МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри, канд. техн. наук,
доцент _____ Є. О. Давиденко
«___»____2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

DESKTOP-ГРА ЖАНРУ КЛІКЕР НА БАЗІ UNITY

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.21810924

Студент _____ Д. В. Самковський
«___»_____2022 р.

Керівник канд. техн. наук, доцент _____ Г. В. Горбань
«___»_____2022 р.

Консультант канд. техн. наук, доцент _____ А. О. Алексєєва
«___»_____2022 р.

Миколаїв 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ
Зав. кафедри
_____ Є. О. Давиденко
« __ » _____ 2022 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи бакалавра

Видано студенту групи 409 факультету комп'ютерних наук
Самковському Дмитру Валерійовичу .
(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи**:**

Desktop-гра жанру клікер на базі Unity

Затверджена наказом по ЧНУ ім. П.Могили від « 01 » грудня 2021 р. № 314

2. Строк представлення кваліфікаційної роботи: « __ » _____ 202__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є повноцінна гра на ПК у жанкі клікер з виконаними вимогами та повноцінною реалізацією проєту.

4. Перелік питань, що підлягають розробці:

- аналіз предметної області та існуючих аналогів;
- створення блок-схеми алгоритму роботи модулю;
- розробка програмного забезпечення;
- здійснення тестування роботи програмного забезпечення;
- аналіз результатів розробки.

5. Перелік графічних матеріалів:

презентація.

6. Завдання до спеціальної частини

Дослідження питань охорони праці, які пов’язані з безпекою працівників та студентів

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алєксєєва А.О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи канд. техн. наук, доцент Горбань Гліб Валентинович
(посада, прізвище, ім’я, по батькові)

_____
(підпис)

Завдання прийнято до виконання

Самковський Дмитро Валерійович

(прізвище, ім’я, по батькові студента)

_____
(підпис)

Дата видачі завдання « ___ » _____ 202__ р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема кваліфікаційної роботи**:**

Desktop-гра жанру клікер на базі Unity

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	03.10.2021р.	04.10.2021р	виконано
2.	Огляд літератури за темою роботи	06.10.2021р.	08.10.2021р.	виконано
3.	Складання календарного плану КРБ	09.10.2021р.	10.10.2021р.	
4.	Аналіз предметної області	15.10.2021р.	15.10.2021р.	виконано
5.	Розробка проєктних рішень	03.11.2021р.	04.11.2021р.	виконано
6.	Моделювання та конструювання ПЗ	05.11.2021р.	10.11.2021р.	виконано
7.	Кодування, тестування розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	16.11.2021р.	22.02.2022р.	виконано
8.	Розробка спеціальної частини з охорони праці	14.05.2022р.	28.05.2022р.	виконано
9.	Відгук керівника КРБ	14.06.2022р.	14.06.2022р.	виконано
10.	Оформлення КРБ та презентації	08.02.2022р.	20.05.2022р.	виконано
11.	Попередній захист	20.05.2022р.	08.06.2022р.	виконано
12.	Рецензування	15.06.2022р.	15.06.2022р.	виконано
13.	Завершення оформлення КРБ та презентації	15.06.2022р.	21.06.2022р.	виконано
14.	Захист кваліфікаційної роботи	28.06.2022р.	28.06.2022р.	виконано

Розробив студент Самковський Дмитро Валерійович _____

(прізвище, ім'я, по батькові) *(підпис)*

« ___ » _____ 202__ р.

Керівник роботи канд. техн. наук, доцент Горбань Г. В. _____

(посада, прізвище, ім'я, по батькові) *(підпис)*

« ___ » _____ 202__ р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Desktop-гра жанру клікер на базі Unity»

Студент 409 гр.: Самковський Дмитро Валерійович

Керівник: канд. техн. наук, доцент Горбань Г. В.

Дана робота присвячена розробці гри на базі Unity у жанрах клікер та Tower Defence.

Об'єкт роботи: розробка desktop застосунку гри у жанрі Tower Defence у середовищі розробки Unity.

Предмет роботи: технології та алгоритми ігрового режиму у сучасних ігрових проєктах, реалізованих на ПК.

Мета: Вдосконалення напрямку ігрової індустрії за рахунок розробки гри жанру клікер усуненням недоліків ігор-аналогів, а також розширенням ігрового функціоналу.

Кваліфікаційна робота бакалавра складається з вступу, чотирьох розділів, висновків та додатків.

У вступі визначається актуальність теми, що приймається за мету, а також проводиться формування предмету та об’єкту роботи, огляд поставленої задачі та сфери застосування.

У першому розділі проведено опис аналітичної частини, яка являє собою огляд існуючих відеоігор у жанрах клікер та Tower Defence, проведення аналізу застосунків-аналогів допомагає сформувати функціонал гри, що розроблюється. Визначається основна особливість ігор такого жанру, завдяки чому було сформовано загальне розуміння предметної області. Крім того сформовано та описано специфікації вимог до програмного забезпечення.

У другому розділі описується процес розробки проєктних рішень, включаючи моделювання предмету та об'єкту дослідження, крім того, опис функціональних та інформаційних моделей, та архітекруту програмного застосунку. Описано ігрові сценарії гри, збереження ігрового процессу та отримання винагород.

У третьому розділі описується процес вибору мови програмування C#, середовища розробки VisualStudio та ігрового двигуна Unity. Також частиною розділу є опис виконаної роботи з конструювання програмного забезпечення, включаючи розробку UML-діаграм та опис інтерфейсів гри.

У четвертому розділі продемонстровано проведену роботу з кодування, створення моделей та необхідних матеріалів для розміщення на сценах Desktop-гри, при розробці в Unity, а також процес тестування, крім того описується розробка ігрового меню та налаштувань, які дозволять користувачу як найшвидше зрозуміти та адаптуватися під управління гри, та долучитися до ігрового процесу.

У висновках проводиться аналіз роботи та отриманих результатів.

Кваліфікаційна робота бакалавра викладена на 64 сторінок, вона містить 4 розділи, 54 ілюстрацій, 9 таблиць, 19 джерел в переліку посилань.

Ключові слова: *розробка на Unity, відеогра, Desktop-гра, Tower Defence, гра жанру клікер, розробка на C#, UML-діаграми, гра для Windows, створення асету для гри, гра-клікер*.

ABSTRACT

of the Bachelor’s Thesis

«Unity-based clicker desktop game»

Student: Samkovskyi Dmytro

Supervisor: Candidate of Technical Sciences (Ph. D.), Associate Professor

Horban H. V.

This work is devoted to the development of the game based on Unity in the genres of clicker and Tower Defense.

Object of work: development of a desktop application for the Tower Defense game genre in the Unity development environment.

Subject of work: technologies and algorithms of game mode in modern game projects implemented on PC.

Objective: To improve the direction of the gaming industry by developing a game genre clicker by eliminating the shortcomings of analog games, as well as expanding the game functionality.

The bachelor's thesis consists of an introduction, four chapters, conclusions and appendices.

The introduction determines the relevance of the topic, which is taken as a goal, as well as the formation of the subject and object of research, an overview of the task and scope.

The first section describes the analytical part, which is an overview of existing video games in the genres of clicker and Tower Defense, analysis of analog applications helps to form the functionality of the game being developed. The main feature of games of this genre is determined, thanks to which a general understanding of the subject area was formed. In addition, software requirements specifications have been generated and described.

The second section describes the process of developing design solutions, including modeling of the subject and object of research, in addition, a description of

functional and information models, and the architect of the software application. Game scenarios, saving the gameplay and receiving rewards are described.

The third section describes the process of choosing a C # programming language, VisualStudio development environment, and the Unity game engine. Also part of the section is a description of the work performed on software design, including the development of UML diagrams and a description of game interfaces.

The fourth section demonstrates the work done on coding, creating models and the necessary materials for placement on the scenes of the desktop game, during development in Unity, as well as the testing process, also describes the development of game menus and settings control the game, and join the gameplay.In the conclusions, the analysis of the work and the received results is carried out.

The qualification work of the bachelor is presented on 64 pages, it contains 4 sections, 54 illustrations, 9 tables, 19 sources in the list of references.

Keywords: *development on Unity, video game, Desktop game, Tower Defense, a game of the clicker genre, development on C #, UML-charts, game for Windows, creation of an asset for the game, game-clicker.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних
ОС – операційна система
ПЗ – програмне забезпечення
ПК – персональний комп’ютер
СКБД – Система управління базами даних
ЦА – цільова аудиторія

API – Application programming interface
JS – JavaScript
UML – Unified Modeling Language
US – UnityScript
VS – Visual Studio
WAP – Wireless Application Protocol

ВСТУП

Актуальність теми обумовлена тим, що згідно з даними дослідницької компанії Gartner за 2008 рік, на кожні 5-6 чоловік в світі припадає по 1 персональному комп'ютеру, звісно, що у 2021 році статистика інша, але тенденція демонструє збільшення виробництва та продажу ПК, тож і кількість користувачів також збільшилась [1]. Користувачі ПК надають перевагу не лише роботі за ПК, але й відпочинку. Даний факт демонструє ігрова індустрія, так за даними Вікіпедії про випуски ігр на різні платформи, лише за 2020 рік для ОС Windows було створено більш ніж 300-а ігр різних жанрів. Тож тема Desktop-ігор є актульною та розповсюдженою на сьогоднішній день як для розробників, так і для користувачів.

Таке зростання популярності зрозуміти не важно, адже все завдяки широкому росповсюдженню в мережі Інтернет і на відміну від інших розваг комп'ютерні ігри найбільш доступні. Буквально раз на кілька років з'являються нові жанри ігор: королівська битва, а так само інкрементальні ігри чи просто кажучи клікери.

Один із найпопулярніших жанрів – «клікер». Ігри цього жанру прості, вони не потребують забагато зусиль для проходження рівня або гри в цілому. Вони вже посіли свою нішу на ринку ігор.

Зазвичай ігри мають певні вікові обмеження через контент (сюжет, графіка тощо), які міститься в грі, що в свою чергу зменшує кількість користувачів. Ігри-клікери частіше за все не мають такої проблеми, оскільки процес гри є досить простим, щоб зацікавити користувачів різного віку, а сюжет та візуалізація є досить цікавими, щоб втримати користувачів в ігровому світі якомога довше [2].

Об'єкт роботи: розробка desktop застосунку гри у жанрі Tower Defence у середовищі розробки Unity.

Предмет роботи: технології та алгоритми ігрового процесу у сучасних проєктах, реалізованих на ПК.

Мета: Вдосконалення напрямку ігрової індустрії за рахунок розробки гри жанру клікер усуненням недоліків ігор-аналогів, а також розширенням ігрового функціоналу.

Відповідно для досягнення визначеної мети необхідно вирішити наступні **завдання**:

- аналіз застосунків-аналогів;
- визначення вимог та специфікації до застосунку;
- розробка необхідних анімацій та інструментів для удосконалення;
- розробка ігрового меню та меню налаштувань;
- розробка алгоритмів видачі винагород для кожного гравця;
- створення різних режимів гри з вибором складності;
- реалізація прототипу гри.

Сфера застосування: desktop-гра жанру клікер знаходить застосування в повсякденному житті у сфері розваг та дозвілля, оскільки ігри даного жанру мають змогу відволікати людину від повсякденних турбот, а також легко вписуватися в життєвий графік. Простий гемплей та сенс гри дозволяє збільшити ЦА застосунку та бути цікавим як для підлітків, так і для більш дорослої авдиторії.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд застосунків-аналогів ігор жанру клікер

Важливим етапом розробки є проведення аналізу застосунків-аналогів, оскільки це допомогає виявити недоліки як і аналогу, так і гри, що розробляється, для того, щоб не виправити їх при розробці та підвищити якість гри. Для конкурентоспроможності застосунок повинен мати не лише подібний функционал ігор-аналогів, а також власні, індивідуальні можливості для охоплення більш широкої публіки [3]. Для аналізу аналогів обрано наступні застосунки: KingDom Rush (табл. 1.1), Clash of Clans (табл. 1.2), Plants vs. Zombies (табл. 1.3).

KingDom Rush

Гра унікальна тим, що має велику різноманітність проходження рівнів та гри в цілому (рис. 1.1). Вздовж встановленого шляху гравець може встановлювати перешкоди та покращувати їх. У грі присутні 4 типи веж - магічна, артилерійна, лучниця і казарма. Також у гравець має два заклинання, одне для захисту, інше для атаки.

- велика кількість рівнів;
- багато досягнень;
- гарна візуалізація;

Таблиця 1.1 – Опис «KingDom Rush»

Назва	KingDom Rush
Архітектура	3tier application
Виробник	Ironhide Game Studio
Мова реалізації	C++

Кінець таблиці 1.1

Функції	1. зведення оборони від ворожого нападу; 2. покращення своїх будівель для кращої оборони; 3. пасивне добування ігрової валюти; 4. режим компанії з великою кількістю місій; 5. здобування відзнак за різні завдання; 6. різні рівні складності гри.
Переваги	1. гра повністю безкоштовна.
Недоліки	1. присутня донатна система для легшого проходження гри та доступності декількох нових локацій та багатьох інших вдосконалень для комфортної гри.
Вебсайт	https://www.kingdoomrush.ua/

Рисунок 1.1 – Обкладинка гри «KingDom Rush»

Clash of Clans

Гра у якій потрібно чітко планувати свої дії (рис. 1.2), оскільки гравець повинен не тільки захищати своє селище, а і нападати на інших гравців для швидшого розвитку у грі:

- велика кількість різних ігровіх валют;
- різноманітність вибору кожного гравця;
- декілька локацій для розвитку.

Таблиця 1.2 – Опис 3tier application «Clash of Clans»

Назва	Clash of Clans
Архітектура	3tier web application
Виробник	SuperSell
Мова реалізації	Python, Java
Функції	1. гра с дуже повільним проходженням, цим і цікава для багатьох користувачів; 2. онлайн-режим, для гри с друзями; 3. багато різних завдань як для одного користувача, так для кооперативного проходження; 4. чотири різні ігрові валюти, які використовуються для різних цілей.
Переваги	1. гра повністю залежить від самого гравця, від його рішень та вмінь; 2. можливість грати як з друзями, так і проти них; 3. унікальність кожного гравця, кожен вирішує в якому напряму розвиватись.
Недоліки	1. присутня донатна система.
Вебсайт	https://supersell.com/

Рисунок 1.2 – Обкладинка гри «Clash of Clans»

Plants vs. Zombies

Гра була розроблена Windows (рис. 1.3), але згодом вона була портована на приставки, портативні ігрові системи та мобільні пристрої.

Рисунок 1.3 – Обкладинка гри «Plants vs. Zombies»

Таблиця 1.3 – Опис 3tier application «Plants vs. Zombies»

Назва	Plants vs. Zombies
Архітектура	3tier application
Виробник	PopCap Games
Мова реалізації	C#, C++
Функції	1. гра повністю безкоштовна; 2. процес гри продовжується у режимі офлайн; 3. гра підходить для розвивання розумових здібностей.
Переваги	1. розробники створили гру повністю безкоштовну, та з можливістью оффлайн-використання; 2. гра не потребує особих здібностей, все швидко пузнається у процесі гри.
Недоліки	1. гра має не велику кількість рівнів; 2. складність рівнів на всіх рівнях однакова.
Вебсайт	https:// popcapgames.com/

1.2 Аналіз системи, що розробляється

Гра «Defense of the castle» у жанрі Tower Defense призначена для розвитку логічного мислення. Користувачу на різних рівнях доводиться приймати різні рішення, гра не здається одноманітною на кожному рівні. Також у грі присутні позарівневі покращення, які так само кожен гравець вибирає сам для себе.

Гра має бути сумісною з ОС Windows 10 (або версії вище). Система повинна бути в працездатному стані 24 годин на добу, 7 днів на тиждень для підтримки сервісів гри, за вийнятком технічних робіт [4].

Таблиця 1.4 – Опис системи, що розробляється

Призначення ПЗ	Користувач на сам перед повинен отримувати задоволення від гри, не зважаючи на вік гравця. Гра повинна бути простою в багатьох аспектах, та дуже різноманітную в плані вибору для кожного гравця. Користувач має буди зацікавленний у здобутті нових досягнень та отриманні нагород, від складності завдання залежить кількість нагород. Гра повинна забезпечувати багатокористувацький режим. Підтримка з Windows 7 та вище. Гра повинна підтримувати офлайн режим та бути доступною в будь який час.
Функції	1. реєстрація гравця; 2. проходження ігрового навчання; 3. простий вступ до гри; 4. перегляд прогресу та здобуття нагород; 5. глобальний чат з іншими гравцями; 6. календар подій.
Користувачі	1. програміст; 2. тестувальник; 3. користувач.
Сценарії роботи	1. Користувач встановлює гру. 2. Користувач вибирає завдання. 3. Користувач обирає спосіб розвитку у грі. 4. Користувач виконує завдання, яке він обрав. 5. Користувач відмічає прогрес. 6. Користувач отримує нові завдання та відкриває нові досягнення. 7. Тестувальник слідкує за тим, щоб не було помилок при отриманні завдань та отримані нагород за прогресс у грі.

Кінець таблиці 1.4

Сценарії роботи	8. Програміст додає нові завдання, події та розроблює велику кількість нововведень. 9. Користувач продовжує грати та здобуває нові досягнення.

1.3 Загальний алгоритм реалізації проєкту

Для розробки гри в першу чергу потрібно визначитись з осбягом работи та тривалістю. Зазвичай тривалість розробки гри полягає у складності самого проєкту. На розробку великого проєкту знадобляться декілька спеціалістів у різних сферах.

Розробка гри проходить у декілька етапів:

- проєктування меню;
- проєктування рівнів;
- моделювання;
- розробка;
- тестування.

На основі питань які виникають під час підготовки до виробництва створюється GDD (документ ігрового дизайну), у себе він включає:

- жанр;
- ідея;
- сюжет;
- персонажі;
- рівні та складність;
- геймплей.

Даний список може доповнюватися під час розробки гри. Усе це потрібно для того, щоб у процессі розробки було меньше питань при розробці гри. Та для того, щоб в кінці вийшов повноцінний проєкт, який вже менше потрібно буде до

працьовувати. Але гра вийде в бета-версії, для того, щоб можно було виправити деякі деталі, які могли пропустити розробники під час створення проєкту.

1.4 Специфікація вимог до програмного забезпечення «Defense of the castle»

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення системи (застосунку), для якої розробляється програмне забезпечення

Призначенням застосунку є застосування в повсякденному житті у сфері розваг та дозвілля.

Погодження, що ухвалені в програмній документації

Створення загального ПЗ та його злагодженої роботи буде використовувати допоміжні ассети та бібліотеки Unity.

Межі проєкту ПЗ

Крайня дата завершення роботи над ПЗ – 20.06.2022 р.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Застосунок націлений на використання користувачем для проведення дозвілу та відпочинку.

Характеристики користувачів

Основні характеристики користувачів: наявність ПК та доступу до мережі Інтернет.

Загальна структура і склад системи

Основні частини програмного забезпечення: сервер, БД.

Загальні обмеження

Єдиним обмеженням є наявність доступу до мережі Інтернет.

ФУНКЦІЇ СИСТЕМИ DESKTOP-ГРИ ЖАНРУ КЛІКЕР

Функція отримання досягнення

Опис функції

Функція отримання досягнення допомогає гравцю отримувати більше задоволення від гри.

