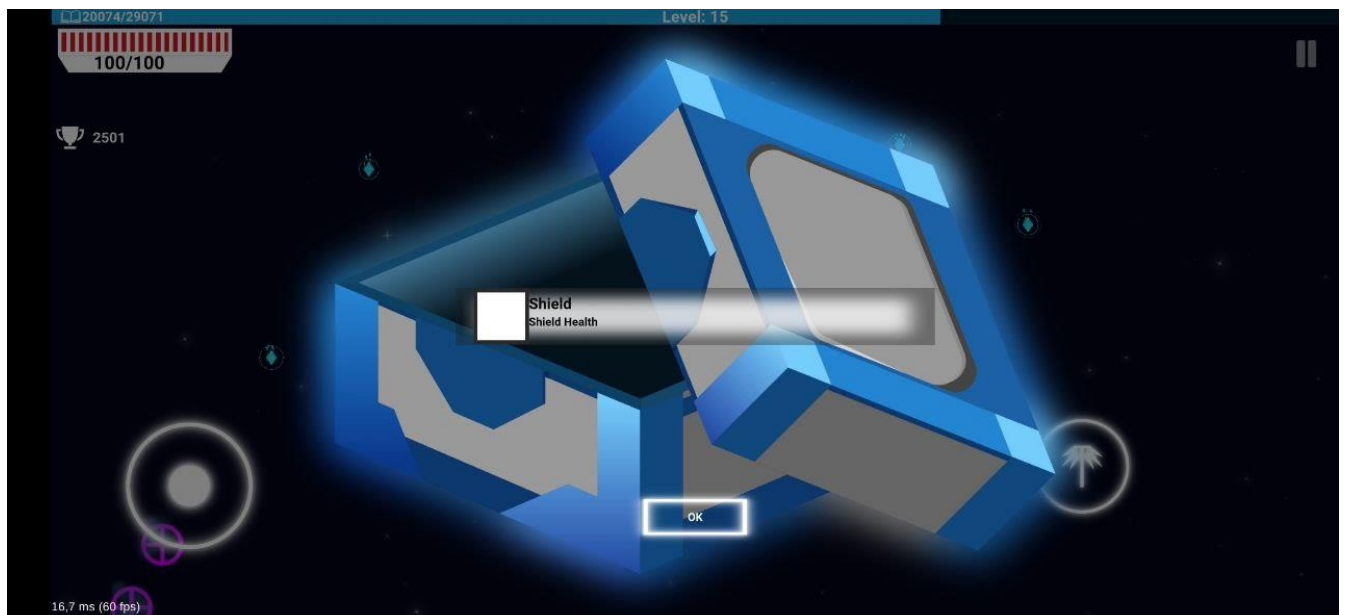


Рис. 3.8 – інтерфейс спливаючого вікна відкритої скриньки

Відчинена скринька відображає здібності які випали, а також одну єдину кнопку задля продовження гри при її натисканні.

Завдяки інтуїтивно зрозумілому інтерфейсу користувач з будь яким рівнем зможе легко розібратися у грі.

Інструкція користувача представлена у додатку Г.

РОЗДІЛ IV ОХОРОНА ПРАЦІ

4.1 ВСТУП

Важливим соціально-економічним завданням є охорона праці. Охорона праці передбачає систему правових, технічних економічних та санітарно-гігієнічних заходів, спрямованих на забезпечення не лише безпечних, а й здорових умов праці. Охорона праці досліджує трудовий процес з позиції забезпечення життя та здоров'я трудящих [13].

Раціональне поєднання можливостей людини та характеристик машин, відповідний розподіл функцій у середині системи, суттєво підвищують її ефективність. Взаємодія людини та техніки в системі виробництва повинна розглядатися при проектуванні та створенні безпечних умов праці, при вирішенні завдань оптимізації праці.

Метою даного розділу є аналіз умов праці, технологічних процесів, апаратури і обладнання на робочому місці програміста. На основі такого аналізу визначаються небезпечні ділянки, співвідношення умов праці на робочому місці до нормативних даних, можливі аварійні ситуації і розробляються заходи щодо їх усунення або обмеження наслідків.

Оскільки мобільна гра у жанрі Casual Action розробляється у лабораторії, було б доцільно розглянути параметри лабораторії та співвідносити їх з необхідними параметрами для розробки забезпечення за комп'ютером.

Задачами, що розглядаються в даному розділі, є:

- визначення оптимальних умов праці;
- аналіз робочого середовища відповідно до нормативів;
- розрахунок значення штучного освітлення та аналіз результатів.

4.2 Аналіз шкідливих та небезпечних факторів на робочому місці програміста

Основним робочим процесом при виконанні роботи є робота на персональному комп'ютері. Довготривала робота на комп'ютері може вплинути поганим чином на організм людини та призвести до небезпечних захворювань.

Серед небезпечних факторів слід виділити наступні [11]:

- а) санітарно-гігієнічні фактори;
- б) ергономічні фактори;
- в) психофізіологічні фактори.

Аналіз умов роботи допоможе визначити, які дії необхідно провести для покращення умов роботи до нормативних, таких, що відповідають закону про охорону праці.

Мікроклімат виробничих приміщень

Мікроклімат виробничих приміщень – умови внутрішнього середовища виробничих приміщень, що впливають на тепловий обмін працюючих з оточенням шляхом конвекції, кондукції, теплового випромінювання та випаровування вологи. Ці умови визначаються поєднанням температури, відносної вологості та швидкості руху повітря, температури оточуючих людину поверхонь та інтенсивністю теплового (інфрачервоного) опромінення.

