

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко
підпис

«___» _____ 2022 р.


КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Мобільна гра-клікер на базі Unity

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408.21810821

Студент

 / Р. А. Парахненко
підпис

«__» _____ 2022 р.

Керівник канд. тех. наук, доцент

_____ Г. В. Горбань
підпис

«__» _____ 2022 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексеєва
підпис

«__» _____
2022 р.

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

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії
програмного забезпечення, канд.

техн. наук, доцент,

_____Є.О. Давиденко

« ____ » _____ 2022 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи бакалавра

Видано студенту групи 408 факультету комп'ютерних наук

Парахненко Ростиславу

_____ Андрійовичу

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

«Мобільна гра-клікер на базі Unity»_____.

Затверджена наказом по ЧНУ від «01» грудня _____ 2021 р. №
314_____

2. Строк представлення кваліфікаційної роботи « ____ » _____ 2022
р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні
Вхідні дані до роботи – функціональні та нефункціональні вимоги до програмного забезпечення мобільної гри-клікер. Результат – застосунок мобільної гри-клікер на базі Unity

4. Перелік питань, що підлягають розробці

- аналіз предметної області методологій розробки;
- аналіз аналогічних систем та на основі отриманих результатів побудувати діаграму використання та описати у вигляді сценаріїв із використанням шаблонів;
- створення сценаріїв використання системи
- побудова діаграми взаємодії між об'єктами, що відповідають потокам подій варіантів використання;
- проектування бази даних системи;
- розробка, тестування та розгортання застосунку.

5. Перелік графічних матеріалів:

Презентація.

6. Завдання до спеціальної частини

Дослідження питань охорони праці, які безпосередньо пов'язані з діяльністю розробника програмного забезпечення.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексеева 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

Парахненко Ростислав Андрійович
(прізвище, ім'я, по батькові студента)

(підпис)

Дата видачі завдання « ____ » _____ 2022 р.

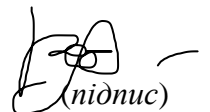
КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: «Мобільна гра-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	07.10.2021р.	08.10.2021р.	виконано
3.	Складання календарного плану КРБ	15.10.2021р.	15.10.2021р.	виконано
4.	Аналіз предметної області	03.11.2021р.	04.11.2021р.	виконано
5.	Моделювання та конструювання ПЗ	05.11.2021р.	10.11.2021р.	виконано
6.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	16.11.2021р.	22.02.2022р.	виконано
7.	Розробка спеціальної частини з охорони праці	14.05.2022р.	28.05.2022р.	виконано
8.	Відгук керівника КРБ	14.06.2022р.	14.06.2022р.	виконано
9.	Оформлення КРБ та презентації	08.02.2022р.	20.05.2022р.	виконано
10.	Попередній захист	20.05.2022р.	08.06.2022р.	виконано
11.	Рецензування	15.06.2022р.	15.06.2022р.	виконано
12.	Завершення оформлення КРБ та презентації	15.06.2022р.	21.06.2022р.	виконано
13.	Захист кваліфікаційної роботи	28.06.2022р.	28.06.2022р.	виконано

Розробив студент Парахненко Ростислав Андрійович

(прізвище, ім'я, по батькові студента)



(підпис)

«___» _____ 2022 р.

Керівник роботи канд. техн. наук, Доцент Горбань Гліб
Валентинович

(посада, прізвище, ім'я, по батькові)

(підпис)

«___» _____ 2022

Анотація

На кваліфікаційну роботу бакалавра

Тема: «Мобільна гра-клікер на базі Unity»

Автор: Парахненко Ростислав Андрійович

Науковий керівник: Горбань Гліб Валентинович

Захищена: « ____ » червня 2022 р.

Короткий зміст праці:

В даний час ринок мобільних ігор не стоїть на місці і активно розвивається. Якщо ще пару років тому цей ринок наздоганяв, то в 2016 році він став найприбутковішим ринком у всій світовій ігровій індустрії, випередивши ігри для персональних комп'ютерів, Mac і консолей. Щомісяця в App Store і Google Play з'являються десятки тисяч нових ігор.