Вхідна і вихідна інформація

Вхідна інформація – проходженння рівня або локації, вихідна інформація – отримання досягнення у профіль гравця.

Функціональні вимоги

Бази даних з досягненнями, які виконані повністю або не до кінця.

Функція надання персонажу імені

Опис функції

Функція привласнення гравцем імені персонажу.

Вхідна і вихідна інформація

Вхідна інформація – ім’я персонажу, вихідна інформація – відображення імені персонажу в налаштуванні та під час ігрового процессу.

Функціональні вимоги

База даних зі збереженням імені для персонажу кожного гравця.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації (даних)

В даному ПЗ основним джерелом вхідної інформації є користувач. Який самостійно вводить данні, такі як отримання досягненнь та додавання ім’я персонажу.

Нормативно-довідкова інформація (класифікатори, довідники тощо)

Вимоги відсутні.

Вимоги до способів організації, збереження та ведення інформації

В якості БД для збереження та ведення інформації в грі обрано реляційну бауз даних – SQLite. Основною перевагою якої є мультиплатформність, крім того дану БД прийнято використовувати при необхідності зберегти компактність та забезпечити швидку взаємодію з даними.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Вимоги до технічного забезпечення: комп’ютер чи ноутбук з оперативною

пам’яттю не меньше 8 Гб та піддримкою програмного забезпечення Unity.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Архітектура програмної системи складається з трьох компонентів: клієнтської частини, серверної частини та БД.

Системне програмне забезпечення

Гра розроблена на платформі Unity з використанням мови програмування C#. В якості БД для застосунку обрано SQLite.

Мережне програмне забезпечення

Для створення програмного забезпечання було використано ОС Windows 10.

Програмне забезпечення ведення інформаційної бази

Через SQLite має відбуватися керування, ведення інформаційною базою застосунку.

Мова і технологія розробки ПЗ

Розробка гри відбувалась на платформі Unity з використанням мови програмування C#.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтерфейс має задовольняти усі вимоги дизайну, він є доволі зручним у використанні .

Апаратний інтерфейс

Апаратним інтерфейсом є ПК користувача, з операційною системою Windows 10.

Програмний інтерфейс

У ході розробки було використано дві категоріх Unity Scripting API : UnityEditor (Animations, Events тощо) та UnityEngine (Analytics, Audio тощо).

Комунікаційний протокол

Гра передбачає використання мережних протоколів WAP — протокол

безпровідної передачі даних та TCP/IP.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Гра є доступною для будь-якого користувача, за умови наявності у користувача апаратного інтерфейсу та доступу до мережі Інтернет.

Супроводжуваність

Гра не потребує супроводжуваності.

Переносимість

Програмне забезпечення може працювати на ОС Windows (10 версія та вище).

Продуктивність

Продуктивність роботи ПЗ залежить від швидкості підключення до мережі Інтернет.

Надійність

Дані надані користувачем при реєстрації є приватними, у кожного користувача буде власний аккаунт з власним ігровим процесом.

Безпека

Кожен користувач авторизується через свій аккаунт.

ІНШІ ВИМОГИ

Усі вимоги сформовано та описано вище, доповнення не вимагається.

Висновки до розділу 1

Під час формування розділу закріплено знання та навички стосовно розробки Desktop-ігор на базі Unity та написання специфікації вимог до програмного забезпечення. Таким чином, у першому розділі кваліфікаційної роботи бакалавра проаналізовано предметну область застосунку, завдяки чому визначено актуальність роботи, в основі якої лежить теза, що комп’ютерні ігри мають вплив на розвиток, а також поліпшення функціонування мозку, через що мають змогу покращувати та закріплювати здібності щодо просторового

мислення, крім того й візуального уявлення стосовно тривимірних об'єктів.

Крім того досліджено поняття комп’ютерних ігор, а також проаналізовано їх класифікації та жанри. Особливої уваги приділено Desktop-іграм жанрів клікер й Tower Defence. Закріплено теоретичні знання стосовно засобів створення Desktop-ігор: платформа Unity та об’єктно орієнтована мова програмування C# [5]. Також проведено аналіз існуючих ігор-аналогів. Проаналізовано основні переваги та недоліки, загальний функціонал та інтерфейс розглянутих застосунків-аналогів, завдяки чому проведенно аналіз гри що розробляється, під час якого сформовано тему гри, також її функціонал, основних користувачів й створено сценарій використання.

В розділі, окрім опису Desktop-гри, визначено специфікацію вимог до ПЗ, що розробляється. В данному підрозділі продемонстровано повний опис поведінки гри, включно з множиною функціональних та нефункціональних вимог.

2 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ГРИ

У проєктуванні обрано мову графічного опису для моделювання об’єктів UML. Він створення для візуалізації та проєктування програмних систем. За допомогою UML-моделей можливо зробити генерацію коду, але UML не є мовою програмування [6].

Для побудови діаграм було використано StarUML. За допомогою діаграм можливо краще розуміти структуру проєкту.

2.1 Написання usecase

Короткий usecase

Користувач має виконувати завдання для проходженння гри та отримання наступних завдань та бонусів. Для отримання інших завдань необхідно завершити попередні. Бонуси видаються за часте находження в грі, іх кількість збільшуєься від проведеного часу в грі.

Поверхневий usecase

Користувач мая виконувати завдання для проходженння гри та отримання наступних завдань та бонусів. Для отримання інших завдань необхідно завершити попередні. Бонуси видаються за часте находження в грі, іх кількість збільшуєься від проведеного часу в грі [7]. Додаткові бонуси можуть бути у різні святкові події.

Альтернативний сценарій:

1) користувач не встигає виконати завдання своєчасно, та втрачає певну кількість нагород;

2) користувач довго не заходив в гру, його пасивний дохід переповнений, від рівня кожного гравця залежить кількість пасивного доходу;

3) користувач закінчує завдання завчасно, через що досягнення можуть бути отримані автоматично.

Таблиця 2.1 – Повний usecase

Primary Actor	Користувач
Scope	Desktop-гра клікер на базі Unity
Level	Мета користувача
Preconditions	Користувач авторизований в застосунку
Stakeholders and interests	1. програміст: зацікавлений у розробці своєї задачі; 2. тестувальник: зациікавлений у знаходженні помилок у грі, програміст: для дороблення; 3. користувач: зацікавлений в здобутті досягнень та розвитку за найкоротший час.
Main Success Scenario: 1. Користувач встановлює гру. 2. Користувач вибирає завдання. 3. Користувач обирає спосіб розвитку у грі 4. Користувач виконує завдання яке він обрав 5. Користувач відмічає прогрес. 6. Користувач отримує нові завдання та відкриває нові досягнення. 7. Тестувальник слідкує за тим щоб не було помилок при отриманні завдань та отримані нагород за прогресс у грі. 8. Програміст додає нові завдання, івенти та розроблює велику кількість нововведень. 9. Користувач продовжує грати та здобуває нові досягнення.	
Result	Користувач

Кінець таблиці 2.1

Extensions:	
1.	1. Користувач не заходить в гру: 1.1.Користувач втрачає прогресс завдання. 1.2.a. Користувая втречає здобуття бонусів за довгу відсутність в грі. 1.2.b. Користувач втрачає велику кількість ігрової валюти. 1.3.Користувач мусить проходити заново незавершене завдання. 1.4.Користувач дотримується порад для відновленя завдань. 1.5.Користувач відмічає прогрес.
2	2. Тестувльник знаходить помилку у грі: 2.1a. Помилка не глобальна. 2.2a. Програміст її швидко виправляє. 2.1b. Помилка глобальна 2.2b. Потрібно знаходити та вирішувати помилку та зупиняти гру на технічне обслуговування.
Special Requirements:	1) система перекладу тексту на декілька мов (українську, англійську); 2) адаптивний інтерфейс; 3) повне підключення до інтернету.
Frequency of Occurrence	Гра працює в офлайн режимі, але щоб зайти до неї потрібно підключення до інтернету.

Діаграми варіантів використання описують взаємовідносини та залежності між групами варіантів використання та дійових осіб, що беруть участь у процесі. Важливо розуміти, що діаграми варіантів використання не призначені для

відображення проєкту та не можуть описувати внутрішній пристрій системи. Діаграми варіантів використання призначені для спрощення взаємодії з майбутніми користувачами системи [8], з клієнтами, і особливо знадобляться визначення необхідних характеристик системи. Іншими словами, діаграми варіантів використання говорять про те, що система повинна робити, не вказуючи самі методи, що застосовуються (рис. 2.1).

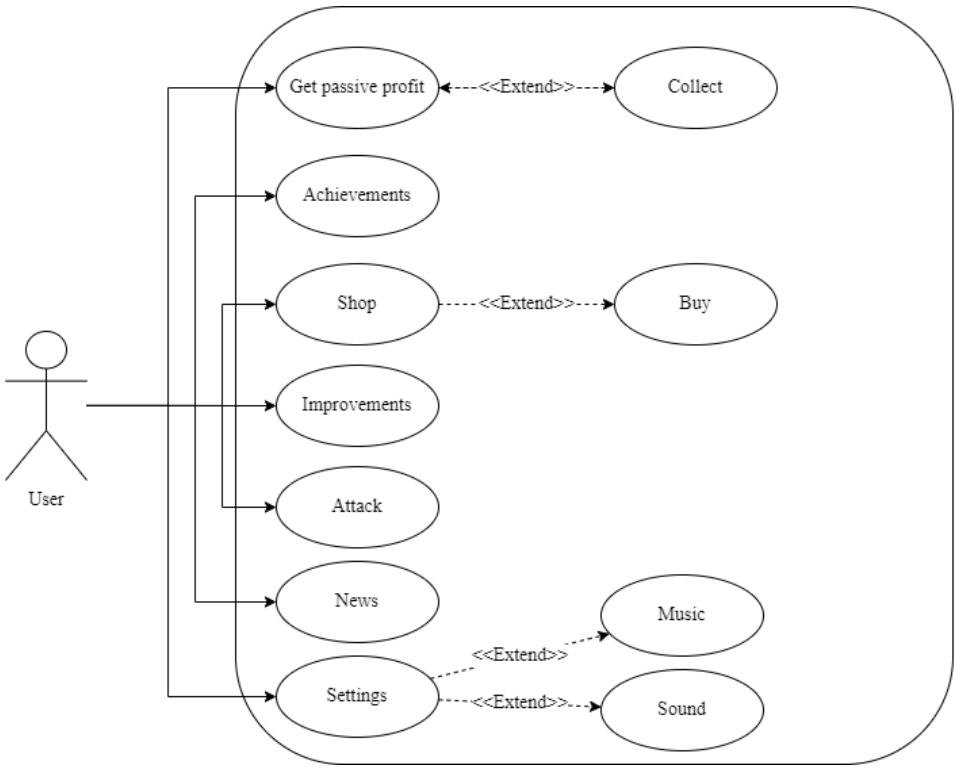

Рисунок 2.1 – Діаграма варіантів використання

Опис елементів моделі на діаграмах варіантів використання:

Актори – це прості користувачі, які взаємодіють з системою. Користувач може бути тільки людиною.

Підсистеми – у моделях UML підсистеми існують як тип стереотипних компонентів, поведінкові одиниці в системі [9]. Підсистеми використовуються в діаграмах класів, компонентів і варіантів використання для представлення великомасштабних компонентів у системі, яку ви моделюєте.

Зв'язки та відношення – зв’язки між елементами моделі. Відношення – тип елемента моделі, який додає семантику до моделі, допомогая визначити структуру та поведінку між елементами моделі.

Випадки використання – описує функцію, яку виконує система для досягнення мети користувача. Випадок використання повинен давати спостережуваний результат, який є цінним для користувача системи.

2.2 Побудова та використання діаграм взаємодії

Під час аналізу системи потрібно розуміти як користувач буде зберігати досягнення. Перша діаграма (рис. 2.2) відповідає за збереження гри та рекорду поставленого в останній раз. Рекорд у грі буде оновлюватись кожен раз, коли значення рекорду буде вище за минулий, таким чинном буде збережений найкраший результат.

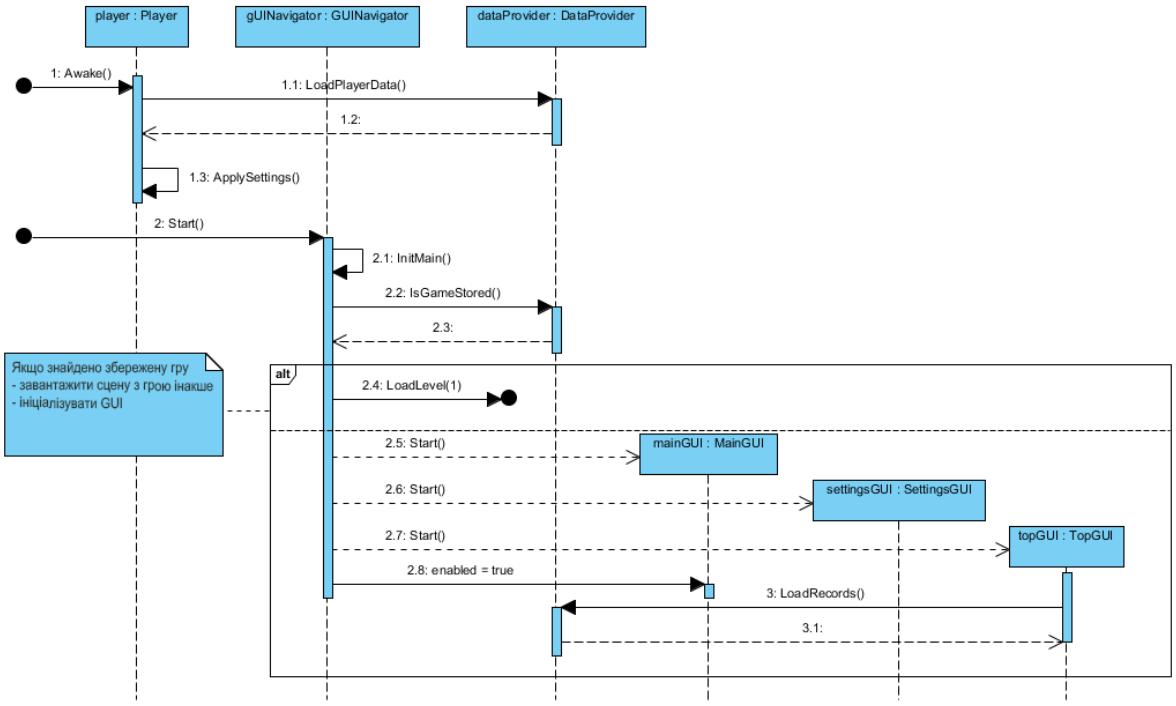

Рисунок 2.2 – Діаграма взаємодії зарахування рекорду

Друга діаграма (рис. 2.3) відповідає за збереження останього процесу в грі, вона автоматично ставиться на паузу для того, щоб при наступному вході до гри прогрес був збережений. Також для паузу під час знаходження в грі.

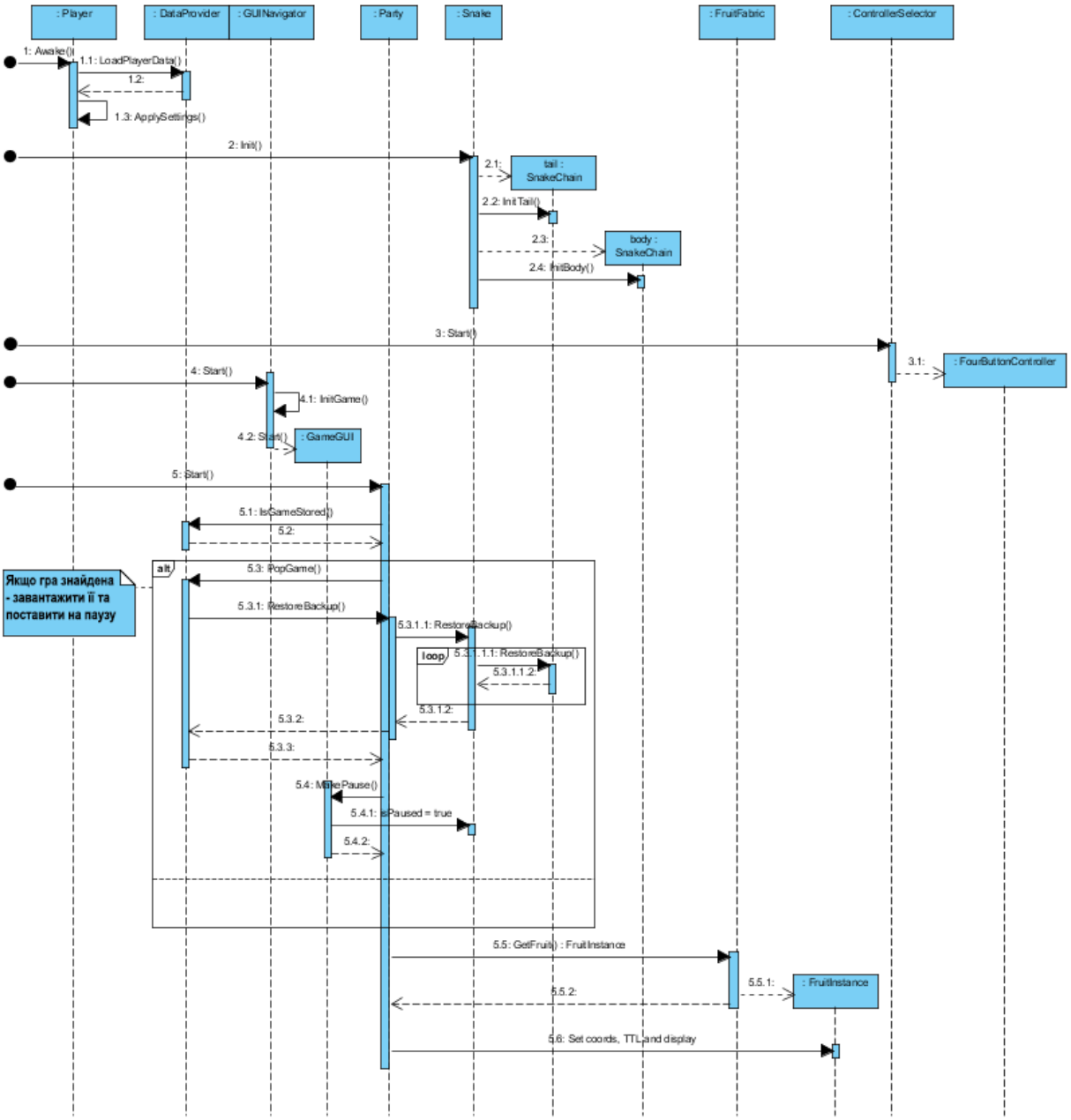

Рисунок 2.3 – Діаграма взаємодії збереження ігрового процесу

2.3 Алгоритм роботи програмного забезпечення

Activity diagrams діяльності є дуже схожою з аналогом блок-схем, вони відображають послідовність дій, яка виконується під час процесу реалізації якогось варіанту використання чи в цілому функціонування системи [10]. Відображають орієнтований граф, де вершини це дії, а ребра – це переходи між ними.