Упродовж робочого часу в робочих приміщеннях забезпечується мікроклімат, що відповідає фізіологічним потребам організму працюючих, із врахуванням енергетичних витрат на виконувану роботу згідно з ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень».

У приміщеннях для відпочинку, для чергового персоналу, їдальнях, приміщеннях санітарно-побутового призначення і для надання медичної допомоги температура повинна відповідати призначенню цих приміщень відповідно до СНиП 2.09.04-87.

Мікрокліматичні умови виробничих приміщень характеризуються такими показниками [11]:

- температура повітря;
- відносна вологість повітря;
- швидкість руху повітря;
- інтенсивність теплового (інфрачервоного) опромінення;
- температура поверхні.

За ступенем впливу на тепловий стан людини мікрокліматичні умови поділяють на оптимальні та допустимі.

Для робочої зони виробничих приміщень встановлюються оптимальні та допустимі мікрокліматичні умови з урахуванням важкості виконуваної роботи та періоду року.

Оптимальні мікрокліматичні умови – поєднання параметрів мікроклімату, які при тривалому та систематичному впливі на людину забезпечують зберігання нормального теплового стану організму без активізації механізмів терморегуляції.

Допустимі мікрокліматичні умови – поєднання параметрів мікроклімату, які при тривалому та систематичному впливі на людину можуть викликати зміни теплового стану організму, що швидко минають і нормалізуються та супроводжуються напруженням механізмів терморегуляції в межах фізіологічної адаптації.

Категорія робіт – розмежування робіт за важкістю на основі загальних енерговитрат організму.

Легкі фізичні роботи (категорія I) охоплюють види діяльності, при яких витрата енергії дорівнює 105-140 Вт (90-120 ккал/год.) – категорія Ia та 141-175 Вт (121-150 ккал/год.) – категорія Ib. До категорії Ia належать роботи, що виконуються сидячи і не потребують фізичного напруження. До категорії Ib належать роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням [11].

Фізичні роботи середньої важкості (категорія II) охоплюють види діяльності, при яких витрата енергії дорівнює 176-232 Вт (151-200 ккал/год.) – категорія IIa та 233-290 Вт (201-250 ккал/год.) – категорія IIб. До категорії IIa

належать роботи, пов'язані з ходінням, переміщенням дрібних (до 1 кг) виробів або предметів в положенні стоячи або сидячи і потребують певного фізичного напруження. До категорії Пб належать роботи, що виконуються стоячи, пов'язані з ходінням, переміщенням невеликих (до 10 кг) вантажів та супроводжуються помірним фізичним напруженням [11].

Важкі фізичні роботи (категорія III) охоплюють види діяльності, при яких витрати енергії становлять 291-349 Вт (251-300 ккал/год.). До категорії III належать роботи, пов'язані з постійним переміщенням, перенесенням значних (понад 10 кг) вантажів, які потребують великих фізичних зусиль.

Робота над дипломним проектом відноситься до категорії робіт Ia (легкі роботи), так як виконується сидячи і потребує невеликих витрат енергії: 75 ккал/год.

Для виявлення інформації про те, чи відповідає робоче приміщення вказаним нормам, наводиться таблиця 3.1.

Допустимі величини температури, відносної вологості та швидкості руху повітря в робочій зоні виробничих приміщень для категорії робіт Ia на постійних робочих місцях

Таблиця 3.1

Допустимі величини температури, відносної вологості та швидкості руху повітря в робочій зоні виробничих приміщень для категорії робіт Ia на постійних робочих місцях [11]

Період року	Температура, град.С		Відносна вологість (%) на робочих місцях	Швидкість руху (м/сек) на робочих місцях
	Верхня межа	Нижня межа		
Холодний період року	25	21	75	не більше 0,1
Теплий період року	28	22	55 – при 28 град. С	0,2-0,1

Згідно даних таблиці 3.1 можна зробити висновок, що робоче приміщення повністю задовольняє оптимальним умовам мікроклімату.

Освітленість при роботі за комп'ютером

Робота з комп'ютером супроводжується тривалими зоровими навантаженнями і негативно впливає на здоров'я очей. Правильно виконане освітлення робочого місця здійснює позитивний вплив на психофізіологічний стан людини, сприяє підвищенню ефективності та високій працездатності. Досягнення оптимальних умов роботи проводиться шляхом забезпечення природнього освітлення в світлий час доби та сприятливого штучного освітлення в темний період доби.

Освітлення робочих зон повинно відповідати вимогам ДБН В.2.5-28-2006 «Природне і штучне освітлення», затверджених наказом Міністерства будівництва, архітектури та житлово-комунального господарства України від 15.05.2006 № 168.

Робочі зони, в яких працівникам у випадку відключення штучного освітлення може загрожувати небезпека, обладнуються достатнім аварійним освітленням [15].

Для забезпечення умов, що необхідні для зорового комфорту, в системі освітлення повинні бути реалізовані наступні вимоги [15]:

- рівномірне освітлення;
- оптимальна яскравість;
- відсутні відблиски;
- правильний контраст;
- правильна кольорова гамма;
- відсутній стробоскопічний ефект або пульсація світла.

Робота над дипломним проектом відноситься до IV розряду зорових робіт (мінімальний розмір об'єкту, що розглядається – товщина напису 0,3 мм; розряд зорової роботи – робота високої точності) при великому контрасті та світлому фоні (під розряд зорової роботи «г»).