Сьогодні більшість людей мають смартфон. Це досить доступно для покупки. Кожен має можливість грати в мобільні ігри.

Смартфони завжди під рукою. Якщо подивитись навколо, то можна побачити, що люди грають у мобільні ігри скрізь: у чергах, у маршрутках, на роботі, у школах на перервах. Ігри – це чудова можливість скоротити час, а в сумісності з компактністю смартфона це сприяє популярності мобільної платформи.

Одним з популярних жанрів є «клікер». Ігри цього жанру прості, вони не карають гравця, а підбадьорюють, навіть коли він не грає. Адже повертатися, коли на вас чекає винагорода, завжди приємніше. Клікери чудово скорочують час. Вони вже зайняли свою нішу на ринку мобільних ігор.

Метою кваліфікаційної роботи метою роботи є розробка кликера на платформі Unity, монетизація гри шляхом додавання до неї реклами та внутрішньоігрових покупок, а також розміщення програми на ринку ігрового застосунка Android.

Для досягнення поставленої мети у роботі необхідно вирішити наступні завдання:

- 1) Розглянути інструменти розробки програмного забезпечення для ігор для мобільних операційних систем;
- 2) Переглянути існуючі аналоги ігрового застосунку;
- 3) Розробити мобільний застосунок;
- 4) Реалізувати ігровий застосунок;
- 5) Протестувати ігровий застосунок.

Об'єктом кваліфікаційної роботи є процес моделювання прикладної системи, який допоможе користувачеві отримувати задоволення від гри та приносити прибуток розробникам.

Предмет кваліфікаційної роботи є технологія розробки мобільної гри-клікер на базі Unity.

КРБ викладена на 54 сторінки, вона містить 4 розділи, 22 ілюстрацій, 2 таблиці, 18 джерел в переліку посилань.

ABSTRACT
of the Bachelor's Thesis

Topic: «Mobile game-clicker on base Unity»

Student of group 408: Parahnenko Rostislav Andriyovich

Supervisor: Horban Hlib Valentionovich

Summary of work:

The topic Currently, the mobile games market does not stand still and is actively developing. If a couple of years ago this market was catching up, then in 2016 it became the most profitable market in the entire global gaming industry, overtaking games for personal computers, Mac and consoles. Every month, tens of thousands of new games appear in the App Store and Google Play app stores.

Nowadays, most people have a smartphone. It's affordable enough to buy. Everyone has the opportunity to play mobile games.

Smartphones are always at hand. If you look around, then you can see that people play mobile games everywhere: in queues, in minibuses, at work, in schools during breaks. Games are a great opportunity to pass the time, and in compatibility with the compactness of a smartphone, this contributes to the popularity of the mobile platform.

One of the popular genres is “clicker”. Games of this genre are simple, they do not punish the player, but encourage him even when he is not playing the game. After all, it is always more pleasant to return when a reward awaits you. Clickers are great for passing the time. They have already occupied their niche in the mobile games market.

The purpose of the qualification work is the aim of the work is to develop a clicker game on the Unity platform, monetize the game by adding advertising and in-game purchases to it, and also place the application on the Android gaming application site.

To achieve this goal in the work it is necessary to solve the following tasks:

- 1) To review software development tools for games for mobile operating systems;
- 2) To review the existing analogues of the gaming application;
- 3) Design a mobile application;
- 4) Implement a game application;
- 5) Test the game application.

The object of the qualification work is the process of modeling the application system, which will help the user to enjoy the game and bring profit to the developers.