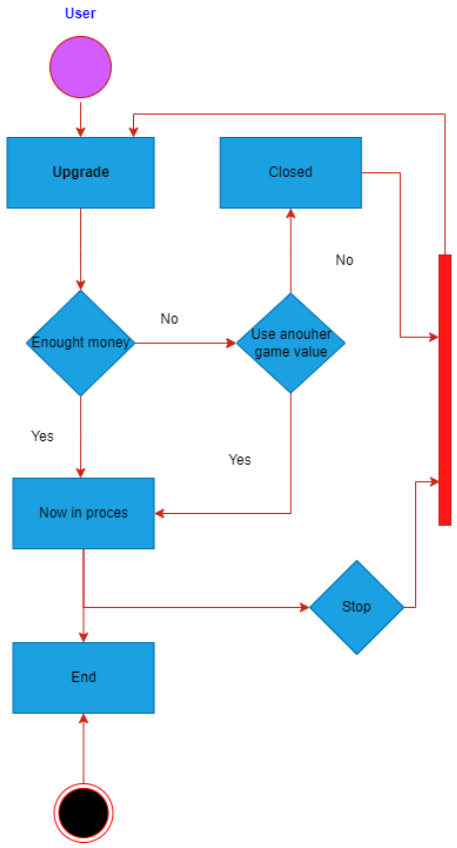

Рисунок 2.4 – Діаграма діяльності покращення

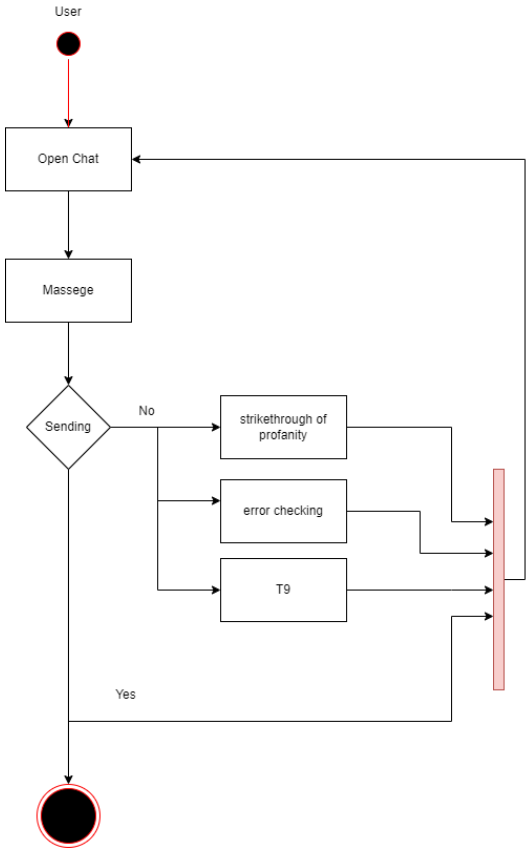

Рисунок 2.5 – Діаграма діяльності чату

Створено дві діаграми діяльності:

1) Покращення (рис. 2.4). Діаграма перевіряє на наявність ігрової валюти для оновлення та вибір за допомогою якої валюти зробити оновлення, також припинення всього процесу оновлення.

2) Додавання прогресу (рис. 2.5). Діаграма перевіряє на наявність помилок в листі для ігрового чату та відправку листа. Після відправлення листа, відкривається чат для написання нового листа.

2.4 Розробка діаграми розгортання

Діаграма розгортання — діаграма в UML (рис. 2.6), на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об’єкти, що виконуються на цих вузлах [11]. Компоненти відповідають представленню робочих екземплярів одиниць коду.

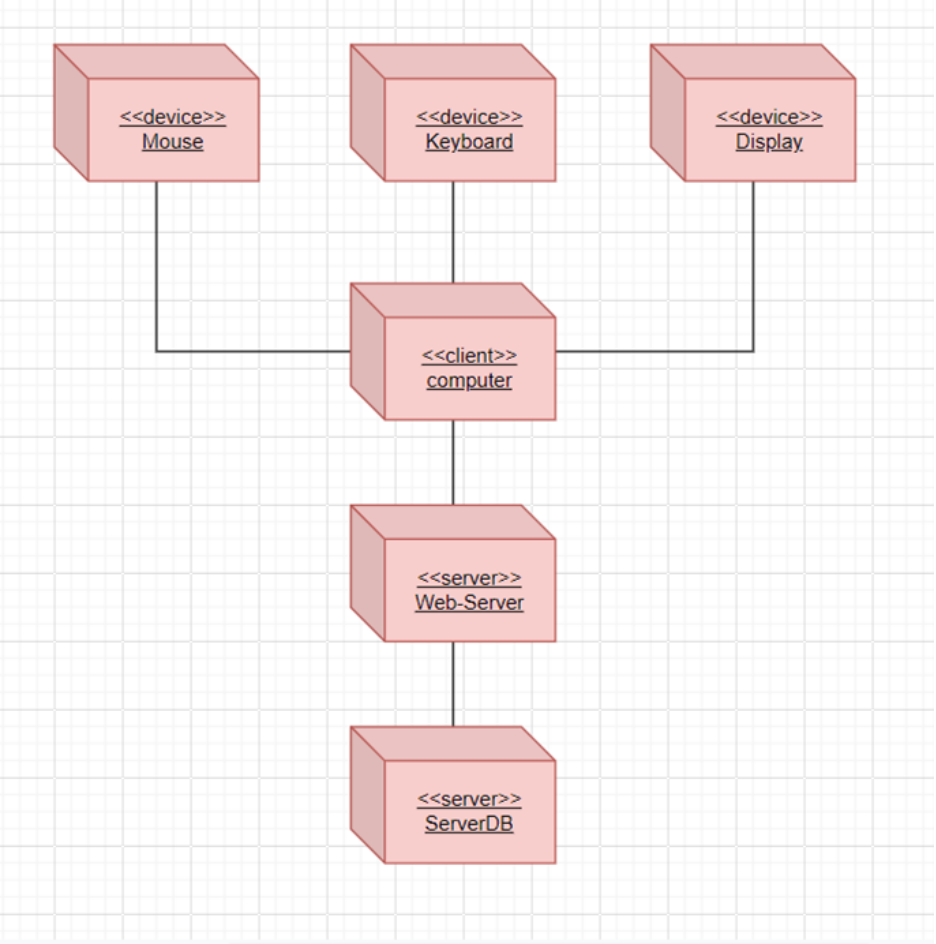

Рисунок 2.6 – Діаграма розгортання для Desktop-гри жанку клікер

Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонентів [12]. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв’язки між типами компонентів.

Висновки до розділу 2

У другому розділі описано архітектуру програмного застосунку, а також проєктування та моделювання. У розділі описано систему у вигляді діаграм:

1. діаграма варіантів;
2. дві діаграми взаємодії;
3. дві діаграми діяльності;
4. діаграма розгортання.

Описано коротку, поверхневу та повну форму usecase, де описано ігрові сценарії. У поверхневому usecase описано успішний та альтернативний сценарій гри. У діаграмах взаємодії описано процес збереження гри та ігрового рекорду, та процес збереження ігрового процесу після виходу з гри для подальшьго проходження. У діаграмах діяльності описано процес покращення у грі з вожливістю відмінити покращення, у разі якщо гравець допустив помилку у ході гри.

3 ПРОЄКТУВАННЯ ІГРОВОГО ЗАСТОСУНКУ ТА ВИБІР ЗАСОБІВ РОЗРОБКИ

3.1 Розробка UML-діаграм

UML-діаграми відіграють важливу роль у створенні проєкту та розробці програмного забезпечення. Мова UML допомогає знайти помилки в структурі програми [13]. Це дуже розповсюджена мова, що вважається найкращою для розробників, власників компаній, підприємців із різних галузей, фахівців та менеджерів проєктів.

3.1.1 Діаграма класів

Діаграма класів – статичне представлення структури моделі, що відображує статичні елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів частіше за все містить позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об’єктно-орієнтованого програмування [14]. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відношення. Створена діаграма (рис. 3.1) має наступні особливості:

1. Діаграма класів відображає лише взаємодію між користувацькими класами, що містять логіку застосунку. Стандартні класи фреймворку не було додано на діаграму.
2. Діаграма єдина для всіх класів, тобто вони всі пов’язані між собою, тому не варто розривати зв’язки між ними.
3. Певні типи зв’язків не представлені на діаграмі, оскільки їм важко знайти логічне застосування в межах цього застосунку.

Кожен з класів має власну назву, яка має пояснювати, за що відповідає клас, а також методи та параметри, що входять до цього класу.

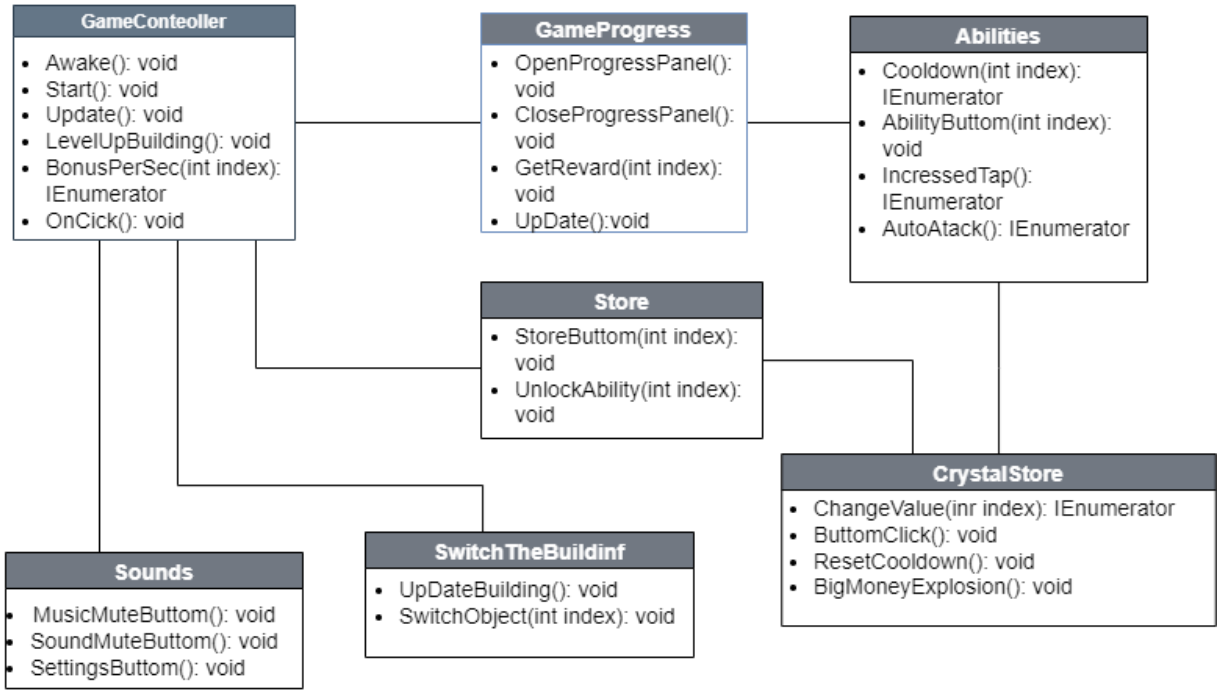

Рисунок 3.1 – Діаграма класів

Опис класів:

Клас **GameController** – відповідає за основні дії програми. В ньому вираховується пасивний дохід гравця, обробляється додавання монет за натискання на ігрове поле, а також панель покращень будівель та головного героя [15]. У цьому класі здійснюється збереження під час закриття програми, а також завантаження під час пробудження. Під час пробудження вираховується пасивний дохід гравця за час відсутності та час відновлення здібностей.

Клас **Store** – описує роботу з покупкою ігрових покращень та здібностей.

Клас **Abilities** – відповідає за роботу із здібностями. У ньому описуються всі події, вироблені здатністю і спрацьовування анімації, і навіть тут обчислюється час відновлення здібностей.

Клас **CrystalStore** – відповідає за покупку покращень та одноразових здібностей за кристали.

Клас **SwitchTheBuilding** – відповідає за відображення та перемикання будівель на екрані.

Клас **Sounds** – відповідає за включення та відключення звуків та музики в ігровій програмі, а також відтворення звуків у грі.

Клас **Progress** – відповідає за видачу нагород за виконання певних умов гравцем.

3.1.2 Діаграма станів та переходів

Діаграми станів демонструють сценарії, які будуть виконуватись під час гри.

Таким чином було *розроблено 2 діаграми станів:*

1) діаграма основного меню для гри (рис. 3.2);
2) діаграма для виконання завдань (рис. 3.3).

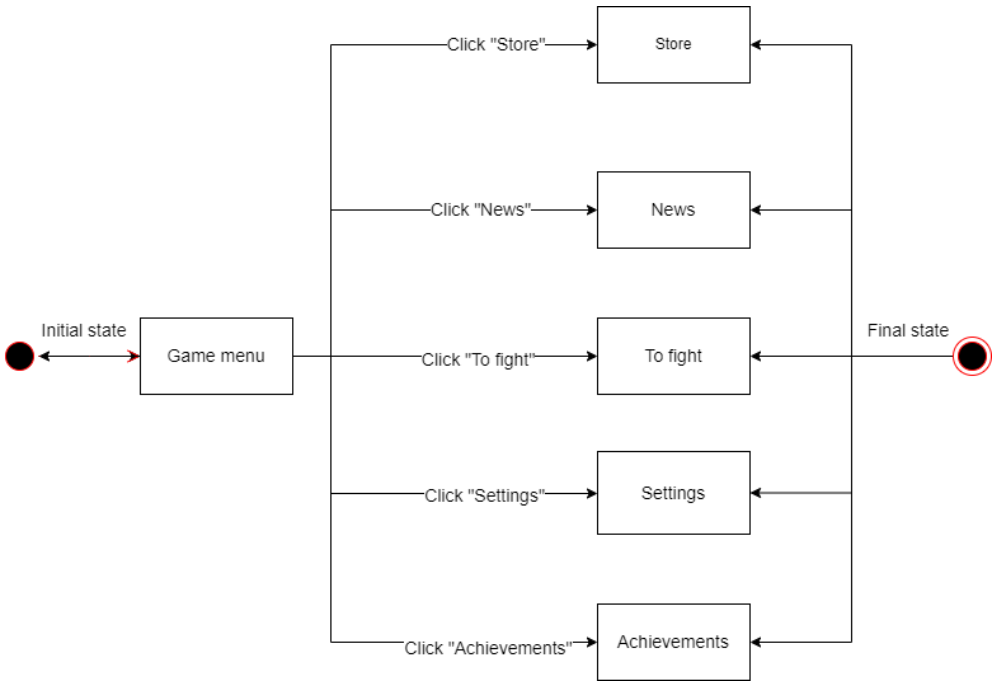

Рисунок 3.2 – Діаграма станів для всієї гри

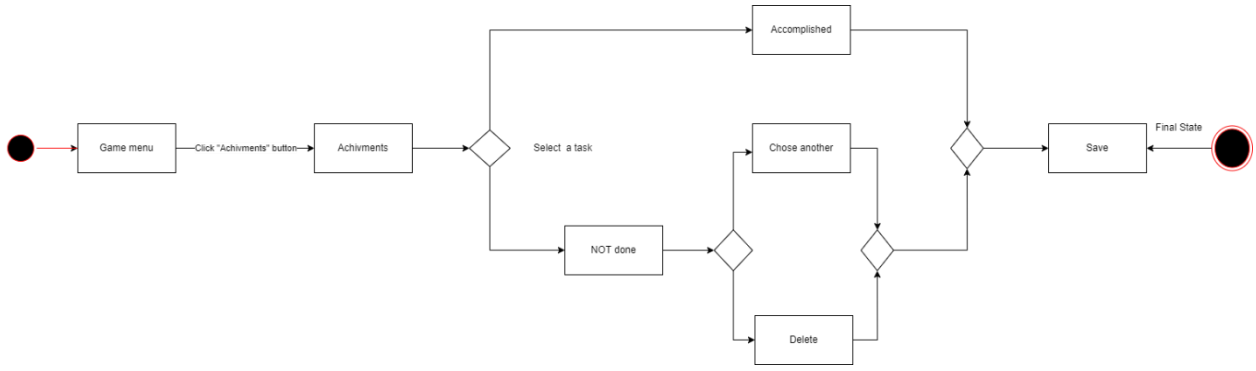

Рисунок 3.3 – Діаграма станів для виконання досягнень

В даних діаграмах описано основні можливості в грі, зі стартового меню є перехід майже до всіх функцій гри такі як: Store, News, To fight, Settings, Achivements. Також опис виконання дисягнегь, де можливо завершити виконане завдання або відмінити ще не виконане завдання для початку нового.

3.1.3 Діаграма пакетів.

Діаграма пакетів (рис. 3.4) необхідна для елементів у групі з метою спрощення структури та організації роботи з модельною організацією системи [16]. Пакет – це інструмент групування, який дозволяє взяти будь-яку конструкцію UML і об’єднати її елементи в одиницях високого рівня.

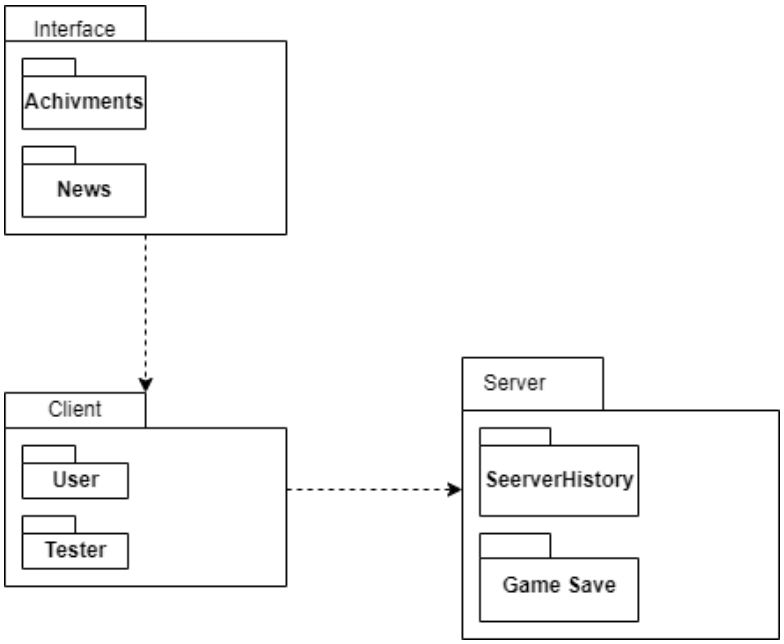

Рисунок 3.4 – Діаграма пакетів

Пакет «Interface» має 2 класи: Achivments та News. Клас пов’язаний відношенням направленої асоціації з компонентом «Client» (тобто залежить від нього), тому використано відношення залежності – штриховану стрілку.

Пакет «Client» містить класи User, Tester які приймають данні та інформацію, про учасників. Дані беруться с пакету «Server».

Пакет «Server» має стереотип database, та класи Game Save, SerserHistory, які містять всі необхідні дані.

3.2 Вибір засобів розробки ігрового застосунку

Перед розробкою програмного забезпечення проаналізовано та порівняно декілька середовища розробки для створення ігор. Вибір платформи важливий, від нього залежить якість ПЗ [17]. Підібрано найбільш відповідні інструменти, які використовуються для розробок ігор.