Робоче місце освітлюється у світлий період доби через вікно (природнє бокове освітлення), яке виходить на південно–західний бік, і сонячне світло не перекривається посторонніми об'єктами. У темний час доби штучне освітлення забезпечується люмінесцентними лампами. Часта переадаптація ока до різних значень яскравості і дистанцій є одним з головних негативних факторів при

роботі з дисплеєм. Неприятливим фактором світлового середовища є невідповідність нормативним значенням рівнів освітленості робочих поверхностей стола, екрану, клавіатури. Часто на екрані помітне дзеркальне відображення джерел світла і оточуючих предметів. Враховуючи всі дані, можна зробити висновок, що робота ускладнюється та може призвести до порушень основних функцій зорової системи.

У лабораторії для розробки дипломної роботи було доцільно обрахувати рівень штучного освітлення за нормами згідно приміщень для роботи за персональними комп'ютерами.

Розрахунок освітленості. Нормованим параметром природного освітлення являється коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт.

Для штучного освітлення нормованим параметром виступає $E_{\min}$ – мінімальний рівень освітленості, та $K_{\text{п}}$ – коефіцієнт пульсації світлового потоку, який не повинний бути більшим ніж 20 %.

Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 400-700 лк [15].

Для штучного освітлення у приміщенні використовуються люмінесцентні лампи. Розрахунок штучного освітлення проведемо для кімнати площею 16 м², ширина якої складає 4 м, довжина – 4 м, висота – 3 м.

Скористаємося методом використання світлового потоку. Для визначення потрібної кількості світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою (1.1)

$$F = \frac{E * K * S * Z}{\eta}, \quad (1.1)[15]$$

де F – світловий потік, що розраховується, лм;

E – нормована мінімальна освітленість, лк; $E = 400$ лк;

S – площа освітлюваного приміщення (у нашому випадку $S = 16$ м²);

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1,1... 1,2, в нашому випадку $Z = 1,1$);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку $K = 1,5$);

η – коефіцієнт використання світлового потоку, (виражається відношенням світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп, і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{\text{стін}}$) і стелі ($\rho_{\text{стелі}}$)), значення коефіцієнтів дорівнюють $\rho_{\text{стін}} = 40 \%$ і $\rho_{\text{стелі}} = 60 \%$.

Обчислимо індекс приміщення за формулою (1.2)

$$I = \frac{S}{h(A + B)}, \quad (1.2)[15]$$

де S – площа приміщення, $S = 16 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 2,9 \text{ м}$;

A – ширина приміщення, $A = 4 \text{ м}$;

B – довжина приміщення, $B = 4 \text{ м}$.

Підставивши значення отримаємо значення індексу рівне 0,69. Знаючи індекс приміщення I , $\eta = 0,24$.

Підставимо всі значення у формулу для визначення світлового потоку F отримаємо його значення рівне 44000 лм.

Для освітлення використані люмінесцентні лампи типу ЛБ-40-4, світловий потік яких $F = 3000 \text{ лм}$. Розрахуємо необхідну кількість ламп у світильниках за формулою (1.3)

$$N = \frac{F}{F_{\text{л}}}, \quad (1.3)[15]$$

де N – кількість ламп, що визначається;

F - світловий потік, $F = 44000 \text{ лм}$;

$F_{\text{л}}$ - світловий потік лампи, $F_{\text{л}} = 3000$ лм.

Кількість ламп дорівнює 14. На момент виконання роботи у приміщенні було 8 люмінесцентних ламп, тому рівень штучного освітлення не задовольняв санітарним нормам.

Електробезпека виробничих приміщень

Згідно з правилами улаштування електроустановок існує три класи приміщень, що розрізняються за ступенем ризику ураження електричним струмом: приміщення без підвищеної небезпеки, приміщення з підвищеною небезпекою і приміщення особливо небезпечні [14].

Робоче приміщення, в якому пишеться дипломна робота, відноситься до приміщень без підвищеної небезпеки, так як воно не сире (вологість повітря не перевищує 75%), температура не перевищує $+28^{\circ}\text{C}$ (середнє значення $+22^{\circ}\text{C}$), і регулярно проводиться прибирання приміщення, що не дозволяє утворюватися струмопровідній пилюці. Робоча напруга в мережі 220 В. Проводи ізольовані і розташовані таким чином, що ймовірність випадкового контакту людини з проводами значно знижена.

Вплив шуму на працездатність

Шум визначають як сукупність аперіодичних звуків різної інтенсивності і частоти.

Під час облаштування робочого місця працівника з екранними пристроями необхідно обирати таке устаткування, яке не створює зайвого шуму та не виділяє надлишкового тепла. Рівні шуму на робочих місцях осіб, які працюють з екранними пристроями, мають відповідати вимогам Санітарних норм виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99, затверджених постановою Головного державного санітарного лікаря України від 01 грудня 1999 року № 37 [10].

Шуми різняться за різними параметрами, бувають такі, як:

- низько-, середньо-, і високочастотні;
- постійні і непостійні;
- тривалі і короткочасні.

Велике значення надається амплітудно-тимчасовим, спектральним і імовірнісним параметрами непостійних шумів, які характерні для сучасного виробництва. Інтенсивний шум сприяє зниженню працездатності, знижує концентрацію і швидкість роботи, є причиною накопичення втоми [10].

4.3 Розробка заходів щодо зменшення впливу виробничих факторів на працюючого

Закон України «Про охорону праці» визначає основні положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності, на належні, безпечні і здорові умови праці, регулює за участю відповідних органів державної влади відносини між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні.

Роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці [13].