The subject of the qualification work is the technology of developing Mobile clicker game based on Unity.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧОК.....	3
ВСТУП.....	4
1 ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ СТВОРЕННЯ ЗАСТОСУНКІВ.....	5
1.1 Основні поняття, методи та етапи розробки	5
1.2 Актуальність предметної сфери	9
1.3 Огляд аналогів	12
1.4 Технічне завдання	14
1.5 Специфікація вимог	15
Висновки до розділу 1	17
2 ОГЛЯД ЕТАПІВ СТВОРЕННЯ ЗАСТОСУНКУ.....	18
2.1 Основні вимоги до технічного забезпечення	18
2.2 Вибір фреймворку	21
2.4 Вибір програми для розробки дизайну та моделей застосунку	25
Висновки до розділу 2.....	28
3 ПРОЄКТУВАННЯ СИСТЕМИ	29
3.1 Структурно-функціональне моделювання реалізації застосунку.....	29
3.2 Моделювання діаграми варіантів застосування	33
3.3 Розробка макету інтерфейсу застосунку	35
Висновки до розділу 3	36
4 РОЗРОБКА КОМПОНЕНТІВ ЗАСТОСУНКУ	37
4.1 Файлова структура	37
4.2 Опис алгоритмів	37
4.3 Реалізація користувацького інтерфейсу	40
4.4 Функціонал застосунку	41
Висновки до розділу 4	46
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	48
ДОДАТОК А. HireUpdate	50
ДОДАТОК Б. CollectUpdate.....	54

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧОК

ПК – персональний комп'ютер

МП – мова програмування

ОС – операційна система

CMS – content management system

HMVC – hierarchical model-view-controller

MVP - Minimum viable product

GPU - graphics processing

API - Application Programming Interface

VR – virtual reality

AR – augmented reality

ВСТУП

На даний момент ринок мобільних ігор не застоюється і активно розвивається. Якщо ринок наздогнав кілька років тому, то в 2016 році це був найприбутковіший ринок у світовій ігровій індустрії, випереджаючи ігри на ПК, Mac і консолях. Щомісяця в App Store і Google Play з'являються десятки тисяч нових ігор.

Смартфони завжди в наявності. Якщо озирнутися, ви побачите людей, які грають у мобільні ігри скрізь: у черзі, у маршрутці, на роботі, під час шкільних канікул. Ігри — чудовий спосіб скоротити час, а в поєднанні з компактністю смартфонів це сприяє популярності мобільних платформ.

Об'єкт роботи: процес моделювання прикладної системи, який допоможе користувачеві отримувати задоволення від гри та приносити прибуток розробникам.

Предмет роботи: технологія розробки мобільної гри-клікер на базі Unity.

Метою кваліфікаційної роботи метою роботи є розробка клікера на платформі Unity, монетизація гри шляхом додавання до неї реклами та внутрішньоігрових покупок, а також розміщення програми на ринку ігрового застосунка Android.

Для досягнення поставленої мети у роботі необхідно вирішити наступні завдання:

- 1) Розглянути інструменти розробки програмного забезпечення для ігор для мобільних операційних систем;
- 2) Переглянути існуючі аналоги ігрового застосунку;
- 3) Розробити мобільний застосунок;
- 4) Реалізувати ігровий застосунок;
- 5) Протестувати ігровий застосунок.

1 ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ СТВОРЕННЯ ЗАСТОСУНКІВ

1.1 Основні поняття, методи та етапи розробки

Проаналізуємо основні технології, що використовуються при розробці програм для мобільних телефонів. Першою технологією є Java 2 Micro Edition (J2ME). Це набір специфікацій та технологій, призначених для різних типів портативних пристроїв. Існує дві основні області: конфігурація підключених пристроїв (CDC) та конфігурація підключених обмежених пристроїв (CLDC). Напрямок визначає тип конфігурації центральних бібліотек Java, а також параметри віртуальної машини Java (в якій будуть використовуватись програми). Пристрої, що використовують технологію CDC, будуть досконалішими, наприклад, комунікатори. До пристроїв CLDC відносяться звичайні мобільні телефони, які мають скромніші можливості. Спеціальні режими дають змогу визначити функціональність конфігурацій для різних типів пристроїв. Режим профілю мобільного інформаційного пристрою (MIDP) призначено для портативних пристроїв CLDC з можливістю зв'язку. Режим MIDP визначає функціональність - інтерфейс користувача, збереження налаштувань, мережеву і прикладну модель. CLDC та MIDP становлять основу реалізації J2ME [1]. Код Java інтерпретується безпосередньо самим пристроєм за допомогою так званої віртуальної машини Java. Цей механізм дозволяє вільно розповсюджувати Java-програми, оскільки вони працюють на всіх пристроях з аналогічною платформою Java[1].