3.2.1 Засоби створення комп’ютерних ігор

Unity

Unity – найпоширеніша платформа у світі, використовується для створення та управління 3D-вмістом у режимі реального часу [18]. Велика кількість популярних ігор були розроблені на Unity, такі як: Hearthstone: Heroes of Warcraft, Cities Skylines, Cuphead, Pokemon Go (рис. 3.5-3.6).

Рисунок 3.5 – Логотип «Unity»

Unity демонструє гарні показники у абсолютно різних гейм-продуктів. Різноманітність жанрів такі як: пригодницькі ігри, пісочниці, велика кількість симуляторів, в яких велику роль грає фізика, карткові ігри та рольові стратегії. Також деякі з ігор підтримують багатофункціональне використання, можливо використовувати смартфон, ПК, консолі або пристрої віртуальної реальності.

Рисунок 3.6 – Інтерфейс платформи «Unity»

Unreal Engine

Unreal Engine – розроблений компанією Epic Games у 1988 році. Був створений для шутерів від першої особи. На сьогодні використовується для створення ігор у різних жанрах, найпоширеніші: MMORPG, Stealth та багатьох RPG іграх [19]. Для розробки у середовищі використовують C++ (рис. 3.7-3.8). Unreal Engine створює ігри для різних платформ починаючи від Windows закінчуючи PSP.

Рисунок 3.7 – Логотип «Unreal Engine»

Переваги Unreal Engine: Якість графіки, постійно оновлювання інтерфейсу користувача новими інструментами, мова програмування C++.

Велика кількість розробників використовують платформу Unreal Engine, вона проста у кодуванні і використовує вузли Blueprints, які допомагають розробникам створювати ігри високої якості без кодування та написання сценаріїв.

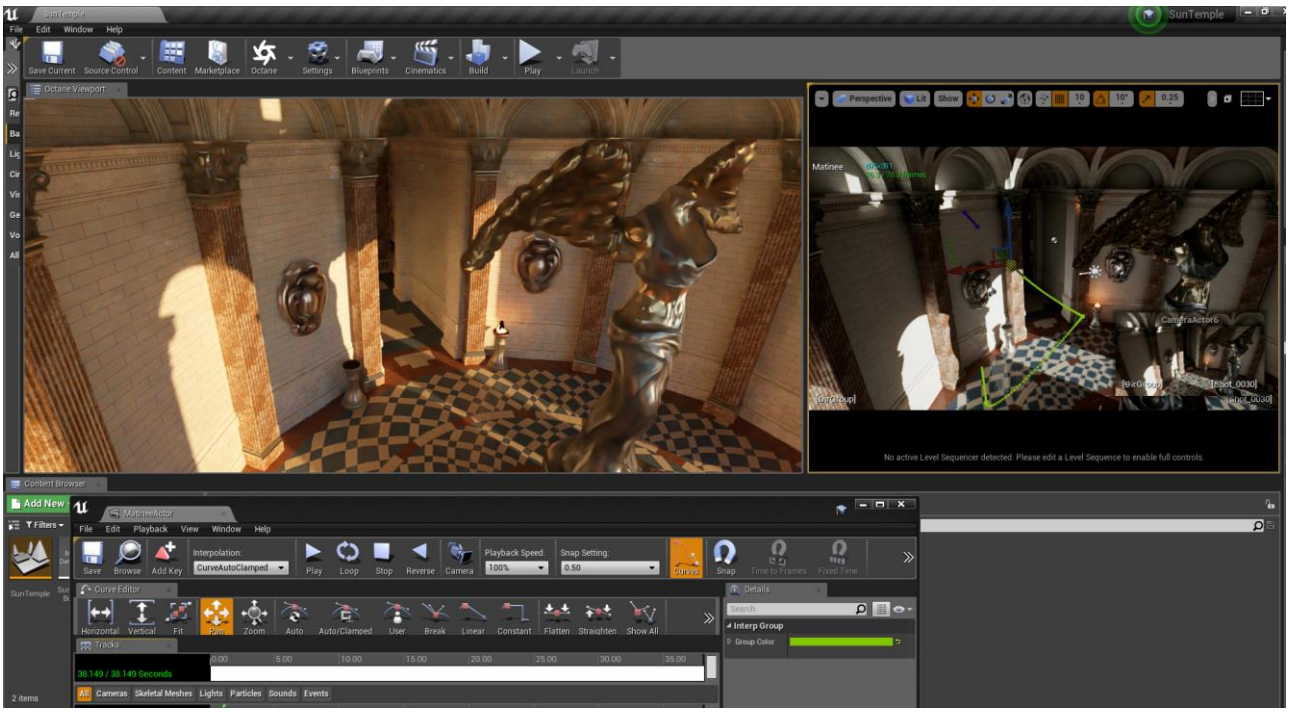

Рисунок 3.8 – Інтерфейс платформи «Unreal Engine»

CryEngine

CryEngine – Як і Unreal Engine був створений для шутерів від першої особи. Став дуже відомим з виходом своєї першої гри Far Cry, гра яка і на сьогодні захоплює серця мільйонів гравців своєю графікою та неймовірним сюжетом. Платформа CryEngine створює ігри для ПК та консолей (рис. 3.9), (рис. 3.10). З розвитком програми компанія почала випускати нові шутери, популярним шутером після Far Cry став Crysis який був створений на другій версії платформи.

Рисунок 3.9 – Логотип «CryEngine»

Але платформа доволі складна у засвоєнні, початковому розробнику потрібно багато часу, щоб оволодіти нею та почати ефективно використовувати. За допомогою CryEngine створено багато популярних ігор та після освоєння почали створюватися ігри в інших жанрах. Але шутери CryEngine виходять найпопулярнішими, так у 2012 році компанія випустила WarFace, яка стрімко зросла у рейтингу онлайн-шутерів того часу.

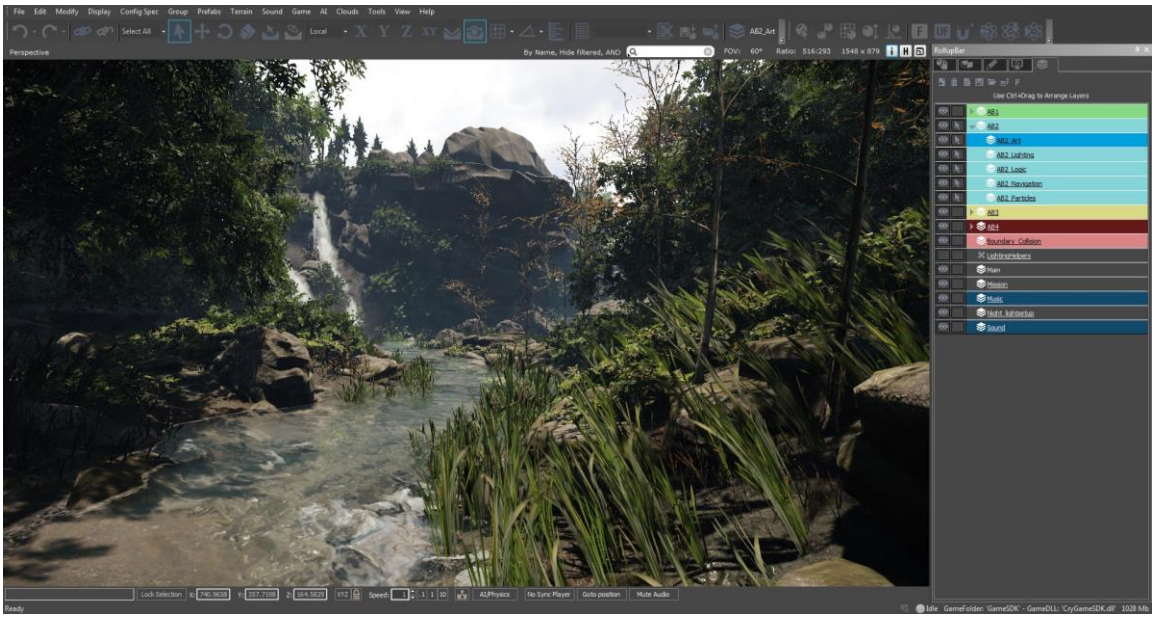

Рисунок 3.10 – Інтерфейс платформи «CryEngine»

Cocos2D

Cocos2D – від усіх інших платформерів використовується здебільше для мобільних застосунків. Застосовується для створення інтерактивних програм на

основі графічного інтерфейсу (рис. 3.11-3.12). Написаний на мові C++, з тонким шаром, що залежить від платформи.

Рисунок 3.11 – Логотип «Cocos2D»

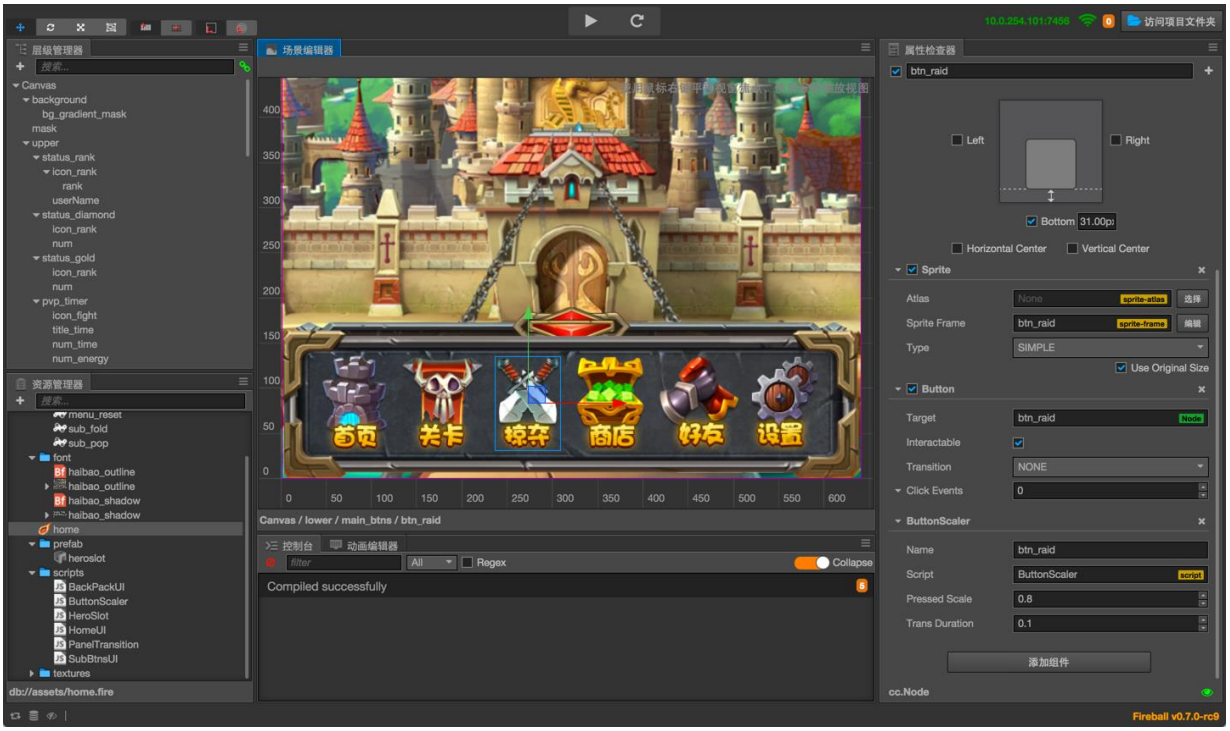

Рисунок 3.12 – Інтерфейс платформи «Cocos2D»

Таблиця переваг та недоліків для порівняння платформ розробки (табл. 3.1-3.2): Unity3, Unreal Engine, CryEngine, Cocos2D-x.

Таблиця 3.1 – Переваги та недоліки Unity3 та Unreal Engine

№	**Unity3D**		**Unreal Engine**	
	Переваги	*Недоліки*	*Переваги*	*Недоліки*
1	Розробка без специфічних інструментів та знань	Потребує встановлення багатьох плагінів, часто - платних	Підтримка усіх платформ(PC, Xbox, PlayStation, Mobile)	Виникають проблеми при створенні великих безшовних світів
2	Велика кількість асетів та плагінів для полегшення процесу розробки	Продуктивність швидкодії гри для складних сцен	Blueprints	Додавання на локацію багатьох AI-entity викликає просідання fps
3	Підтримка великої кількості технологій	Розробка ігор складніших за клікер та платформер потребує програміста з хорошими знаннями C#		Зручність використання: призначений для професіоналів
4	Безкоштовний			

Таблиця 3.2 – Переваги та недоліки CryEngine та Cocos2D-x

№	**CryEngine**		**Cocos2D-x**	
	Переваги	*Недоліки*	*Переваги*	*Недоліки*
1	Розробка ігор з фотореалістичною графікою	Труднощі при розробці: – наявність багів IDE, – слабке community.	Cross-platform software	Погано складена документація
2		Застаріла\відсутня документація	Open source	

Завдяки таблиці порівнянь можна обрати платформу для розробки програмного забезпечення (рис. 3.13).

	Unity3D	**Unreal Engine**	**CryEngine**	**Cocos2D-x**
Scripting language support	JavaScript, C#	C++	C#, C++, LUA	C++, Lua, JavaScript
Target platforms	multiplatform	multiplatform	multiplatform	multiplatform
Documentation	https://docs.unity3d.com/Manual/index.html	https://docs.unrealengine.com/4.27/en-US/	https://docs.cryengine.com/	https://docs.cocos2d-x.org/cocos2d-x/v4/en/upgradeGuide/
Free to use	+	+	+	+
Ease of use	+	-	-	+
2D & 3D Support	+	+	+	-
Community	+	+	+	+
Asset Store	+	+	+	+
Integrated Ad Monetization Framework	+	+	+	+

Рисунок 3.13 – Порівняльна таблиця

Для розробки застосунку було обрано Unity, є декілька переваг використання даної платформи:великий список ассетів і плагінів, можливість експорту та імпорту, Unity працює з DirectX, поєднання з системами рендеринга.

3.2.2 Оглядання технологій

Мова програмування – C#, оскільки вона використовує об’єктно-орієнтований підхід програмування. Альтернативою C# є UnityScript (US), який є дуже схожим на JS, але з особливостями розробки для Unity. Хоча US не складна у використанні, обрано C#, через велику кількість бібліотек та шаблонів, що дозволить не писати все вручну, зменшивши кількість часу, витрачену на розробку гри. Крім того, мова є повністю безкоштовною та простою у застосуванні, оскільки містить чітку документацію, яка буде зрозуміла для використання навіть новачкам.

Середовищем програмування застосунку обрано Visual Studio, що надає багато можливостей, наприклад, додавання проєкту, при експортуванні з Unity, задля редагування коду та створення класів поведінки, також середовище допомагає визначити метод, який планується використовувати, легко визначити нові шейдери. VS працює з тією ж системою проєкту, тому внесені зміни автоматично синхронізуються.

Завдяки використанню IntelliSense, тобто технології автоматичного доповнення від Microsoft, розробка пришвидшується, а точність коду – збільшується. Для налагодження ігрового ядра Unity, VS чудово підійде, оскільки має для цього всі необхідні інструменти, таким чином можна швидко виявити та усунути проблеми, крім того це зручно, оскільки можна задавати точки зупинки та оцінювати змінні й вирази. Також середовище розробки містить Unity Project Explorer, який дозволяє розробнику орієнтуватися та переміщуватися по проєкту швидко та зручно, як в самому застосунку Unity.

Базою даних обрано реляційну БД – SQLite, яка є сумісною з Unity та досить розповсюдженною у використані, при розробці ігор. Основною перевагою БД є можливість її використання для різних платформ, інакше кажучи кросплатформеність. Вимогами до БД є швидкість взаємодії та компактність, SQLite повністю задовільняє поставлені вимоги. Для роботи з БД використано DB Browser, вибір утиліти обґрунтований легкістю у створені таблиць та зв’язків

між ними, а також наповнення таблиць даними, без використання SQL. Дана утиліта допомагає спростити роботу та взаємодію з БД, а так же зменшити кількість витраченого часу на розробку БД для гри. Для роботи БД з Unity є досить багато бібліотек, які можна використовувати в залежності від платформи розробки. Також перевагами даної реляційної БД є:

- сумісність з Unity;
- компактність;
- розповсюдженість;
- чітко сформована документація;
- кросплатформеність;
- відкритий код;
- використання парадигми клієнт-сервер, тобто БД – частина проєкту, а не окремий процес;
- висока продуктивність (зменшення часу відгуку та спрощення програми;
- простота реалізації та використання в Unity.

Висновки до розділу 3

У даному розділі спроєктовано Desktop-гру клікер та розглянуто та проаналізовано ігрові платформи. Проведено аналіз з виявленням переваг та недоліків, обрано платформу Unity та мову програмування C# для розробки застосунку. Саме ця платформа та мова програмування буде більш зручною, дозволяючи розробляти гру без спеціальних навичок.

Розроблено діаграми: пакетів, класів, станів та переходів. Діаграми створені детально та лаконічно, що значно спрощують процес створення гри. В них описаний основний сюжет гри та основні можливості та функції. Виконання завдань та основний функціонал є основою гри.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІГРОВОГО ЗАСТОСУНКУ

Гра «Defense of the Castle» складається з головного меню в якому міститься чотири кнопки для переходу у інші вікна гри. Ігрове меню є основною обкладинкою гри, яке повинно буди простим та зрозумілим для кожного користувача.

4.1 Огляд дизайну гри

Головне меню гри є стартовим після запуску застосунку. Воно включає в себе: налаштування, ігровий магазин, вкладка з винагородами та кнопка для початку гри (рис. 4.1).

Рисунок 4.1 – Головне меню гри

Якщо перейти до меню налаштувань, ми переходимо до окремого вікна в якому відкривається доступ до звуку та музики в грі. Якщо натиснути на крайню ліву кнопку редагування звуку або музики, це поставить показники на мінімальне значення. Та навпаки, якщо натиснути на крайню праву кнопку, це поставить на максимальне значення музики або звуку (рис. 4.2).

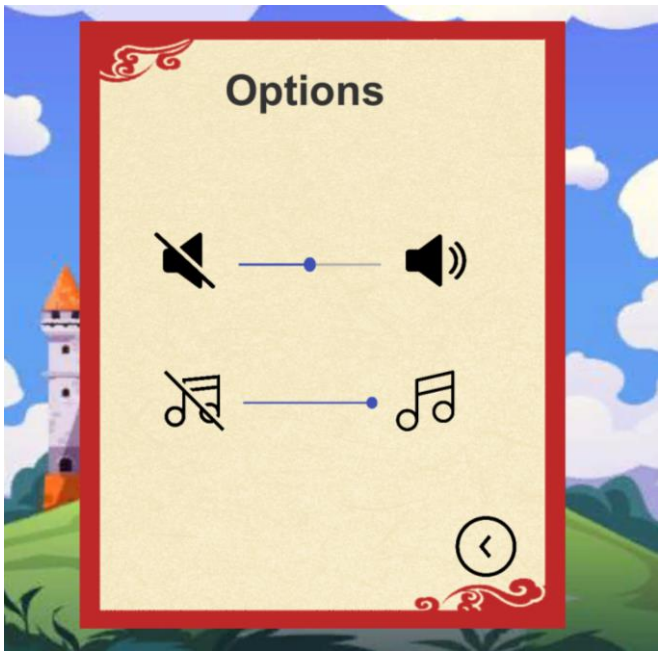

Рисунок 4.2 – Меню налаштувань

В кожному меню окрім основного є кнопка повернення до головного меню, вона розташована у правому нижньому кутку. У меню магазину користувач може придбати або продати покращення для веж. Якщо гравець бажає замінити старі покращення на більш нові, він може продати їх за половину ціни (рис. 4.3).