З цією метою роботодавець забезпечує функціонування системи управління охороною праці, а саме:

- створює відповідні служби і призначає посадових осіб, які забезпечують вирішення конкретних питань охорони праці, затверджує інструкції про їх обов'язки, права та відповідальність за виконання покладених на них функцій, а також контролює їх додержання;
- забезпечує виконання необхідних профілактичних заходів відповідно до обставин, що змінюються;

- впроваджує прогресивні технології, досягнення науки і техніки, засоби механізації та автоматизації виробництва, вимоги ергономіки, позитивний досвід з охорони праці тощо;

- забезпечує належне утримання будівель і споруд, виробничого обладнання та устаткування, моніторинг за їх технічним станом;

- забезпечує усунення причин, що призводять до нещасних випадків, професійних захворювань, та здійснення профілактичних заходів, визначених комісіями за підсумками розслідування цих причин.

Роботодавець несе безпосередню відповідальність за порушення зазначених вимог.

На підприємствах, де працює менше 50 працівників, функції служби охорони праці можуть виконувати в порядку сумісництва особи, які мають відповідну підготовку. На підприємствах з кількістю працюючих менше 20 осіб для виконання функції служби охорони праці можуть залучатись сторонні фахівці, які мають відповідну підготовку. Власник з урахуванням специфіки виробництва опрацьовує та затверджує Положення про службу охорони праці підприємства.

На підприємстві з кількістю працюючих 50 і більше осіб рішенням трудового колективу може створюватись комісія з питань охорони праці. Комісія складається з представників власника, профспілки, спеціалістів з охорони праці тощо. Рішення комісії мають дорадчо-рекомендаційний характер. Загальні збори (конференція) трудового колективу затверджують Положення про комісію з питань охорони праці підприємства, яке розробляється за участю сторін. Комісія у своїй діяльності керується законодавством про працю, міжгалузевими і галузевими нормативними актами з охорони праці, Положенням про комісію з питань охорони праці підприємства. Основними завданнями комісії є [13]:

- захист законних прав та інтересів працівників у сфері праці;
- підготовка рекомендацій власнику та працівникам щодо профілактики травматизму, професійних захворювань, практичної реалізації державної політики з охорони праці на підприємстві;

– вироблення пропозицій щодо внесення до колективного договору окремих питань з охорони праці та ін.

Навчання з питань охорони праці – це навчання працівників, учнів, курсантів, студентів, слухачів з метою отримання необхідних знань і навичок з питань охорони праці або безпечного ведення робіт. Організацію навчання та перевірки знань з питань охорони праці працівників під час підготовки, перепідготовки, здобуття нової професії, підвищення кваліфікації на виробництві здійснює служба охорони праці або інші спеціалісти, яким доручена ця робота. У відповідних навчальних програмах має передбачатися теоретичне та практичне навчання. Працівники, зайняті на роботах з підвищеною небезпекою або там, де є потреба у професійному доборі, повинні проходити попереднє спеціальне навчання і не рідше ніж один раз на рік – перевірку знань відповідних нормативних актів про охорону праці.

Особи, яких приймають на роботу, пов'язану з підвищеною пожежною безпекою, повинні попередньо пройти спеціальне навчання (пожежно-технічний мінімум). Працівники зайняті на роботах з підвищеною пожежною небезпекою, один раз на рік проходять перевірку знань відповідних нормативних актів з пожежної безпеки, а посадові особи до початку виконання своїх обов'язків і періодично (один раз на три роки) проходять навчання і перевірку знань з питань пожежної безпеки.

ВИСНОВОК

У дипломній роботі розроблено інформаційну систему мобільної гри «Astro Survival». На початку роботи було проаналізовано предметну область, досліджено процес та структуру створення ігор. Виконано дослідження об'єкта, аналіз інформаційних процесів, проведено формування вимог. Розроблено концепцію мобільної гри, технічний проект, проведено дослідження покращення умов праці та техніки безпеки. Для реалізації системи базуючись на розробленому технічному проекті, було визначено сучасні програмні та технічні засоби, для розробки проекту: об'єктно орієнтована мова програмування C#, ігровий двигун Unity, система управління базами даних phpMyAdmin.

Реалізовано проект мобільної гри «Astro Survival». який має у собі такі функції: запит до СУБД та отримання таблиці лідерів, пересування гравця, знищення ворогів, вибивання предметів, збільшення рівня гравця та отримання могутніх здібностей. Можливість перегляду реклами заради відродження персонажу, обмеження ігрової зони. Це дало змогу створити цікаву, драйвову, динамічну, повну несподіванок – мобільну гру.