Наступною популярною сьогодні технологією є технологія Qt. Вона в основному використовується як кросплатформне середовище, що дозволяє використовувати програми, написані за його допомогою, на різних пристроях та операційних системах, включаючи Windows, Mac OS X, Linux, Symbian, Android та інші [1]. Починаючи з версії Qt 4.0, можна програмувати для

мобільних пристроїв. Зі зростанням бази користувача Qt зростає потреба в 17 вбудованих, мобільних застосунках і розробниках інтерфейсу користувача. Qt — одна з найуспішніших бібліотек для C++. Налагодження програм, розроблених для мобільних пристроїв, здійснюється за допомогою емулятора, що міститься у середовищі розробки. Таким чином, можна писати складні програми для мобільних пристроїв, використовуючи бібліотеки C++ та підтримку платформи. В даний час останньою версією є бета-версія Qt 5. Для запуску Qt на мобільних пристроях необхідно встановити певний фреймворк [1].

Дуже рідко, але все ж таки можливо в мобільній розробці зустріти використання таких мов як Python, C# і Unity, особливо після виходу нових фреймворків та бібліотек для них, адже за замовчуванням у цих мовах немає вбудованих інструментів для мобільних пристроїв. Наприклад, вихід фреймворку Kivy Python швидко просунув використання мови Python у мобільному програмуванні [2].

Головна особливість у розробці мобільних застосунків полягає у факторі самих пристроїв, під які розробляються програми, тому що це смартфони, планшети, які мають маленький або невеликий екран, або нестандартний (квадратний, прямокутний) екран. Крім того, якщо розглянути смарт-годинник, то це екран 2x2 см або взагалі круглий екран. Тому дуже важливим моментом створення мобільного застосунку є так званий User Experience, тобто взаємодія з користувачем [2].

Існує 2 методи розробки мобільних програм [2]:

- 1) Нативна розробка;
- 2) Кросплатформна, або гібридна технологія (ReactNative, Flutter, Xamarin).

Нативна технологія передбачає створення програми для мобільного пристрою конкретною мовою під конкретну платформу. Нативні програми

досить продуктивні і не мають обмежень у розробці (Java та Kotlin – для Android, а Swift – для iOS). До плюсів такої розробки можна віднести досить швидку реакцію на події користувача, можливість мати прямий доступ до апаратної частини та розробити найбільш звичний для користувача конкретної платформи інтерфейс. До недоліків можна віднести досить високу вартість розробки та підтримки і тривалий час, необхідний на розробку.

Розробляти один і той же застосунок під різні платформи (iOS/Android) довго і дорого, до того ж їх тестування також займе досить тривалий час (наприклад, компіляція оновлень для Андроїд займає близько 30 сек), тому якщо потрібно створити простий застосунок відразу для двох платформ, вдаються до гібридного способу розробки. Кросплатформна розробка проводиться за допомогою вебтехнологій - HTML, CSS і JavaScript - які дозволяють розробити програму відразу на кілька платформ. Але для того, щоб застосунок працював відповідно до своєї платформи, його потрібно «перекласти» на зрозумілу платформу мову, або додати проміжну ланку-перекладач. Розробляти один і той же застосунок під різні платформи (iOS/Android) довго і дорого, до того ж їх тестування також займе досить тривалий час (наприклад, компіляція оновлень для Андроїд займає близько 30 сек), тому якщо потрібно створити простий застосунок відразу для двох платформ, вдаються до гібридного способу розробки. Кросплатформна розробка проводиться за допомогою вебтехнологій - HTML, CSS і JavaScript - які дозволяють розробити програму відразу на кілька платформ. Але для того, щоб застосунок працював відповідно до своєї платформи, його потрібно «перекласти» на зрозумілу платформу мову, або додати проміжну ланку-перекладач [2].

Найбільш популярним варіантом для створення застосунків з доповненою реальністю є двигун від Unity3D - це міжплатформне середовище для розробки ігор і програм не тільки для мобільних пристроїв, але і для

Windows, Linux, консолей PlayStation, Xbox, Nintendo, і для всіляких VR-і AR-пристроїв.