Рисунок 4.3 – Меню магазину

У магазині доступно дві вкладки, купити та продати. У вкладці купити

потрібно вибрати потрібну вежу та обрати покращення для неї. В наступній вкладці, продати, гравець може продати покращення. Нові покращення будуть відкриватися у ході гри, чим більше рівнів пройде гравець тим краще будуть покращення.

У відзнаках кожен гравець може відслідковувати свої результати, та отримувати винагороди за виконані досягнення (рис. 4.4). За виконані завдання гравець може отримати покращення для вежі, ці завдання складніші за звичайні. У ході виконання завдання заповнюватимемось шкала, яка показує на скільки відсотків завдання виконане. При виконаному завданні кнопка отримання винагороди виділятиметься зеленим кольором.

Рисунок 4.4 – Меню винагород

У меню старт відображаються рівні, які гравець пройшов або ні. За кожний рівень можна отримати до трьох зірок. Якщо гравець отримав зірку за рівень, вона відображатиметься зеленим кольором, а якщо у гравця не вдалося отримати всі зірки, будуть відображатися сірі зірки (рис. 4.5).

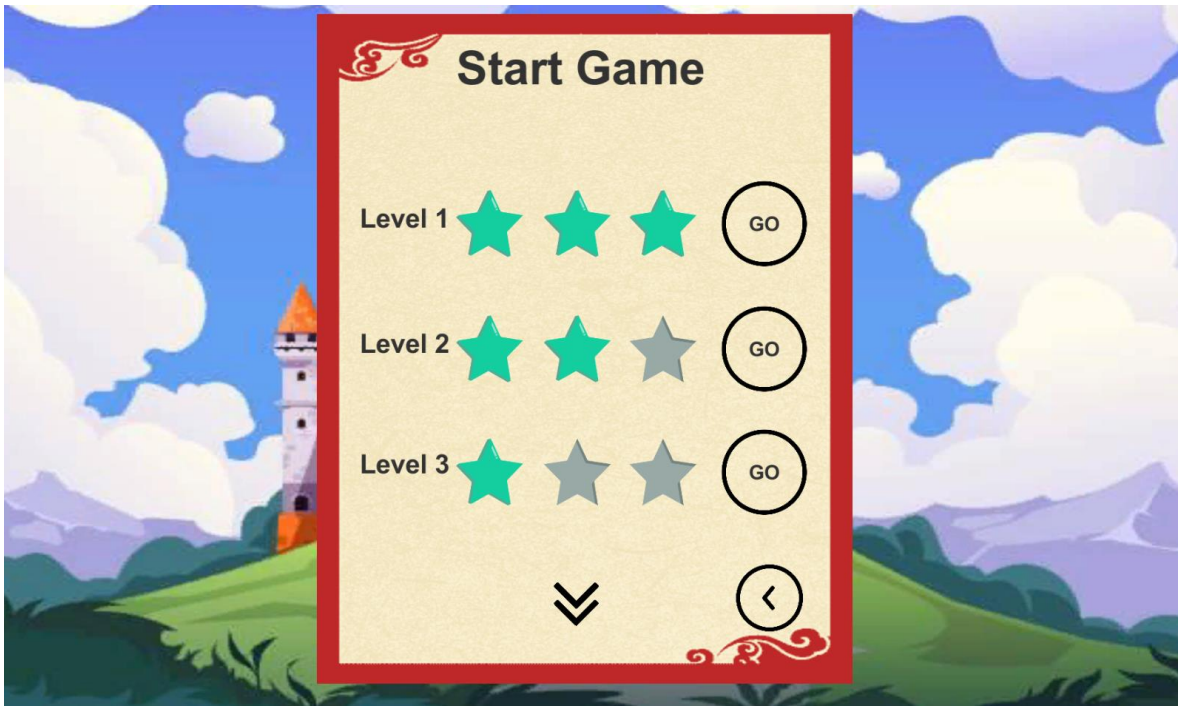

Рисунок 4.5 – Меню старту гри

Навіть якщо гравець пройшов рівень на три зірки, він все одно може проходити ще раз. Внизу є перехід до наступного рівня.

4.2 Сценарій

Гра має наступний сюжет: після того як гравець обирає рівень, він попадає на окрему мапу, кожен рівень має свою унікальність. Жанр гри Tower Defense включає в себе захист конкретного місця. На кожному рівні гравцеві потрібно захищати свій замок від ворогів, для цього потрібно будувати та покращувати вежі.

Рівень складається з:

– дороги, яка веде до замку;
– спеціально відведенні місця для веж;
– супротивників;
– таймеру;
– внутрішньоігрової валюти.

За відведенний до початку гри час гравець повинен поставити мінімум одну вежу, після чого запуститься гра. Супротивники які після підготовки до

рівня намагатимуться пройти до вашого замку. Найголовніше завдання – не дати їм пройди до замку. Для цього у відведенних на кожному рівні місцях потрібно ставити вежі (рис. 4.6).

Рисунок 4.6 – Ігрове поле

На ігровому полі позначені спеціально відведенні місця для веж. Також кнопка паузи гри та вихіж у головне меню. Якщо гравець не пройде до кінця рівень та вийде в головне меню, гра автоматично збереже не завершений процес.

Рисунок 4.7 – Зведення вежі

На початку кожного рівня дається різна кількість початкових кошт на взведення веж, гравець повинен обрати місце для возведення вежі, для більш ефективної оборони. Кожна вежа має свою ціну як для покупки так і для покращення (рис. 4.7). Після покупки вежі гравець натиснувши на неї може покращити її або продати. Продаж вежі поверне половину коштів використаних на неї з урахуванням покращень (рис. 4.8-4.9).

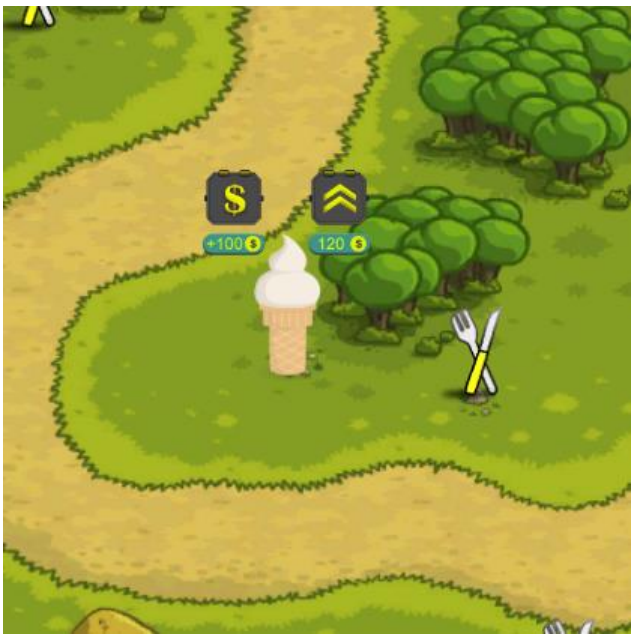

Рисунок 4.8 – Придбання вежі

Рисунок 4.9 – Придбання покращення для вежі

У грі необхідно зводити вежі для того, щоб супротивники не змогли дійти до замку. На кожному рівні гравець заробляє зірки. Якщо до замку не дійшли супротивники, гравець отримає три зірки за рівень (рис. 4.10).

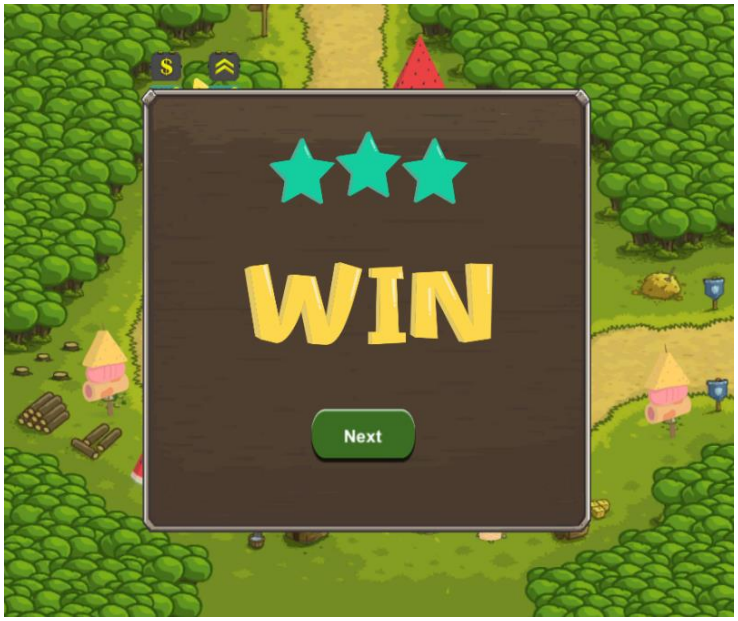

Рисунок 4.10 – Перемога у рівні

Поразкою у рівні вважається тільки коли гравець не набирає жодної зірки. Тобто за наявності однієї зірки, рівень буде пройдено, але цього може не вистачити для розблокування наступного рівня (рис. 4.11).

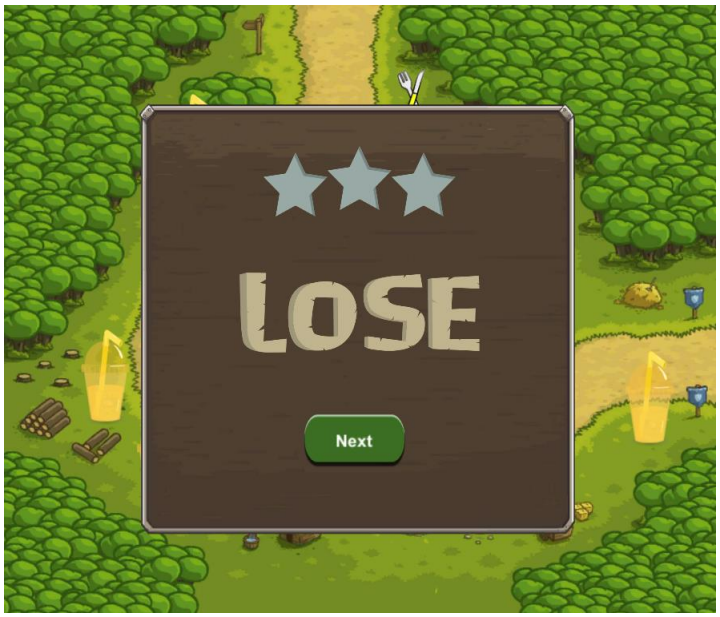

Рисунок 4.11 – Поразка у рівні

Перший рівень вважається пробним рівнем, за нього гравець обов’язково отримає хоча б одну зірку. Він є не складним, протягом рівня у гравця є багато часу, щоб обдумати правильність позицій та придбання веж.

Для того, щоб отримати доступ для другого рівня гравцю буде потрібна лише одна зірка. Але для отримання доступу на третій рівень гравець має отримати в сумі 4 зірки за перші два рівня. З кожним рівнем гравцю буде складніше заробляти зірки. Для того, щоб на першому рівні отримати 2 зірки треба щоб не більше 7 супротивників пройшли до замку, на другому рівні це значення вже становить 5. Так с кожним наступним рівнем буде все важче пройти його.

Для гри розроблені мапи для кожного з рівнів, моделі супротивників та анімація для них, також анімація для веж. Вежі змінюють анімацію атаки після покращення. Та в кожному рівні є свої особливості.

Для першого рівня створена особлива мапа з конкретними місцями для веж. У навчальному уроні гравцю відкрити всі види веж, але не все можна покращити. Так само в першому рівні не велика кількість супротивників (рис. 4.12).

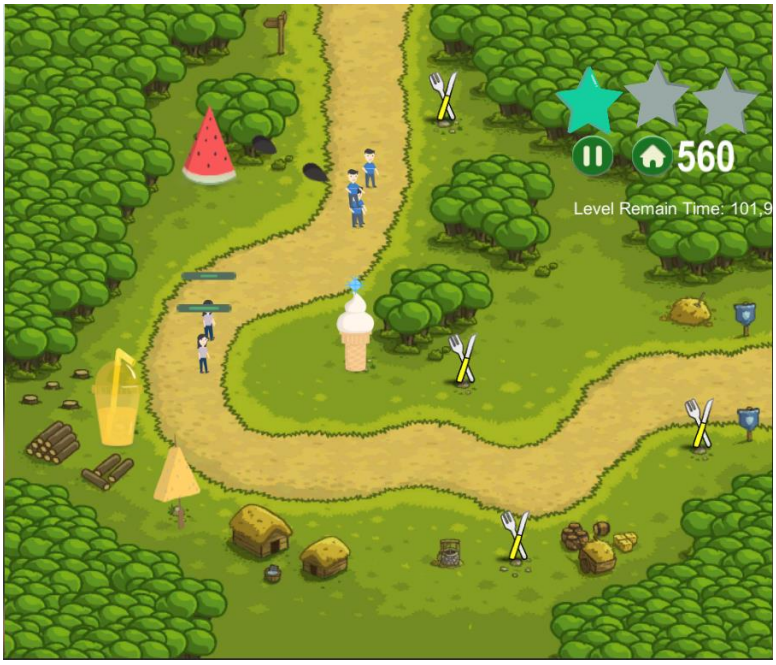

Рисунок 4.12 – Огляд першого рівня

У наступних рівнях поступово відкриватимуться більше покращень для веж та більше різновидів супротивників (рис. 4.13-14). Кожен вид противника має різну швидкість передвіщення і кількість здоров’я, над кожним противником буде відображатися кількість його здоров’я, коли вежа влучає по ньому. Але кожен із противників враховується як одиниця при проході до замку.

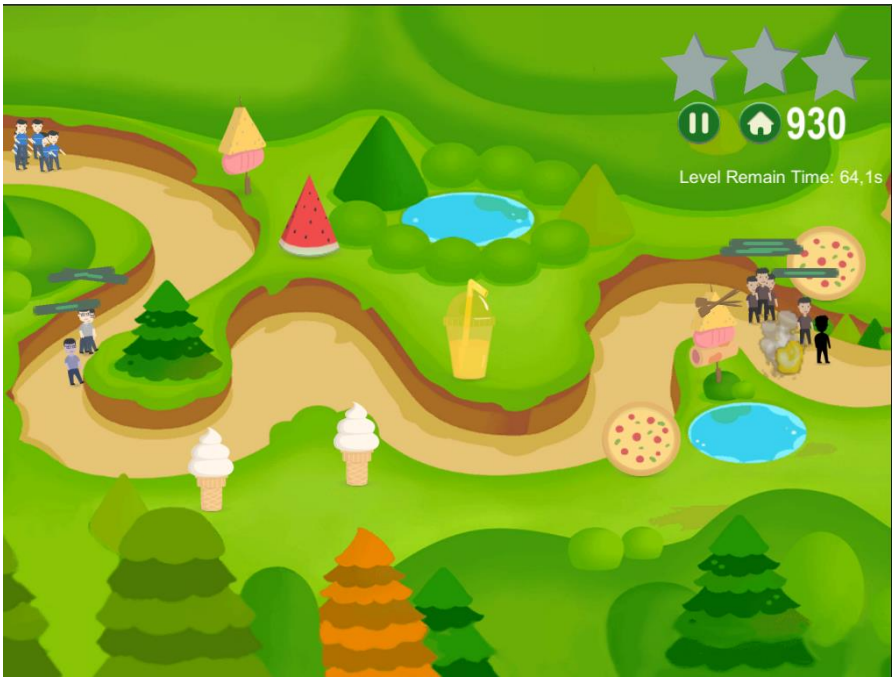

Рисунок 4.13 – Огляд другого рівня

Рисунок 4.14 – Огляд третього рівня

Гра включає в себе великий набір скриптів, завдяки яким розроблена анімація атаки та супротивників, та з їх використанням створюється багато рівнів без створення нової механіки рівня. Одним з головних скриптів є клас GameSceneController. Цей клас поєднує з собою всі інші скрипти, у ньому поєднується основний процес гри (рис. 4.15-4.16).

```
public class GameSceneController : System.Object, UserClickOp, GameFlow, MenuCli
    private static GameSceneController instance;
    private GameStatus myGameStatus;
    private GameModels myGameModels;
    private UIManager myUIManager;
    private GameMonitor myGameMonitor;

    private EnemyFactory enemyFactory;
    private CannonBallFactory cannonBallFactory;

    public static GameSceneController getInstance() {
        if (instance == null)
            instance = new GameSceneController();
        return instance;
    }

    private GameSceneController()
    {
        enemyFactory = EnemyFactory.getInstance();
        cannonBallFactory = CannonBallFactory.getInstance();
    }

    internal void setGameStatus(GameStatus _myGameStatus) {
        if (myGameStatus == null) {
            myGameStatus = _myGameStatus;
        }
    }

    internal void setGameModels(GameModels _myGameModels) {
        if (myGameModels == null) {
            myGameModels = _myGameModels;
        }
    }

    internal void setUIManager(UIManager _myUIManager)
    {
        if (myUIManager == null)
            myUIManager = _myUIManager;
```

Рисунок 4.15 – GameSceneController

```
public void buildDrinkTower() {
    myGameModels.buildDrinkTower();
}

public void buildPizzaTower() {
    myGameModels.buildPizzaTower();
}

public void buildMelonTower() {
    myGameModels.buildMelonTower();
}

public void buildGuandongTower() {
    myGameModels.buildGuandongTower();
}

public void sellTower() {

    myGameModels.sellTower();
    myGameStatus.resetClickedTowerIndex();
}

public void upgradeTower() {
    myGameModels.upgradeTower();
}

public void reward(int value)
{
    myGameModels.reward(value);
}

public void sufferDamage(int num)
{
    myGameStatus.sufferDamage(num);
    myUIManager.updateStarView(myGameStatus.getCurrentStar());
}
```

Рисунок 4.16 – Основний функціонал гри

4.3 Підключення та використатння Unity analytics та Unity remote config до проекту.

Unity Analytics — це проста, але потужна платформа даних, яка надає аналітику проєктів в Unity. Завдяки чому можна дізнатися: хто є гравцями у грі та їх поведінку під час ігрового процесу.

Платформи, що підтримуються:

- iOS;
- Android;
- Tizen;
- Universal Windows Platform;
- Mac, PC, Linux Standalone;
- WebGL - для версій 5.3 і вище.

Стандартні події легко інтегруються з Unity Analytics, а інструмент Analytics Event Tracker дозволяє їх швидко впроваджувати. Так можна вивчати поведінку гравців, завдяки спеціально розробленим подіям, орієнтованим на такі *ігрові процеси*, як:

- зручність навігації;
- просування гравця;
- створення першого враження про гру;
- залучення гравців;
- монетизація.

Додавання аналітики в проєкт.