СПИСОК ВИКОРИСТОВАНОЇ ЛІТЕРАТУРИ

1. Apple knight: Action Platformer. URL: <https://play.google.com/store/apps/details?id=online.limitless.appleknight.free> (дата звернення 12.03.2022).
2. Caves (rogulike) . URL: <https://play.google.com/store/apps/details?id=thirty.six.dev.underworld> (дата звернення 12.03.2022).
3. Developing 2D Games with Unity: Independent Game Programming with C# Jared Halpern New York, NY, USA (дата звернення 18.03.2022).
4. Game design basics: How to start creating video games. URL: <https://www.cgspectrum.com/blog/game-design-basics-how-to-start-building-video-games> (дата звернення 10.04.2022). – Текст: електронний.
5. Game maker Studio — обзор программы. URL: <https://public-pc.com/game-maker-studio-obzor-programmyi/> (дата звернення 12.03.2022). – Текст: електронний.
6. Pixel dungeoun. URL: <https://play.google.com/store/apps/details?id=com.watabou.pixeldungeon> (дата звернення 12.03.2022).
7. Space Horizon – Космическое выживание вид сверху. URL: <https://play.google.com/store/apps/details?id=com.IndieGamesDev.SpaceHorizonSurvival> (дата звернення 12.03.2022).
8. Годот лучше, чем Unity для разработки игры? Каковы плюсы и минусы? URL: <https://otvetus.com/godot-luchshe-chem-unity-dlya-razrabotki-igri-kakovi-plyusi-i-minusi-31412> (дата звернення 12.03.2022). – Текст: електронний.
9. Движок Unity – особенности, преимущества и недостатки. URL: <https://cubiq.ru/dvizhok-unity/> (дата звернення 12.03.2022). – Текст: електронний.
10. Дослідження впливу шуму на працездатність. URL: <https://library.if.ua/book/86/6047.html> (дата звернення: 09.06.2022). - Текст: електронний.

11. Дослідження мікроклімату у виробничому приміщенні. URL: <https://zakon.rada.gov.ua/rada/show/va042282-99#Text> (дата звернення: 09.06.2022). - Текст: електронний.
12. Огляд нереального двигуна: плюси, мінуси та придатність. URL: <https://www.newgenapps.com/uk/blogs/unreal-engine-review-pros-cons-and-suitability/> (дата звернення 12.03.2022). – Текст: електронний.
13. Охорона праці на підприємстві. URL: <https://te.dsp.gov.ua/ohorona-pratsi-na-pidpryyemstvi-shho-potribno-znaty/> (дата звернення: 09.06.2022). - Текст: електронний.
14. Оцінка показників електробезпеки у виробничому приміщенні. URL: <https://www.sop.com.ua/article/ru/756-elektrobezopasnost> (дата звернення: 09.06.2022). - Текст: електронний.
15. Оцінка якості освітлення. URL: <https://buklib.net/books/35234/> (дата звернення: 09.06.2022). - Текст: електронний.
16. Стоит ли связываться с С#. URL: <https://habr.com/ru/company/vdsina/blog/522934/> (дата звернення 12.03.2022). – Текст: електронний.

ДОДАТОК А – Словник

- Indie¹ – це категорія розробників які створили свій проект самостійно.
- Нодам² – це такі об'єкти, як події, виклики функцій, операції керування потоком, змінні тощо.
- Pool³ - це спосіб оптимізації ваших проектів і зниження навантаження на центральний процесор, коли доводиться швидко створювати та знищувати нові об'єкти. Це хороша практика та модель проектування, яку слід пам'ятати, щоб допомогти розвантажити процесорну потужність центрального процесору для обробки більш важливих завдань і не засипатися повторюваними викликами створення та знищення.
- Skill⁴ – здібність які надає гравцю могутності.
- Pop-up⁵ - це область відображення графічного інтерфейсу користувача (GUI), зазвичай невелике вікно, яке раптово з'являється («спливає») на передньому плані візуального інтерфейсу.
- Particles⁶ - це надійна система ефектів частинок, де можна імітувати рухомі рідини, дим, хмари, полум'я, магічні заклинання та цілу низку інших ефектів.
- VFX⁷ – візуальні ефекти.
- Prefab⁸ – це контейнер для усіх закріплених у ньому об'єктах, скриптів, анімацій, ефектів та ін.
- UI⁹ – інтерфейс програми

ДОДАТОК Б – Технічне завдання

Вступ

Завданням даної дипломної роботи є розробка мобільної гри. В таблиці 1 представлено загальні відомості про систему.

Таблиця 1

Загальні відомості про систему

Повна назва системи	Мобільна гра «Astro Survival»
Умовне позначення системи	Astro Survival
Розроблювач	Маліцький Роман Петрович
Замовник	Кафедра інформаційних управляючих систем та технологій

1. Підстави для розробки

Astro Survival розробляється на підставі завдання для дипломного проектування за спеціальністю 124 – «Системний аналіз».

2. Призначення розробки

Задовільнити потреби геймерів

Основними потребами геймерів є:

- цікава та захоплююча гра;
- можливість здобувати могутні навички;
- змагання за перше місце у таблиці лідерів;
- випробування себе на міцність і ін.

3. Мета створення системи

Основною метою створення Astro Survival є комбінація декількох жанрів ігор та введення незвичайних ігрових механік аби будь-який геймер міг знайти у грі те, що йому подобається, але у інших подібних проектах це не було реалізовано разом.

4. Вимоги до програми

4.1 Вимоги до функціональних характеристик

ІС, що буде спроектована, повинна виконувати наступні завдання:

- здійснювати відображення таблиці лідерів після запиту гравця до СУБД;
- здійснювати запис рекорда гравця після його програшу;
- здійснювати відображення локальної таблиці лідерів;
- коректне функціонування наступних механік:
 - пересування гравця (виконується за допомогою джойстика);
 - поява ворогів (виконується за допомогою особливого скрипта по обробці позицій гравця та можливих точок появи ворогів);
 - знищення ворогів (обробляє тип знищення (лазер, бомба, здібність та ін.));
 - випадіння предметів (залежить від типа знищення та кількості предметів у гравця (відноситься до ракет));
 - підіймання предметів (обробка зіткнення двох колайдерів – предмета та гравця);
 - підвищення рівня;
 - набуття здібностей;
 - активація здібностей;
 - обмеження ігрової зони.

Змінювати параметри гри неможливо після зібрання apk файлу, але в Unity це цілком можна зробити відкривши спеціальний ScriptableObject а саме – GameplayData.