Unity, хоч і є приватною розробкою компанії Unity Technologies, але доступний для розробників безкоштовно, хоч і з деякими обмеженнями (наприклад, власнику безкоштовної версії проект не повинен приносити більше 100 тис. доларів на рік) [2].

Unity надає API для доступу до засобів введення та налаштувань Android. Unity дозволяє викликати функції, написані на C/C++ безпосередньо зі скриптів (Java-функції можуть бути викликані непрямим способом) [3]. Для покращення продуктивності в Android-версії Unity динамічна типізація JavaScript завжди відключена (якби до кожного скрипту автоматично застосовувалася `#pragma strict`). Це важливо пам'ятати, якщо переносити на Android проект зі старших платформ, і якщо зіткнутись з помилками компіляції – проблему варто шукати насамперед у типізації. З такими помилками зазвичай легко впоратися, переконавшись, що типи всіх змінних явно вказані або наведені під час ініціалізації. Хоча Unity Android і підтримує DXT/PVRTC/ATC текстури, Unity розпакує текстури у формат RGB(A) при виконанні, якщо ці методи стиснення підтримуються на конкретному пристрої. Це може серйозно вплинути на продуктивність GPU, тому рекомендується використовувати формат ETC. Він є фактичним стандартом для Android, і має підтримуватись усіма сучасними пристроями. З іншого боку, ETC не підтримує альфа-канал, і іноді 16-бітові RGBA текстури будуть кращим вибором за критеріями якості, розміру та швидкості рендерингу там, де потрібний альфа-канал [3].

Однією з переваг Unity перед іншими інструментами є те, що вона дозволяє створювати ігри для шоломів віртуальної реальності. Віртуальна реальність - це високорозвинена форма комп'ютерного моделювання, що дозволяє користувачеві поринути в штучний світ і діяти безпосередньо в

ньому за допомогою спеціальних сенсорних пристроїв, пов'язувати його рухи з аудіовізуальними ефектами. Одночасно зорові, слухові, тактильні та рухові відчуття користувача замінюється їхньою комп'ютерною імітацією. Характерними рисами віртуальної реальності є: моделювання у реальному часі; імітація довкілля з високим ступенем реалістичності; здатність впливати на оточення та мати зворотний зв'язок [4].

1.2 Актуальність предметної сфери

Смартфони зробили наше життя набагато простіше та швидше. У міру розвитку технологій, що лежать в основі смартфонів, індустрія мобільних програм також постійно розвивається. З більш ніж 2,96 млн застосунків (і їх кількість зростає) у магазині Google Play, Android з моменту своєї появи набрав обертів і випередив конкурентів із часткою ринку понад 85%. З кожним роком Android продовжує пропонувати нові ідеї, інновації, методи та інструменти у розробці мобільних застосунків. Майбутнє розробки застосунків для Android дуже перспективні, що допоможе вам випередити своїх конкурентів.

Миттєві програми - не нова тенденція в розробці Android, але з тих пір вони набули популярності. Основна ідея миттєвих програм – дати користувачам можливість спробувати програми або ігри, не встановлюючи їх на пристрій. Миттєві програми утримують користувачів від завантаження програми та дозволяють використовувати їх швидко та прямо з Інтернету, навіть не встановлюючи їх. Миттєві програми не займають додаткового місця на смартфонах і зводять до мінімуму небажане вторгнення в систему. Суттєвими перевагами миттєвих програм є отримання повного доступу до програм Android без їх завантаження, більше місця для зберігання, відмінний дизайн UX/UI, сумісність з усіма пристроями Android та багато іншого [4].

У рекомендаціях Google з архітектури описуються передові методи створення програм для Android і пропонуються високопродуктивні та стабільні програми. Компоненти архітектури Android є гарною відправною точкою для створення більш надійних програм. За допомогою сучасних компонентів архітектури корпоративних мобільних застосунків розробники Android можуть створювати програми з меншою кількістю шаблонного коду та більше уваги приділяти коду. Архітектура застосунків Android допоможе спільноті розробників застосунків Android створювати стабільні програми та спростити шаблон простою для Android, такий як MVC, MVP або MVVM[4].