Перший крок – увімкнення функції аналітики в проєкті, це можна зробити у вкладці Services (рис. 4.17)

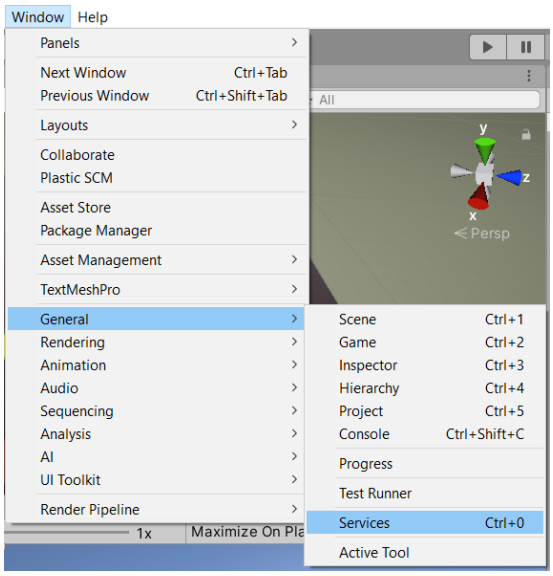

Рисунок 4.17 – Розташування вікна «Сервіси»

У вікні треба обрати функцію «Legacy Analytics» (рис. 4.18) або «Analytics», в більш ранніх версіях.

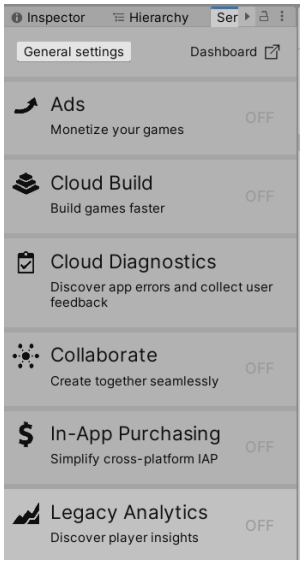

Рисунок 4.18 – Вигляд вікна «Сервіси»

Далі вказуємо організацію, в даному випадку особистий акаунт GGdays (рис. 4.19) та перемикаємо налаштування аналітики з режиму «Off» на «On».

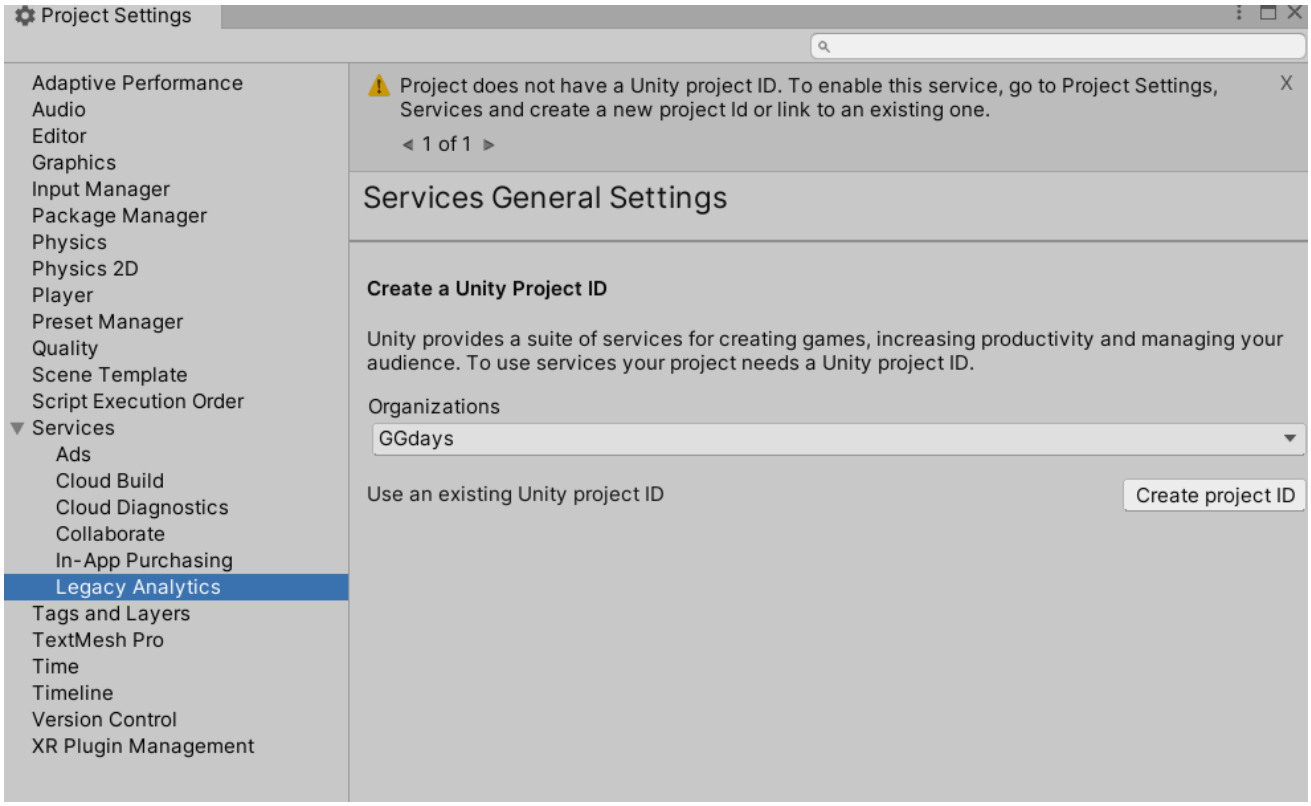

Рисунок 4.19 – Створення проєктного ID

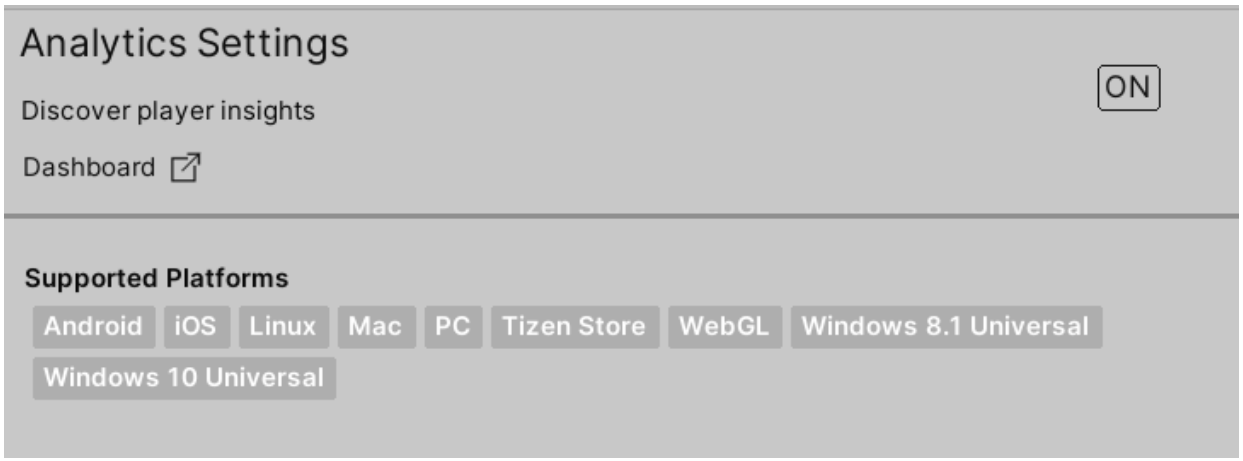

Рисунок 4.20 – Вмикання аналітики

Після чого перевіряємо на сайті справність підключенної аналітики до проєкту (рис. 4.21).

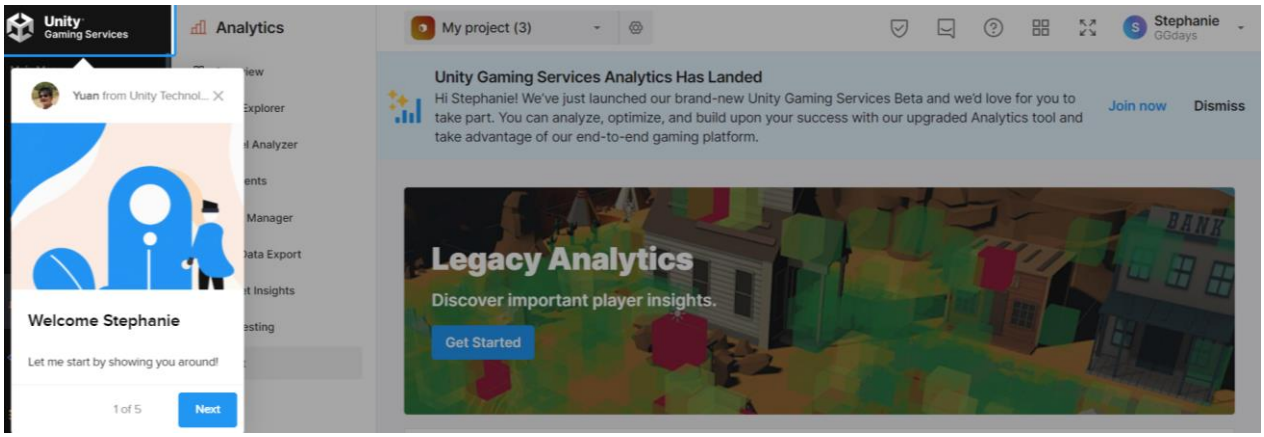

Рисунок 4.21 – Вигляд головної сторінки

Скрипт для роботи з аналітикою

Коли до проєкту підключена аналітика – її можна використовувати, для чого написано спеціальний скрипт «MyAnalytics» (рис. 4.22). Але оскільки скрипт буде допомагати збирати дані необхідно визначити, які саме. В даному випадку аналітика буде допомагати розробнику отримати відповіді на наступні питання:

– На якому рівні найчастіше програє гравець?
– За якого саме персонажа найчастіше програє?
– Який рівень користується найбільшою популярністю?
– На якому персонажі грають найчастіше?

Оскільки, на жаль, обчислювальна потужність комп'ютера не дозволяє рендерити рівні, тому було вирішено спростити задачу, написавши код-обгортку «TestAnalyt» (рис. 4.23), що буде надавати рандомні дані (у вказаних діапазонах) та список з іменами персонажів. Для перевірки працездатності скрипту та роботи Unity Analytics має бути достатньо.

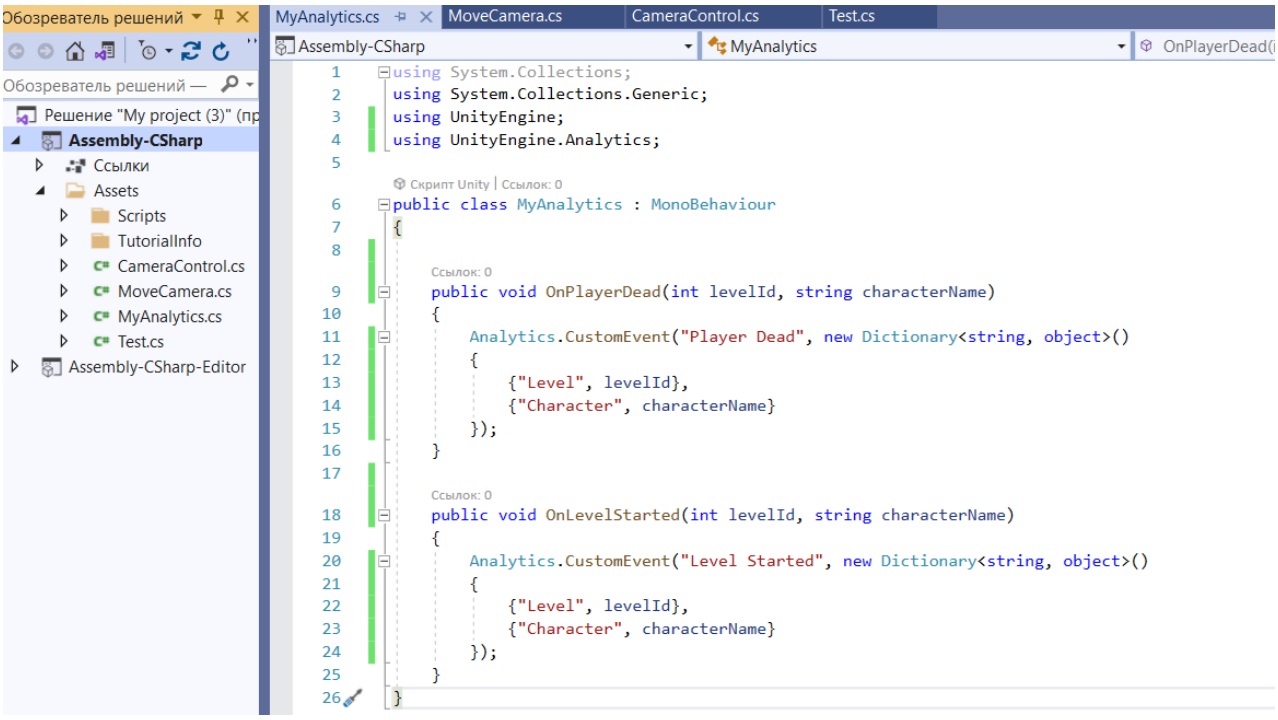

Рисунок 4.22 – Код скрипту

```
Скрипт Unity | Ссылок: 0
public class TestAnalyt : MonoBehaviour
{
    [SerializeField] private MyAnalytics analytics;
    [SerializeField] private string[] Names;

    Ссылок: 0
    public void PlayerDead()
    {
        int levelId = Random.Range(1, 11);

        int i = Random.Range(0, this.Names.Length + 1);
        string chatacterName = this.Names[i];

        this.analytics.OnPlayerDead(levelId, chatacterName);

    }

    Ссылок: 0
    public void LevelStart()
    {
        int levelId = Random.Range(1, 11);

        int i = Random.Range(0, this.Names.Length + 1);
        string chatacterName = this.Names[i];

        this.analytics.OnLevelStarted(levelId, chatacterName);

    }
}
```

Рисунок 4.23 – Скрипт-обгортка з даними

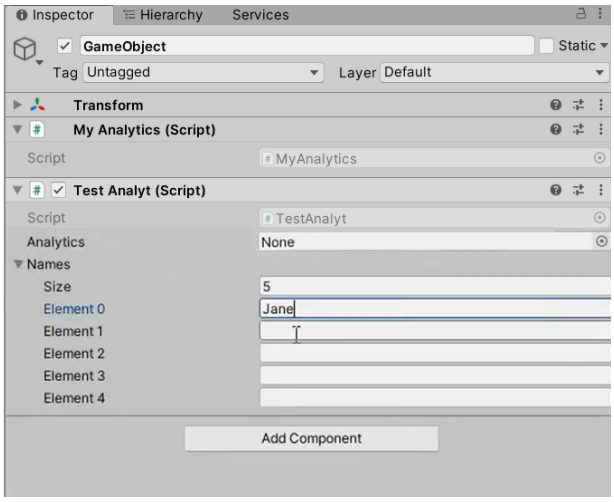

Рисунок 4.24 – Прив’язка скриптів до об’єкту та заповнення даними

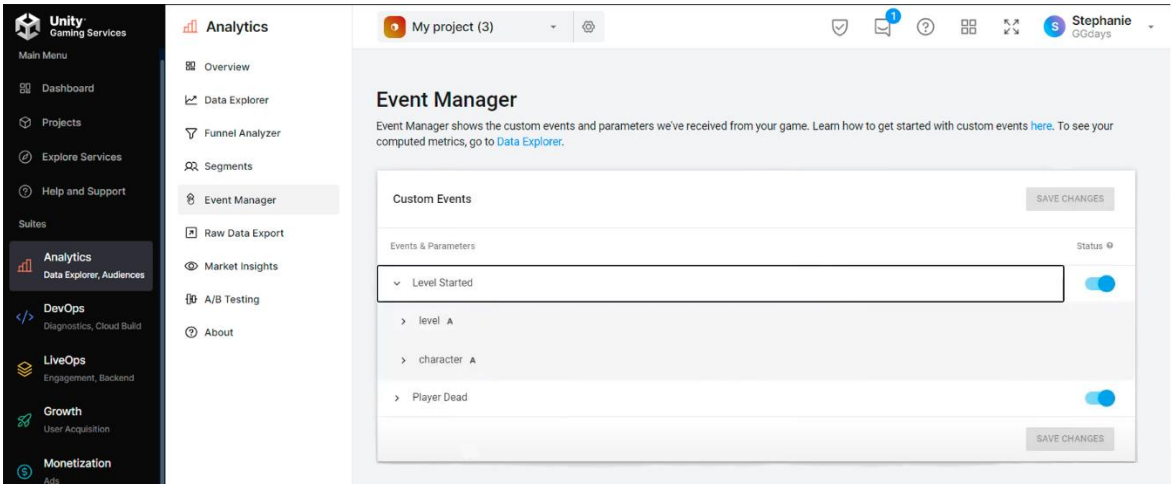

Рисунок 4.25 – Вигляд даних на dashboard.unity3d.com

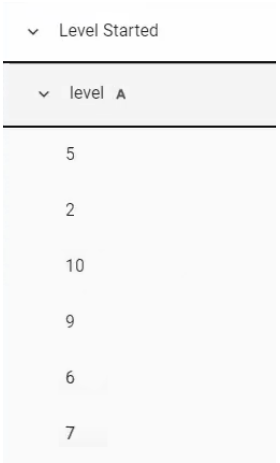

Рисунок 4.26 – Відображення деталей

Remote Config — це хмарний сервіс, який дозволяє налаштувати дизайн гри без розгортання нових версій програми. За допомогою Remote Config можна:

– Адаптувати свою гру під різні типи гравців;

– Визначити заміни гри, що визначають, які гравці отримують оновлення налаштувань і коли.

Установка плагіну

Для роботи з плагіном його треба встановити у проєкті, завдяки Package Manager (рис. 4.27)

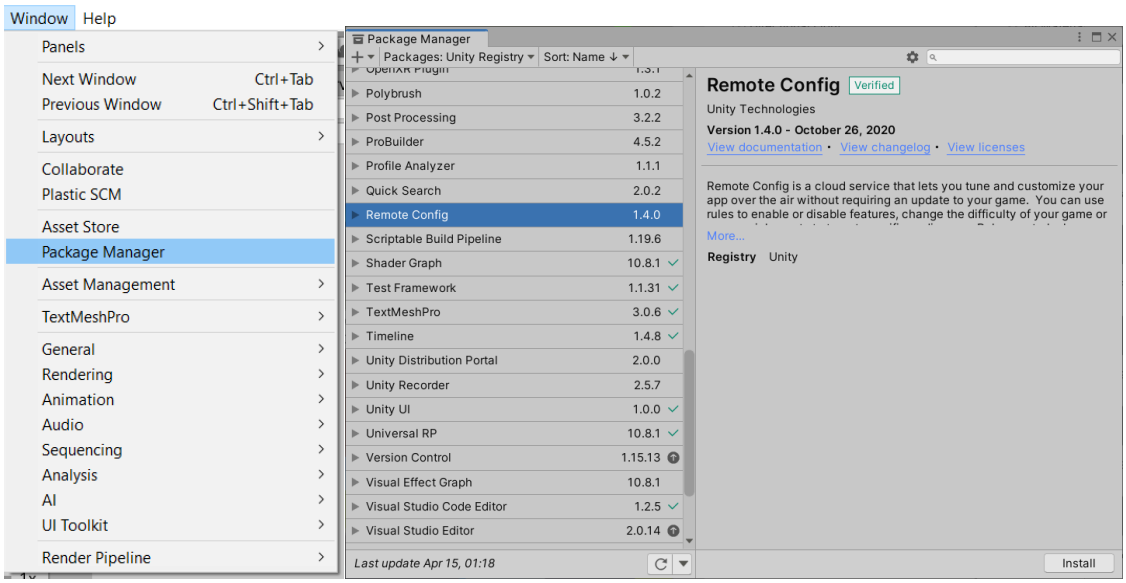

Рисунок 4.27 – Установка Remote Config

Після встановлення треба відкрити вікно Package Manager та натиснути кнопку «Pull», яка візьме дані з dashboard. Після чого можна додавати власні дані (рис. 4.28) та відправляти до dashboard кнопкою «Pull» (рис. 4.29).