4.2 Вимоги до надійності

Повинні використовуватись передові патерни проектування систем задля швидкості, надійності та, головне, для подальшого розширення проекту та додавання нових функцій.

4.3 Умови експлуатації

Astro Survival розроблено на мові програмування C#, у якості СУБД було використано phpMyAdmin

4.4 Вимоги до інформаційної та програмної сумісності

Мобільний пристрій користувача повинен мати систему Android з версією не нижче шостого.

Вимоги до сервера:

- Процесор з частотою 550 МГц або більш потужний;
- Для підвищення швидкодії рекомендується 2-х процесорна платформа;
- Оперативна пам'ять 1GB, рекомендується 4Gb.

4.5 Вимоги до технічного забезпечення

Технічні вимоги до мобільних пристроїв користувачів визначаються вимогами операційної систем.

5. Вимоги до програмної документації

Усі регламентуючі документи є елементами Astro Survival, які безпосередньо входять до цього. Регламент системи описує правила її ефективного використання. В таблиці 2 наведено склад документів.

Таблиця 2

Склад документів організаційно-методичного та експлуатаційного забезпечення

	Назва	Стадія
	Опис організаційної структури ІПГІА	Технічне проектування
	Керівництво користувача	Реалізація системи

6. Стадії та етапи розробки

В таблиці 3 представлено стадії та етапи по створенню Astro Survival.

Таблиця 3

Стадії та етапи робіт по створенню АРМ СВК

	Стадія	Документи
	Аналіз предметної області	Звіт з описом предметної області
	Системне проектування	Технічне завдання
	Ескізне проектування	Ескізний проект
	Технічне проектування	Технічний проект
	Робоче проектування	Робочий проект

ДОДАТОК В – Текст програми

Базовий клас ворога:

```
using UnityEngine;
using TandC.RunIfYouWantToLive.Common;
using System;

namespace TandC.RunIfYouWantToLive
{
    public abstract class Enemy
    {
        public event Action<Enemy> DestroyEvent;
        public event Action<Enemy> DeathEvent;
        public event Action<GameObject, Enemy> OnCollisionEvent;
        public event Action<Enemy> OnEndEnemyLifeTime;
        public GameObject SelfObject { get => _selfObject; }
        private GameObject _selfObject;
        private Animator _animator;
        protected Rigidbody2D _rigidbody2d;
        private Enemies _enemies;
        private OnBehaviourHandler _onBehaviourHandler;
        private SpriteRenderer _enemySprite;

        protected float _lifeTimer;

        private float _currentHealth;

        public int EnemyId { get; private set; }
        public float Health { get; private set; }
        public int Damage { get; private set; }
        public int GainedExperience { get; private set; }
        public float MovementSpeed { get; set; }
        public Enumerators.EnemyType EnemyType { get; set; }
        private bool _isAlive;
        private bool _wasKilled;
        protected float _attackCooldown = 1f;
        protected float _actualCooldown;
        public Rigidbody2D Rigidbody2D;
        public bool IsCanAction { get; set; }

        public Transform EnemyTransform { get; set; }

        private bool _isInitialize;
        protected Transform _playerTransform;

        private bool _isDamaged;
        private float _blinkTimer;
        private const float _blinkTime = 0.15f;

        private Color _enemyColor;
        public bool IsBoss = false;
    }
}
```

```

private float _frozenStartenTime = 5f;
private float _frozenTimer;
private bool _isFroze = false;

public int DropChance;

public bool IsEndLifeTime = false;

private GameObject _lastTouchedObject;

public Enemy(Transform parent, Enemies data, Transform
playerTransform, Vector2 position, bool isBoss = false, bool init =
true)
{
    if (init)
    {
        _enemies = data;

        EnemyId = _enemies.enemyId;
        MovementSpeed = _enemies.movementSpeed;
        Damage = _enemies.damage;
        _attackCooldown = 1f;
        _playerTransform = playerTransform;
        _selfObject =
MonoBehaviour.Instantiate(_enemies.enemyPrefab, parent);
        _selfObject.transform.position = position;
        _selfObject.name = $"{_enemies.type}";
        _animator =
_selfObject.GetComponent<Animator>();
        _rigidbody2d =
_selfObject.GetComponent<Rigidbody2D>();
        this.Rigidbody2D = _rigidbody2d;
        _onBehaviourHandler =
_selfObject.GetComponent<OnBehaviourHandler>();
        _onBehaviourHandler.Trigger2DEntered +=
OnBehaviorHandler;
        _onBehaviourHandler.Trigger2DStay +=
OnCollisitonStayHandler;
        GainedExperience = data.experience;
        EnemyTransform = _selfObject.transform;
        Health = data.health;
        _enemySprite =
_selfObject.GetComponent<SpriteRenderer>();
        _enemyColor = data.enemyColor;
        _enemySprite.color = _enemyColor;
        _selfObject.transform.localScale =
data.scaleFactor;

        _isAlive = true;
        _isInitialize = true;
        IsCanAction = true;
        IsBoss = isBoss;
        EnemyType = data.type;
    }
}