Розробка блокчейн-застосунків користується успіхом серед мобільних користувачів та розробників завдяки своїй безпеці та здійсненності. Індустрія мобільних застосунків помітила підйом у технології блокчейн, коли в Google Play Store було завантажено мільйони завантажень програми блокчейн-гаманця. Розробка блокчейн-застосунків стає останньою тенденцією у розробці застосунків для Android, оскільки до 2024 року ринок мобільних застосунків із блокчейном досягне 20 мільярдів доларів. З огляду на це підприємства активно інвестують у розробку блокчейнів, щоб максимізувати рентабельність інвестицій і отримати високий прибуток [4].

Майбутнє розробки мобільних програм для Android - це Flutter, нова технологія від Google. Є кілька причин, через які розробникам Android варто звернути увагу на Flutter у майбутньому. Flutter надає вам надійні власні засоби розробки програм для iOS та Android з єдиної кодової бази. Фреймворк Flutter – це мобільний SDK, що забезпечує реактивні уявлення без використання мосту JavaScript. Багато розробників вибирають Flutter через його гнучкість, сполучність, нативний дизайн, розробку MVP, простоту навчання, розробку гібридних застосунків та багато іншого. Такі програми, як Google Ads, Alibaba і Reflectly, які мають відмінний інтерфейс користувача і розширений набір функцій, розробляються з використанням Flutter [5].

Об'єднання Enterprise Mobile Management (EMM) та Application Performance Management (APM) – це два компоненти розробки корпоративних мобільних застосунків. EMM та APM використовуються для мінімізації повільності мобільного застосунку. З появою розробки мобільних програм він перетворився на засіб перевірки якості для тестування програм. Це не тільки допомагає забезпечити безпеку мобільних застосунків, а й дозволяє співробітникам ефективно обмінюватись даними через мобільні гаджети [5].

Використання штучного інтелекту в розробці мобільних програм не новина. Поєднання штучного інтелекту та машинного навчання (ML) справило значну революцію у розробці застосунків. Мобільні програми з підтримкою AI та ML дозволяють користувачам спростити повсякденну роботу по дому. Ці технології аналізують поведінку користувачів та їхню взаємодію з застосунком та роблять більш точні пропозиції. Найкращим прикладом платформ з підтримкою AI і ML є Netflix, який аналізує, які шоу дивляться користувачі, і рекомендує контент відповідно до їх уподобань. Декілька API-інтерфейсів можуть допомогти інтегрувати AI та ML у застосунок для Android, наприклад Face API, Text API та Barcode API [5].

Розробка програм для Android широко відома тим, що покращує якість обслуговування клієнтів. Додаток на запит — це тенденція розробки програм, що швидко розвивається, яка, ймовірно, визначатиме розвиток мобільних застосунків. Ці програми допомагають виконувати повсякденні завдання швидше та зручніше. Програми на запит служать для безлічі цілей, таких як доставка їжі, роздрібна торгівля, бакалія, краса, охорона здоров'я та інші. Вони можуть аналізувати розташування користувачів та пропонувати необхідні послуги. Клієнти можуть отримувати миттєві послуги з програмами на запит [5].

В даний час робота мобільних програм без хмарного простору стала трохи повільною. Таким чином, розробники найкращих рішень звертаються

до хмарних програм для Android, які полегшують завантаження важких даних. Хмарні програми для Android відіграють важливу роль у безпечному зберіганні значного обсягу даних, не займаючи пам'ять телефону. Користувачі можуть легко отримати доступ до хмарної програми Android з будь-якого місця та в будь-який час [5].

Дивлячись на згадані раніше останні тенденції в розробці Android, можна сказати, що ОС Android безпрецедентно рухається до уніфікованого досвіду, спрямованого на задоволення вимог клієнтів. Якщо бізнес з нетерпінням чекає на розробки застосунків для Android, то розгляд згаданих вище тенденцій, безсумнівно, допоможе максимізувати рентабельність інвестицій[4].

1.3 Огляд аналогів

У ході виконання дипломної роботи було проаналізовано кілька ігор жанру «Клікер» на майданчику Google Play. З великої різноманітності проєктів було виділено кілька популярних ігрових програм.