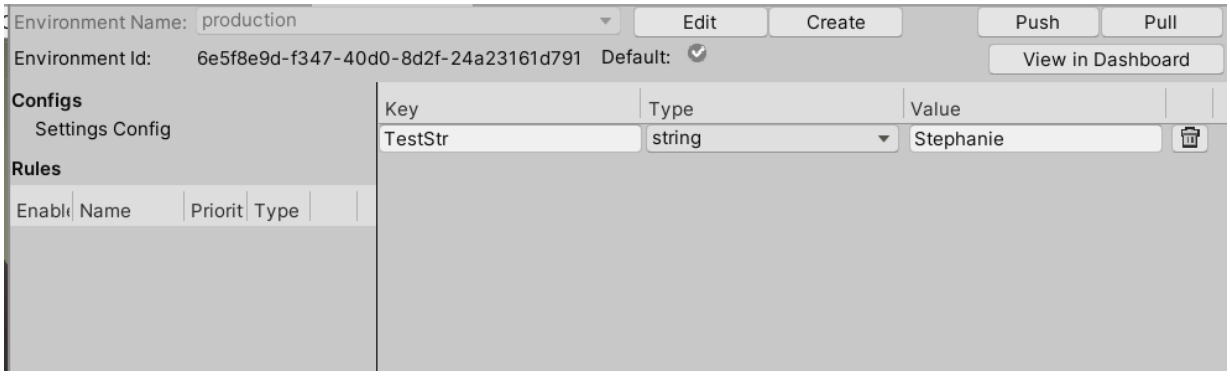

Рисунок 4.28 – Додавання даних

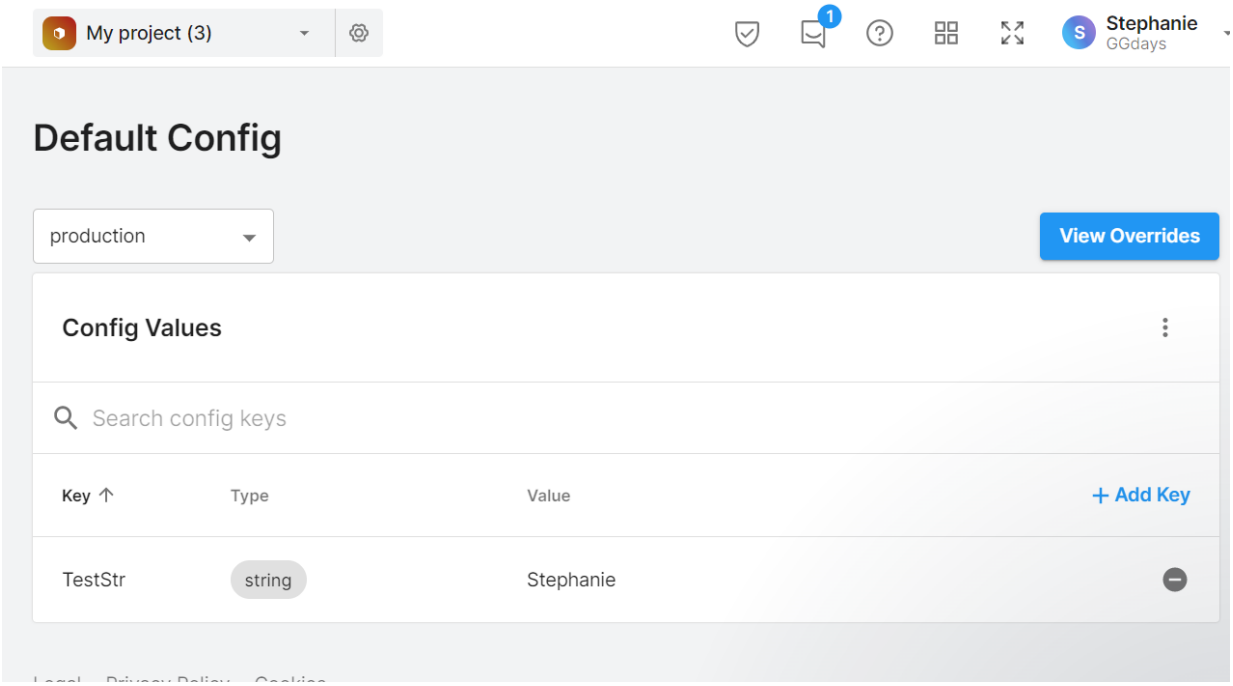

Рисунок 4.29 – Перегляд на сайті

Також можна змінювати дані (операціями CRUD) і на сайті, але після цього необхідно власноруч оновити дані в проєкті.

Скрипт для роботи з Remote Config

Оскільки інколи необхідно внести певні зміни під час ігрового процесу, не перериваючи його, можна створити скрипт «RCtestUPD» (рис. 4.30), що дозволить використовувати Remote Config та вносити зміни при запущеній грі.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Unity.RemoteConfig;

Скрипт Unity | Ссылок: 0
public class RCtestUPD : MonoBehaviour
{
    // Необхідні структури для RemoteConfig
    Ссылок: 2
    public struct userAttributes { }
    Ссылок: 2
    public struct appAttributes { }

    Сообщение Unity | ссылка: 1
    void Start()
    {
        ConfigManager.FetchConfigs<userAttributes, appAttributes>
            (new userAttributes(), new appAttributes());

    }

    // Update is called once per frame
    Сообщение Unity | Ссылок: 0
    void Update()
    {
        Debug.Log(ConfigManager.appConfig.GetString("TestStr"));
        Debug.Log(ConfigManager.appConfig.GetString("TestStr2"));
        Debug.Log(ConfigManager.appConfig.GetInt("TestInt"));
        Debug.Log(ConfigManager.appConfig.GetBool("Bool"));