```

```

        _lifeTimer = data.lifeTime;
    }
    else
        _isInitialize = false;
}
public void FrozeEnemy()
{
    _rigidbody2d.constraints =
RigidbodyConstraints2D.FreezeAll;
    _frozenTimer = _frozenStartenTime;
    _isFroze = true;
}

public abstract void Action();

public void AddLifeTime(float time)
{
    _lifeTimer = time;
}

public void IncreaseParam(float mulriplier)
{
    Health += mulriplier;
    Damage = (int)(Damage * mulriplier);
}

public void Update()
{
    if (IsEndLifeTime)
    {
        //Dispose();
        return;
    }
    if (!_isAlive)
        return;
    if (!_isInitialize)
        return;

    if (!IsCanAction)
    {
        _actualCooldown -= Time.deltaTime;
        if (_actualCooldown <= 0)
        {
            Action();
        }
    }

    if (_isDamaged)
    {
        _blinkTimer -= Time.deltaTime;
        if(_blinkTimer <= 0)
        {
            SetNormalColor();
        }
    }
}

```

```

        _lifeTimer -= Time.deltaTime;
        if (_lifeTimer >= 0)
        {

            return;
        }
        OnEndEnemyLifeTime?.Invoke(this);
    }

    public abstract void Move();

    public void FixedUpdate()
    {
        if (IsEndLifeTime)
            return;
        if (!_isAlive)
            return;
        if (!_isInitialize)
            return;
        if (_isFroze)
        {
            _frozenTimer -= Time.deltaTime;
            if (_frozenTimer <= 0)
            {
                _rigidbody2d.constraints =
RigidbodyConstraints2D.None | RigidbodyConstraints2D.FreezeRotation;
                _isFroze = false;
            }
            return;
        }
        Move();
    }

    public void TakeDamage(float damage)
    {
        if (!_isAlive)
            return;
        if (!_isInitialize)
            return;
        Health -= damage;
        SetDamageColor();
        if (Health > 0)
        {

            _blinkTimer = _blinkTime;
            _isDamaged = true;
        }
        if (Health <= 0)
        {
            Dispose();
            DeathEvent?.Invoke(this);
        }
    }
    private void SetNormalColor()
    {

```

```

        _enemySprite.color = _enemyColor;
    }
    private void SetDamageColor()
    {
        _enemySprite.color = Color.red;
    }

    private void OnCollisitonStayHandler(GameObject
collider)
    {
        //if (collider.tag == _selfObject.tag ||
collider.tag == "Border")
        //{
            //    return;
        //}
        //if(_lastTouchedObject != null)
        //{
            //    if(_lastTouchedObject == collider)
            //    {
                Debug.LogError("Same");
                return;
            //    }
        //}
        //_lastTouchedObject = collider;
        //OnCollderEvent?.Invoke(collider.gameObject,
this);

    }

    private void OnBehaviorHandler(GameObject collider)
    {
        if(collider.tag == _selfObject.tag || collider.tag
== "Border")
        {
            return;
        }
        OnCollderEvent?.Invoke(collider.gameObject, this);
    }

    public void Dispose()
    {
        if (!_isAlive)
            return;
        if (!_isInitialize)
            return;

        _isAlive = false;

        Destroy();
    }

    //public void Disapear(Action callback)
    //{
        //    callback?.Invoke();

```

```

        //}

        protected void PlayAnimation(string name, int layer =
-1, float normalizedTime = 0)
        {
            _animator.Play(name, layer, normalizedTime);
        }

        public void Destroy()
        {
            MonoBehaviour.Destroy(_selfObject);
            _isInitialize = false;
            //DestroyEvent?.Invoke(this);
        }
    }
}

```

Повний код програми представлений за посиланням:
<https://github.com/Balthazariy/AstroSurvival>.

ДОДАТОК Г – Інструкція з використання

Astro Survival - мобільна двовимірна гра з видом зверху, де гравець з'являється на обмеженому ігровому полі та має лише одну мету – протриматися якомога більше часу та заробити максимум очків аби зайняти найвище місце у таблиці лідерів.

Після запуску гри – гравця буде вітати головне меню де є п'ять кнопок:

- Start – запускає гру
- Upgrades – меню прокачки базових характеристик персонажа
- Settings – меню налаштування
- Leaderboard – таблиця лідерів
- Credits – меню інформація про розробників

В меню Upgrades гравець може збільшити свої базові характеристики за монети назавжди. В меню Settings гравець може вимкнути музику, звуки, та, якщо потрібно вимкнути додаткові ігрові ефекти. В меню Leaderboard гравець може побачити таблицю лідерів, яка є двох типів – локальна, та глобальна. Локальна таблиця покаже усі досягнення гравця з пристрою (якщо запустити гру з іншого пристрою, то цих досягнень не буде). А глобальна таблиця лідерів покаже досягнення усіх гравців які коли небудь грали. У меню Credits можна побачити інформацію про розробників – сторінку в інстаграмі, та коротку інформацію про людину.

Сторінка з досягненнями буде відображати усі поточні досягнення гравця, ці досягнення будуть прив'язані до Google акаунту користувача. Після натискання кнопки Start гравець з'явиться на ігровому полі, аби почати двигатися потрібно використовувати лівий стік. Зверху зліва є кнопка паузи на випадок, якщо потрібно зупинити гру на необмежений час у цьому випадку з'явиться меню у якому буде лише дві кнопки – повернення у головне меню, та продовжити гру, друга у свою чергу викмне паузу та гра продовжиться.

Гра розділена на фази – тобто чим більше персонаж у грі «живе» тим складніше стає гра. У грі є вороги які будуть намагатися забрати життя у гравця аби цього не сталося гравець має змогу знищувати їх за допомогою різноманітної зброї та навичок.

У гравця є базова автоматична зброя яка стріляє у напрямку в який повернений гравець. За допомогою неї потрібно знищувати різноманітних ворогів.

Види ворогів:

- базовий(зелений квадрат)
- живучий(повільний фіолетовий шестикутник)
- швидкий(швидка жовта зірка)
- літаюча міна(коло з центральним вирізом – блимає червоним у центрі і чим ближче до гравця тим швидше блимає)
- пила(коло з зубцями пили)
- чорна діра
- троянське коло(розділяється на частинки)

Усі вороги наслідуються від базового, тому мають його параметри, але для деяких ворогів ці параметри переписані.

Параметри базового ворога:

- slow speed
- weak attack
- attack speed – one attack per second
- low health
- no armor

Унікальні навички ворогів:

- літаюча міна при зіткненні з гравцем буде вибухати та наносити гравцю колосальний урон
- пила при зіткненні з гравцем буде наносити урон увесь час доки вона зіткнена з гравцем

- чорна дірка з'являється лише в кінці ігрового поля, повільно пересуваючись вона переслідує гравця, а коли зіткнеться – поверне гравця у центр поля зі зменшеним у половину здоров'ям.
- троянське коло після смерті розділяється спочатку на дві частини, де кожна частина в свою чергу розділяється ще на дві частини після смерті. Чим менша ця частина тем менше в неї здоров'я та тим менше вона завдає пошкодження гравцю.

З кожного вбитого ворога є шанс випадання одного з наступних предметів:

- small experience crystal – надає малу кількість очків підймання рівня
- medium experience crystal – падає рідше за small experience crystal та надає більше очків підймання рівня
- big experience crystal – випадає переважно з босса надає багато очків підймання рівня
- medkit – повне відновлення здоров'я гравця
- coin – дає гравцю монети одного з номіналу: одна, п'ять, десять та двадцять п'ять
- rocket – предмет для здібності Rockets
- frozen mine – міна заморозки. Випадає рідко та при підйманні заморожує ворогів у радіусі дії
- bomb – при підйманні завдає багато пошкодження ворогам у радіусі дії

Коли гравець підіймає один з кристалів він збільшує свої очки підймання рівня і коли кількість очків досягає максимуму – рівень гравця збільшується. З'являється спливаюче вікно в якому гравцю потрібно вибрати одну з трьох унікальних здібностей які вибираються випадково з усього числа здібностей, а потім підтвердити свій вибір. Кожна здібність має короткий опис того, що вона робить. Стати безсмертним неможливо тому що кожна здібність має максимальний рівень. Здібності поділені на два типи – ПАСИВНИЙ та АКТИВНИЙ – пасивні здібності збільшують одну з характеристик гравця на певне числове значення. А активна здібність активується сама або по натисканню кнопки. Для кожної активної здібності є хоча б одна ДОДАТКОВА пасивна здібність.

Типи здібностей та методи їх активації:

- Пасивні здібності:

- Fire rate – збільшує частоту пострілів,
- Damage – збільшує пошкодження гравця,
- Critical hit – збільшує шанс нанесення подвійного пошкодження,
- Critical multiplier – збільшує відсоток подвійного пошкодження базове значення 100%,
- Bullet speed – збільшує швидкість пулі,
- Max health – збільшує максимальну кількість здоров'я гравця,
- Speed – збільшує швидкість пересування гравця,
- Armor – збільшує броню гравця,
- Luck – збільшує вдачу гравця(діє на випадіння предметів та на кількість здібностей зі скриньки яка випадає з боса гри),
- Experience – збільшує кількість отриманих очків підіймання рівня,
- Health regen – повільна регенерація здоров'я,
- Double shoot – подвійний постріл,
- Bullet count – збільшує кількість випущених гравцем пуль.

- Активні здібності:

- Camouflage – маскує гравця дозволяючи йому проходити скрізь ворогів,
- Grenade – кидає гранату у ворогів,
- Auto drone – автоматичний дрон який обертається навколо гравця та атакує ворогів,
- Dash – швидке переміщення гравця у напрямку куди повернений гравець,
- Rockets – запускає ракету у напрямку куди дивиться гравець,
- Laser – вистрілює лазером у напрям куди дивиться гравець та знищує усіх ворогів у радіусі атаки лазера,
- Shield – активний щит дозволяє знешкодити одну атаку ворога,
- Auto gun – автоматична зброя яка вистрілює у будь яку сторону.

- Додаткові здібності:

- Drone gamage – збільшення пошкодження дрону,
- Drone speed – швидкість обертання дрона,
- Drone count – кількість дронів,
- Grenade reload speed – швидкість створення нової гранати,
- Grenade damage – збільшення пошкодження гранати,
- Rocket damage – збільшення пошкодження ракети,
- Rocket count – кількість ракет які гравець може хранити у собі,
- Laser cooldown – швидкість перезаряджання лазера,
- Laser radius – збільшення радіуса лазера.

Завдяки цьому арсеналу здібностей гравець може себе зробити дуже сильним та заробити багато очків. Однак інколи буде з'являтися босс. Це дуже міцний та сильний ворог який має свої особливі навички. Він також буде переслідувати гравця та намагатися його знищити. Після його знищення випаде скринька зі здібностями – це додатковий спосіб отримати одну, три, або п'ять здібностей за раз не підіймаючи свого рівня.

Кількість здібностей які гравець може взяти – обмежено. П'ять пасивних та активних.

Коли гравця торкається ворог його здоров'я зменшується, якщо воно паде нижче/до нуля то гра закінчена – з'явиться відповідне меню яке покаже кількість очків та копійок які заробив гравець та кнопку повернення у головне меню звідки можна почати гру знову, але вже без навичок – з чистого листа.

Усе що було прокачане у меню UPGRADES залишається назавжди.