        Start();
    }
}
```

Рисунок 4.30 – Код скрипта

Key ↑	Type	Value	+ Add Key
Bool	bool	true	⊖
TestInt	int	409	⊖
TestStr	string	Stephanie	⊖
TestStr2	string	Bondarenko	⊖

Рисунок 4.31 – Дані, що використовує скрипт

Тепер при застосуванні скрипта дані (рис. 4.31) будуть оновлюватися та їх можна буде змінювати під час гри.

4.3 Тестування застосунку

Таблиця 4.1 – Реєстрація користувача

Діючі особи	Користувач, система

Кінець таблиці 4.1

Мета	Реєстрація користувача.
Передумова	Користувач не має акаунту.
Успішний сценарій: 1. Користувач переходить до меню реєстрації. 2. Користувач вводить ім’я, яке служитиме логіном та пароль. 3. Система отримані дані та зберігає. 4. Система створює новий акаунт. 5. Користувач входить у власний акаунт.	
Сценарій успішний. Створенно новий акаунт.	
Розширення	
1а	Таке ім’я вже існує. Система повідомить про це, та запитає у користувача інший логін
Усі сценарії розширення успішно виконані.	

Таблиця 4.2 – Отримання винагород

Діючі особи	Гість, система
Мета	Отримання винагороди.
Передумова	Користувач має виконати завдання.
Успішний сценарій: 1. Користувач переходить до меню винагород. 2. Користувач вибирає завдання яке виконав. 3. Система перевіряє процес виконання звадання. 4. Система видає винагороду. 5. Користувач отримує винагороду.	

Кінець таблиці 4.2

Сценарій успішний. Отримання винагороди.	
Розширення	
1а	Завдання не виконане. Система не видасть винагороду користувачу.
Усі сценарії розширення успішно виконані.	

Висновки до розділу 4

В четвертому розділі створенно основне меню гри. Розроблене головне меню, меню налаштуваннь, магазин, меню досягнень, меню налаштувань. Описані основні функції гри, розроблені моделі та анімації для веж та супротивників. Описано розробку мап для рівнів та покращення для веж.

Створено скрипти, що дозволяють працювати з Unity Analytics та Remote Config. Ознайомлено з платформою даних Unity Analytics, що надає аналітику проєктів в Unity. Завдяки чому можна дізнатися хто є гравцями у грі та їх поведінку під час ігрового процесу. Також набуто практичних навичок у роботі з хмарним середовищем Remote Config, який дозволяє налаштувати дизайн гри без розгортання нових версій програми.

Проведенно тестування основних функцій гри з успішними сценаріями та розширеннями. Завдяки проведеному аналізу усунено недоліки які можуть завадити комфортній грі для користувача.

ВИСНОВКИ

Під час виконання кваліфікаційної работи бакалавра створенно Desktop-гру клікер у жанру Tower defense на базі Unity. Описано загальний алгоритм проєкту.

Для реалізації застосунку виконано наступні завдання:

– проведенно аналіз застосунків-аналогів;
– розроблено загальний алгоритм реалізвції;
– сформовано специфікацію вимог;
– побудованно UML-діаграми;
– розроблено основний алгоритм роботи;
– розроблено моделі та анімації;
– реалізовано ігровий застосунок.

Описано основний фунціонал гри та основні специикації, у результаті чого розроблено інтерфейс застосунку, головного меню та інших основних функцій гри. Гра складається із сцен головного меню та рівнів. Описано сюжет гри та геймплей, основні можливості користувача.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 3D Adventure Game Using Unity / R. R et al. Bonfring International Journal of Software Engineering and Soft Computing. 2019. Vol. 9, no. 2. P. 16–20. URL: https://doi.org/10.9756/bijsesc.9015 (дата звернення: 10.05.2022).

2. Ahearn L. 3D game environments: Create professional 3D game worlds. Amsterdam : Elsevier/Focal Press, 2008. P. 4-23 (дата звернення: 10.05.2022).

3. Blackman S. Action Objects. Beginning 3D Game Development with Unity. Berkeley, CA, 2011. P. 309–405. URL: https://doi.org/10.1007/978-1-4302-3423-4_7 (дата звернення: 10.05.2022).

4. Blackman S. Game Environment. Beginning 3D Game Development with Unity 4:. Berkeley, CA, 2013. P. 637–677. URL: https://doi.org/10.1007/978-1-4302-4900-9_17 (дата звернення: 10.05.2022).

5. Blackman S. Message Text. Beginning 3D Game Development with Unity. Berkeley, CA, 2011. P. 453–504. URL: https://doi.org/10.1007/978-1-4302-3423-4_10 (дата звернення: 10.05.2022).

6. Dawes A. Creating a Game Framework. Windows 8 and Windows Phone 8 Game Development. Berkeley, CA, 2013. P. 71–110. URL: https://doi.org/10.1007/978-1-4302-5837-7_3 (дата звернення: 10.05.2022).

7. Game Effects. 3D Game Textures. 2016. P. 339–361. URL: https://doi.org/10.1201/b21567-10 (дата звернення: 10.05.2022).

8. Jakoš F., Verber D. Learning Basic Programing Skills With Educational Games. Journal of Educational Computing Research. 2016. Vol. 55, no. 5. P. 673–698. URL: https://doi.org/10.1177/0735633116680219 (дата звернення: 10.05.2022).

9. Lanzinger F. 3D. 3D Game Development with Unity. Boca Raton, 2022. P. 73–87. URL: https://doi.org/10.1201/9780429328725-3 (дата звернення: 10.05.2022).

10. Lanzinger F. A 3D Game. 3D Game Development with Unity. Boca Raton, 2022. P. 25–72. URL: https://doi.org/10.1201/9780429328725-2 (дата звернення: 10.05.2022).

11. Lanzinger F. Epilogue. 3D Game Development with Unity. Boca Raton, 2022. P. 389–390. URL: https://doi.org/10.1201/9780429328725-25 (дата звернення: 10.05.2022).

12. Leighty J. How to develop a strong high school kicking game. West Nyack, N.Y : Parker Pub. Co., 1967. 210 p.

13. Murray J. W. Blaster Game Example. C# Game Programming Cookbook for Unity 3D. 2nd ed. 2021. P. 263–281. URL: https://doi.org/10.1201/9780429317132-15 (date of access: 27.04.2022).

14. Patil P. Software for Programing Contest. International Journal for Research in Applied Science and Engineering Technology. 2019. Vol. 7, no. 5. P. 3712–3714. URL: https://doi.org/10.22214/ijraset.2019.5610 (дата звернення: 10.05.2022).

15. Press M., Corporation M. Visual C++ User's Guide: Microsoft Visual C++ : Development System for Windows 95 and Windows Nt, Version 4 (Microsoft Visual C++). 2nd ed. Microsoft Pr, 1995. 712 p.

16. Publishing V. G. D. Video Game Developing Difficulty Level: Journal / Notebook / Diary Gift - 6 X9 - 120 Pages - White Lined Paper - Matte Cover. Independently Published, 2020. 120 p.

17. Suvak J. Game Programming 101. Learn Unity 3D Programming with UnityScript. Berkeley, CA, 2014. P. 27–60. URL: https://doi.org/10.1007/978-1-4302-6587-0_2 (дата звернення: 10.05.2022).

18. Taylor A. G. Developing with Unity and Visual Studio. Develop Microsoft HoloLens Apps Now. Berkeley, CA, 2016. P. 75–90. URL: https://doi.org/10.1007/978-1-4842-2202-7_9 (дата звернення: 10.05.2022).

19. Visual Studio. Beginning ASP.NET 3.5 in VB 2008. Berkeley, CA, 2007. P. 81–119. URL: https://doi.org/10.1007/978-1-4302-0431-2_4 (дата звернення: 10.05.2022).

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

DESKTOP-ГРА ЖАНРУ КЛІКЕР НА БАЗІ UNITY

СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ

ВПЛИВ КОМП'ЮТЕРНОЇ ТЕХНІКИ НА ЗДОРОВ'Я

Спеціальність 121 «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.21810924

Студент _____ Д. В. Самковський
підпис

«__» _____ 2022 р.

Консультант канд. техн. наук, доцент _____ А. О. Алексєєва
підпис

«__» _____ 2022 р.

Миколаїв – 2022

ЗМІСТ

ВСТУП

Кваліфікаційна робота бакалавра передбачає створення комп’ютерного застосунку-гри у жанрі Tower Defence. Через що, в результаті кваліфікаційної роботи бакалавра, розроблено комп’ютерний застосунок з елементами 3D моделювання та протестовано його в умовах комп’ютерної аудиторії ЧНУ імені Петра Могили.

Охорона праці є важливою складовою будь-якого виробництва, відзначаючи людину, як головну цінність, адже її безпека і хороше здоров'я дозволяють зробити виробничий процес більш чітким, що підвищить рентабельність самого підприємства. Правильно організована система охорони праці дисциплінує самого працівника і, як наслідок, веде до підвищення продуктивності виконуваної роботи і збільшення її ефективності.

Охорона праці спрямована не тільки на безпеку трудового процесу, а й на профілактику захворювань, організацію харчування і відпочинку працівників, забезпечення їх спецодягом та засобами гігієни. Охорона праці також в повній мірі несе відповідальність за виконання власником соціальних гарантій і пільг. Правильно організована охорона праці дозволяє працівникам відчувати себе захищеним, в результаті чого підвищується зацікавленість в роботі і зменшується плинність кадрів.

Тому в даному розділі розглянуто умови праці у комп’ютерному класі ЧНУ імені Петра Могили, та запропоновано деякі заходи з покращення умов праці на підприємстві.

Виконані розрахунки інтегральної бальної оцінки умов праці на підприємстві, ступеню втоми працівників, та ступеню їхньої працездатності. Дані розрахунки виконано до та після заходів, що пропонується впровадити для покращення умов праці на підприємстві.

1 ОЦІНКА УМОВ ПРАЦІ В КОМП'ЮТЕРНІЙ АУДИТОРІЇ

У даному розділі роботи розглянуто питання охорони праці у комп'ютерній аудиторії ЧНУ імені Петра Могили, в якій виконувалось тестування розробленого програмного застосунку.

1.1 Опис обраного виробничого приміщення, робочих місць, їх обладнання та складання вихідних даних для кількісної оцінки умов праці

Приміщення комп'ютернії аудиторії ЧНУ імені Петра Могили розташовано на третьому поверсі чотириповерхової будівлі, що знаходиться у м. Миколаєві на вулиці Десантників, 68. Розміри приміщення складають $a \times b \times H = 12{,}0 \times 8{,}0 \times 4{,}0$ м. У приміщенні влаштовано п'ять металопластикових вікон (з подвійними склопакетами) розмірами $c \times d = 1{,}2 \times 2{,}4$м. Приміщення зовнішньою стіною спрямовано за азимутом в діапазоні Аз =136…225 град.

Приміщення має доволі сучасний офісний інтер'єр. Стеля виконана у вигляді підвісної конструкції із синтетичного матеріалу, та складається з окремих квадратних плиток світло-сірого кольору. Стіни оштукатурені, покриті фарбою зеленого кольору. Підлога покрита лінолеумом, що імітує паркет коричневого кольору. Вікна обладнані світлозахисними пристроями у вигляді горизонтальних регульованих жалюзі.

У приміщенні розташовано 26 робочих місць. всі обладнані робочими стаціонарними комп'ютерами під управлінням ОС Windows. Для доступу кожного з комп'ютерів до локальної обчислювальної мережі на кожні 2 робочих місця розраховано подвійну телекомунікаційну розетку. Для підключення всього обладнання, використовуються мережеві фільтри. За дозволом адміністрації, іноді використовуються кондиціонер або телевізор. Робочі місця розділені між собою перегородками. Загальний вид обраного виробничого приміщення представлено на рис. 1.1.

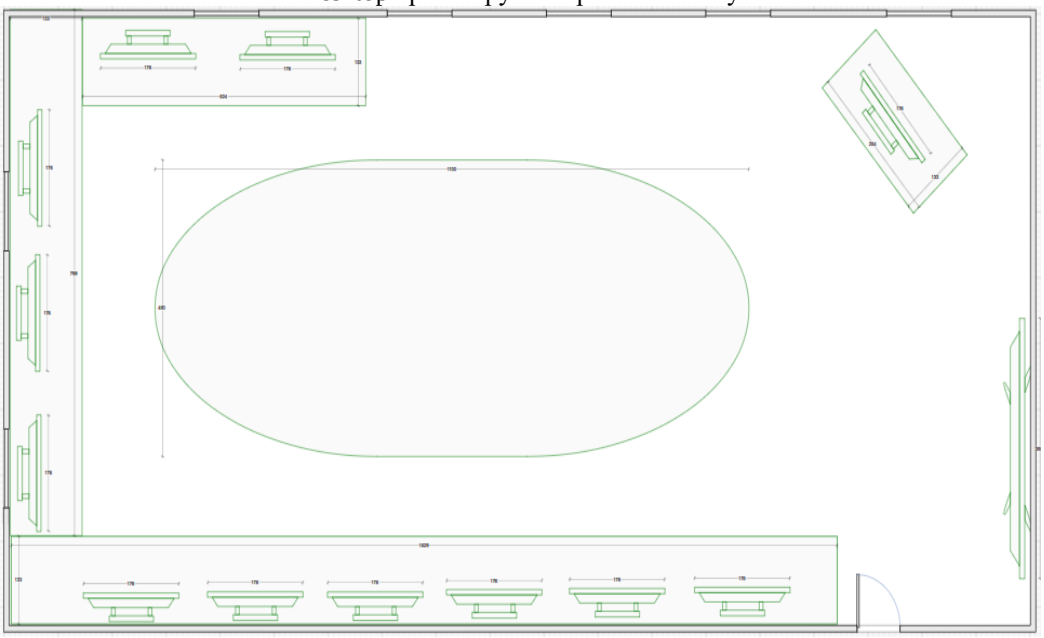

Рисунок 1.1 – Загальний вигляд обраного виробничого приміщення

Напруга джерела живлення електроспоживанні техніки – 220 В. Електромережа виконана у вигляді трипровідної з дотриманням усіх вимог нормативних документів. За небезпекою ураження електричним струмом приміщення відноситься до приміщень без підвищеної небезпеки ураження електричним струмом [2].

Мікрокліматичні умови у літній період (частково у перехідний) забезпечується двома кондиціонерами, потужності яких не завжди вистачає для умови в м. Миколаїв. У зимовий період опалення здійснюється центральною системою, яка також не завжди забезпечує необхідний тепловий режим. Деякий дискомфорт, пов'язаний із виробничим освітленням, може відчуватися у певні часи робочого дня.

Пожежна безпека в обраному виробничому приміщенні забезпечується дотриманням вимог НПАОП 0.00-1.28-10 [4].

За даними контрольних обстежень, виконанням необхідних вимірів, а також експертних оцінок здійснена оцінка умов праці в обраному виробничому приміщенні (табл. 1.1). Наведені у табл. 1.1 відомості являються вихідними даними для виконання подальших розрахункових робіт даного розділу кваліфікаційної роботи бакалавра.

Таблиця 1.1 – Фактори умов праці у відділі розробки програмного забезпечення

№ з/п	Фактори умов праці на робочому місці	Значення показника	Тривалість дії фактору, хв.
1	Температура повітря на робочому місці ℃ (РМ) у виробничому приміщенні, : теплий період холодний період	27 20	420 420
2	Відносна вологість повітря на РМ, %	55	420
3	Швидкість руху повітря на РМ, м/с	0,13	420
4	Освітленість на РМ, лк	220	420
5	Мінімальний розмір об’єкта розпізнавання, мм	0,5	360
6	Виробничий шум, дБА	0,7	420
8	Токсична речовина, озон, кратність перевищення гранично допустимої концентрації	-	420
9	Виробничий пил (паперовий та ін.), кратність перевищення гранично допустимої концентрації	-	420
10	Робоче місце (РМ), поза та переміщення у просторі	РМ стаціонарне, маса переміщення вантажу ≤ 5 кг	420
11	Кількість важливих об’єктів, що спостерігаються при роботі	2	420
12	Тривалість зосередженого спостереження, % від загального часу тривалості робочої зміни	80	384
13	Тривалість повторюваних операцій, с	120	384
14	Змінність роботи	Ранкова зміна	480
15	Тривалість безперервної роботи за добу, годин	7	420

Кінець таблиці 1.1

№ з/п	Фактори умов праці на робочому місці	Значення показника	Тривалість дії фактору, хв.
16	Режим праці та відпочинку	Обґрунтований з включенням музики та гімнастики	480
17	Нервово-емоційне навантаження	Складні дії за заданим планом при дефіциті часу	420
18	Кількість рухів пальців на годину	3000	384
19	Монотонність, тривалість операцій, які повторюються , с	10	384

1.2 Інтегральна оцінка умов праці в обраному виробничому приміщенні

Для інтегральної оцінки умов праці [1] в обраному виробничому приміщенні скористаємося даними табл. 1.1 та здійснимо оцінку питомої ваги кожного із представлених там факторів виробничого середовища та трудового процесу. У табл. 1.2 представлені параметри, що необхідні для інтегральної оцінки умов праці:

- $x_{ні}$ – нормативне значення i – того фактору умов праці;
- $x_{абі}$ – дійсне значення i – того фактору умов праці;
- x_{xi} – оцінка i – того фактору умов праці, балів;
- t_i – тривалість дії i – того фактору умов праці, хв.;
- $t_{питі}$ – відносна тривалість дії i – того фактору умов праці хв., тобто:
- $t_{\text{пит}_\text{і}} = \frac{t_i}{t_p} = \frac{t_i}{480}$;
- $x_{фі}$ – фактична оцінка питомої ваги i – того фактору умов праці:
- $x_{\text{ф}_\text{і}} = x_{x_i} t_{\text{пит}_\text{і}} = x_{x_i} \frac{t_i}{480}$.

За даними табл. 1.2 та 1.3 визначаємо елемент умов праці, який одержав у балах найбільшу оцінку x_{max} . Таких елементів може бути декілька.

Таблиця 1.2 – Параметри, що необхідні для розрахунку інтегральної бальної оцінки умов праці на робочому місці

№ з/п	Фактор умов праці на робочому місці	Нормоване значення фактора x_{Hi}	Абсолютна оцінка x абі	Абсолютна оцінка у балах x_{xi}
1	Температура повітря на робочому місці (РМ) у виробничому приміщенні, °С - теплий період - холодний період	23…25 21…23	26 21	2 1
2	Відносна вологість повітря на РМ, %	40..60	55	2
3	Швидкість руху повітря на РМ, м/с	<0,2	0,13	1
4	Освітленість на РМ, лк	200	220	3
5	Мінімальний розмір об’єкта розпізнавання, мм	>1	0,5	2
6	Виробничий шум, дБА	50	0,7	1
7	Інтенсивність теплового випромінювання, Вт/м²	≤140	130	1
8	Токсична речовина, озон, кратність перевищення ГДК	≤1	-	1
9	Виробничий пил (паперовий), кратність перевищення гранично допустимої концентрації	≤1	-	1

Кінець таблиці 1.2

№ з/п	Фактор умов праці на робочому місці	Нормоване значення фактора $x_{Нi}$	Абсолютна оцінка x абі	Абсолютна оцінка у балах x_{Xi}
10	Робоче місце (РМ), поза та переміщення у просторі	РМ стаціонарне, маса переміщення до 5 кг	РМ стаціонарне, маса переміщення до 5 кг	1
11	Кількість важливих об’єктів спостереження	<5	2	1
12	Тривалість зосередженого спостереження, % зміни	<25	80	4
13	Тривалість повторюваних операцій, с	>100	120	1
14	Змінність роботи	Ранкова	Ранкова	1
15	Тривалість безперервної роботи за добу, годин	<8	7	2
16	Режим праці та відпочинку	Обґрунтований з вкл.. музики та гімнастики	Обґрунтований з вкл.. музики та гімнастики	1
17	Нервово-емоційне навантаження	Прості дії за індивідуальним планом	Виконання складних дій за заданим планом при дефіциті часу	4
18	Кількість рухів пальців на годину	<360	3000	4
19	Монотонність, тривалість операцій, які повторюються , с	>100	10	4

Таблиця 1.3 – Додаткові параметри, що необхідні для розрахунку інтегральної бальної оцінки умов праці на робочому місці

№ з/п	**Фактор умов праці на робочому місці**	**Тривалість дії фактора** t_i	**Тривалість дії фактора у долях робочої зміни,** $t_{пит\ i}$	**Фактична оцінка питомої ваги фактора** $x_{фi}$
1	Температура повітря на робочому місці (РМ) у виробничому приміщенні, °С теплий період - холодний період	420 420	0,875 0,875	1,75 0,875
2	Відносна вологість повітря на РМ, %	420	0,875	1,75
3	Швидкість руху повітря на РМ, м/с	420	0,875	0,875
4	Освітленість на РМ, лк	420	0,875	2,625
5	Мінімальний розмір об’єкта, мм	360	0,75	1,5
6	Виробничий шум, дБА	420	0,875	0,875
7	Інтенсивність теплового випромінювання, Вт/м²	420	0,875	0,875
8	Токс. речовина, кратність перевищення ГДК	420	0,875	0,875
9	Виробничий пил (паперовий), кратність перевищення ГДК	420	0,875	0,875
10	Робоче місце (РМ), поза та переміщення у просторі	420	0,875	0,875
11	Кількість важливих об’єктів спостереження	420	0,875	0,875
12	Тривалість зосередженого спостереження , у % часу зміни	384	0,8	3,2
13	Тривалість повторюваних операцій, с	384	0,8	0,8

Кінець таблиці 1.3

№ з/п	Фактор умов праці на робочому місці	Тривалість дії фактора t_i	Тривалість дії фактора у долях робочої зміни, $t_{пит\ i}$	Фактична оцінка питомої ваги фактора $x_{фi}$
14	Змінність роботи	480	1	1
15	Тривалість безперервної роботи за добу, годин	420	0,875	1,75
16	Режим праці та відпочинку	480	1	1
17	Нервово-емоційне навантаження	420	0,875	3,5
18	Кількість рухів пальців на годину	384	0,8	3,2
19	Монотонність, тривалість операцій, які повторюються , с	384	0,8	3,2

За даними таблиці 1.3, визначимо елемент x_{max}, який набрав найбільшу оцінку. Таким елементом є елемент x_{17}, який пов'язаний із нервово-емоційним навантаженням на робочому місці.

Розрахуємо середнє арифметичне фактичних оцінок питомої ваги факторів (всіх, окрім визначаючого елемента) за формулою

$$\bar{x} = \frac{\sum_{i=1}^{n-1} x_{ф_i}}{n-1}, \quad (1.1)$$

де n = 19.

В даному випадку, $\bar{x}$ = 1,51.

Далі, розрахуємо інтегральну оцінку умов праці на робочому місці у відділі розробки програмного забезпечення за формулою [3]

$$I_n = 10 * (x_{max} + \bar{x}\frac{6-x_{max}}{6}). \quad (1.2)$$

В даному випадку, I_n = 41,29.

Порівнявши отримане значення зі значеннями, наведеними в Додатку Б, можна зробити висновок, що умови праці на визначеному робочому місці відносяться до III категорії – роботи, що відхиляються від ГДК і ГДР та допустимих рівнів психофізіологічних факторів.

Пропонується для деяких факторів, що отримали оцінку, вищу за 2, застосувати заходи для покращення умов праці, а саме:

– фактор 4 – освітленість на рм: встановлення більш потужних освітлювальних пристроїв;

– фактор 12 – тривалість зосередженого спостереження: запровадження іншого режиму роботи, з більшою кількістю (тривалістю) перерв;

– фактор 17 – нервово-емоційне навантаження: запровадження інших моделей менеджменту проєктів, з більш ретельним плануванням.

– фактори 18 та 19 (кількість рухів пальців на годину, та тривалість операцій, які повторюються), не може бути змінена, оскільки ці фактори напряму пов’язані зі специфікою роботи програміста [5].

Припустимо, що значення оцінок оптимізованих факторів дорівнюють 2. Обчислимо інтегральний показник важкості праці за формулою:

$$I_{n_2} = 19{,}7 * a - 1{,}6 * a^2, \quad (1.3)$$

де $a = \sum_{i=1}^{n} \frac{x_i}{n}$.

Для оптимізованих умов праці, інтегральний показник важкості праці дорівнює 25,16. Відповідно до даних, наведених у Додатку Д, цей показник відповідає категорії II – роботам, що виконуються в умовах, які відповідають гранично допустимим концентраціям (ГДК) і рівням (ГДР) санітарно-гігієнічних елементів, а також допустимим рівням психофізіологічних факторів [6].

Визначимо ступінь втоми працівників до та після прийнятих заходів за формулою 4.4:

$$B = \frac{I_n - 15,6}{0,64} \quad (1.4)$$

До прийнятих заходів, ступінь втоми працівників становив 40,14. Після прийнятих заходів, ступінь втоми працівників становив 14,94.

Розрахуємо також ступінь працездатності людини за формулою

$$П = 100 - В \quad (1.5)$$

До впровадження заходів з охорони праці, ступінь працездатності працівників становив 59,86. Після впровадження заходів з охорони праці, ступінь працездатності працівників становив вже 85,06. Вирахуємо зміну продуктивності праці за формулою

$$\Delta П = 0,2 * \left(\frac{П_2}{П_1} - 1\right) * 100 \quad (1.6)$$

Зміна продуктивності праці становить 8,42%.

Виконані розрахунки довели, що проведені заходи з охорони праці призвели до зменшення важкості праці з III до II категорії, і відповідно, зниженню втоми, підвищенню працездатності працівників відділу програмного забезпечення управління слабоформалізованими системами [7].

Крім того, вказані заходи можуть призвести до підвищення продуктивності праці співробітників відділу на 8,42 %.

ВИСНОВКИ

В ході виконання спеціальної частини з охорони праці до кваліфікаційної роботи бакалавра на тему «Desktop-гра жанру клікер на базі Unity» проаналізовано умови праці у аудиторії ЧНУ імені Петра Могили, де проводилася частина розробки та тестування розробленого застосунку.

Запропоновано впровадити такі заходи для покращення умов праці у відділі:

– встановлення більш потужних освітлювальних пристроїв;

– запровадження іншого режиму роботи, з більшою кількістю (тривалістю) перерв;

– запровадження інших моделей менеджменту проєктів, з більш ретельним плануванням.

За попередніми розрахунками, після впровадження вищенаведених заходів, інтегральний показник важкості праці дорівнюватиме 25,16. Відповідно до даних, наведених у Додатку Б, цей показник відповідатиме категорії II – роботам, що виконуються в умовах, які відповідають гранично допустимим концентраціям (ГДК) і рівням (ГДР) санітарно-гігієнічних елементів, а також допустимим рівням психофізіологічних факторів.

Запропоновані заходи також можуть привести до значного зниження ступеню втоми працівників та підвищення продуктивності їхньої роботи на 8,42%.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гетія І.Г., Леонтьєва І.М., Шумілін В.К. Методичні вказівки щодо проведення заняття з дисципліни «Безпека життєдіяльності» на тему: «Визначення інтегральної бальної оцінки тяжкості праці на робочому місці». М: МДАПІ, 2002 . 22 с.

2. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу //Охорона праці. – 2001. –№ 12. – С. 12-20.

3. Жидецький В.Ц. Основи охорони праці. Підручник. Львів: УАД, 2006. 336 с.

4. НПАОП 0.00-1.28-10. Правила охорони праці під час експлуатації електронно-обчислювальних машин.

5. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98. К.: Постанова Головного державного санітарного лікаря України, 1998. № 7.

6. Санітарні норми мікроклімату виробничих приміщень: ДСН 3.3.6.042-99. – К., 2000.- С.16

7. Природне і штучне освітлення : ДБН В.2.5-28:2006. К. : Міністерство будівництва, архітектури та житлово-комунального господарства України, 2006. 68 с. (Національні стандарти України).

ДОДАТОК А

Критерії бальної оцінки умов праці

Таблиця А.1 – Критерії оцінки умов праці

№ п/п	Фактор умов праці на робочому місці	Оцінка, бали					
		1	2	3	4	5	6
1	Температура повітря на робочому місці (РМ) у виробничому приміщенні, °С: теплий період, холодний період	23…25 21…23	26…28 18…20	29…32 15…17	33…35 12…14	35…37 Нижче +12	>37
2	Відносна вологість повітря на РМ, %	40…50	55…60	61…75	76…85	Понад 85	-
3	Швидкість руху повітря на РМ, м/с	Менше 0,2	0,2…0,5	0,6…0,7	0,8…1,2	1,3…1,7	Понад 1,7
4	Освітленість на РМ, лк	300	240…300	160…230	100…150	60…90	30…50
5	Мінімальний розмір об’єкта розпізнавання, мм	> 1,0	1…0,3	< 0,3	0,005…0,3	< 0,05	-
6	Виробничий шум, перевищення ГДР, дБА	< 1	Рівно ГДР	1…5	6…10	> 10	> 10 з вібрацією
7	Інтенсивність теплового випромінювання, Вт/м²	140	141..1000	1001…1500	1501…2000	2001…2500	>2500
8	Токсична речовина, озон, кратність перевищення ГДК	-	1	1…2,5	2,6…4,0	4,1…6	> 6,0
9	Виробничий пил (паперовий), кратність перевищення ГДК	-	1	1…5	6…10	11…30	> 30

Продовження таблиці А.1

№ п/п	Фактор умов праці на робочому місці	Оцінка, бали					
		1	2	3	4	5	6
10	Робоче місце (РМ), поза та переміщення у просторі	РМ стаціонарне, поза вільна, маса переміщене вантажу ≤ 5 кг	РМ стаціонарне, поза вільна, маса переміщене вантажу > 5 кг	Робоче місце стаціонарне, поза не вільна, до 25% часу зміни у нахиленому положенні до 30°	РМ стаціонарне, поза вимушена – до 50 % робочої зміни	РМ стаціонарне, поза вимушена, незручна – більше 50 % робочої зміни	РМ стаціонарне, поза вимушена, незручна, нахили під кутом до 60 град більше 300 разів за робочу змін
11	Кількість важливих об’єктів спостереження	Менше 5	5…10	11…25	Понад 25	-	-
12	Тривалість зосередженого спостереження, % часу зміни	Менше 25	25…50	51…75	76…85	86…90	Понад 90
13	Тривалість повторюваних операцій, с	Понад 100	31…100	20…30	10…19	5…9	1…4
14	Змінність роботи	Ранкова зміна	Дві зміни	Три зміни	Нерегулярні зміни	-	-

Кінець таблиці А.1

№ п/п	Фактор умов праці на робочому місці	Оцінка, бали					
		1	2	3	4	5	6
15	Тривалість безперервної роботи за добу, годин	-	< 8	< 12	> 12	-	-
16	Режим праці та відпочинку	Обґрунтований,	Обґрунтований,	Відсутність	-	-	-
		з включенням музики та гімнастики	, без включення музики та гімнастики	обґрунтованого режиму праці та відпочинку			
17	Нервово-емоційне навантаження	Прості дії за індивідуальним планом	Прості дії за заданим планом з можливістю корегування	Складні дії за заданим планом з можливістю корегування	Складні дії за заданим планом при дефіциті часу	Відповідальність за безпеку людей	Індивідуальний ризик
18	Кількість рухів пальців на годину	< 360	360…720	721…1080	1081…3000	> 3000	-
19	Монотонність, тривалість операцій, які повторюються , с	> 100	31…100	20…30	10…19	5…9	1…4

ДОДАТОК Б

Залежність категорії умов праці від величини інтегральної бальної оцінки

Таблиця Б.1 – Категорії умов праці

Діапазон інтегральної бальної оцінки	**Категорія умов праці**	**Характер роботи**
До 18	I	Роботи, що виконуються в оптимальних умовах
19…33	II	Роботи, що виконуються в умовах, які відповідають гранично допустимим концентраціям (ГДК) і рівням (ГДР) санітарно-гігієнічних елементів, а також допустимим рівням психофізіологічних факторів
34…45	III	Роботи, що відхиляються від ГДК і ГДР та допустимих рівнів психофізіологічних факторів
45,7…53,9	VI	Робота у несприятливих умовах праці
54…59	V	Роботи, що виконуються в екстремальних умовах
Понад 59	VI	Роботи, що виконуються в екстремальних умовах