Tap Tycoon [6]



Рисунок 1.1 – Логотип гри Tap Tycoon

Гра в жанрі "клікер". У цій грі гравцеві необхідно від початку розвинути свій бізнес. При кожному кліку по екрану з неба починають сипатися гроші, що приносять перший дохід. Гравець може покращувати кількість одержуваних грошей за клік, або купувати будівлі, які забезпечать йому пасивний дохід. Гра на даний момент має понад 5 мільйонів завантажень.

Tap Titans [7]

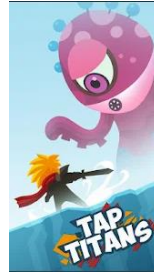


Рисунок 1.2 – Логотип гри Tap Titans

Гра у жанрі «клікер», де гравцеві необхідно боротися із титанами. Втрата противникам наноситься за допомогою кліків, після перемоги над титаном гравець отримує гроші, на які він може покращити свого персонажа, або купити напарників, які самостійно завдаватиме ударів титанам. У грі є боси, з якими гравець повинен впоратися за певний проміжок часу. Tap Titans має гарний візуальний стиль, на екрані постійно відбувається безліч візуальних ефектів, які розважають гравця в битві з титанами. Гра має на даний момент понад 10 мільйонів завантажень.

Make More! [8]



Рисунок 1.3 – Логотип гри Make More!

Гра в жанрі "клікер". Гравець стає власником фабрики, йому в підпорядкування виділяють працівників, які вручну виробляють продукцію. Гравець продає продукцію, за неї отримує прибуток, а на цей прибуток він може покращувати працівників, купувати нових, а також купувати нові фабрики, які приносять ще більше прибутку. Гра має більше 10 мільйонів завантажень.

Проведений аналіз дає зрозуміти, що ігри даного жанру мають велику популярність, безліч налаштувань і безліч позитивних рецензій. Отримана інформація дозволяє зрозуміти, які елементи є обов'язковими для даного жанру, що утримує людей у цих іграх. Це дозволяє правильно спроектувати та реалізувати ігровий застосунок у жанрі "Клікер". Планується спиратися на інтерфейс та ігрову механіку з першого. проаналізованої програми.

1.4 Технічне завдання

Необхідно спроектувати та реалізувати єдиний користувальницький ігровий застосунок, розрахований на тривалий геймплей. Застосунок повинен відповідати наступним вимогам та надавати користувачеві такі можливості:

- 1) при запуску ігрової програми користувач повинен відразу бути залученим до ігрового процесу, головне меню у грі відсутнє;
- 2) користувачу повинен мати доступ до інформації про кількість ресурсів та притоку ресурсів за секунду;
- 3) всі функціональні кнопки доступу до дій з інтерфейсом користувача повинні розташовуватися в нижній частині екрана, так як, програма призначена для мобільних пристроїв, і це пов'язано зі зручністю їх використання у горизонтальній орієнтації;
- 4) основне ігрове поле має бути реалізовано в ізометричній проекції;
- 5) програма має функціонувати на операційній системі Android з версії не нижче 2.3.

В даний час на ринку є багато платформ для розробки ігор. Кожна з них має свої особливості. Для гри необхідно знайти платформу, яка задовольнить наступні вимоги:

- 1) можливість розробки для мобільних пристроїв;
- 2) можливість розробки 2D ігор;
- 3) зручність використання;

- 4) наявність якісної документації;
- 5) вільно розповсюджується.

Виходячи з цих вимог, було обрано програмний продукт Unity. Unity 5 – кросплатформний інструмент для розробки двовимірних та тривимірних застосунків та ігор, що працює під операційними системами Windows та OS X. Дозволяє розробляти під усі найвідоміші платформи, такі як PC, Linux, Mac, IOS, Android, Xbox One, PS4 і т.д. Unity має дуже простий інтерфейс, який розбитий на кілька вікон: Hierarchy, де є назви всіх об'єктів на сцені, які можна групувати; Scene, де можна розглянути ігрове поле під потрібним ракурсом; Inspector, де знаходяться всі властивості виділеного об'єкта та його компоненти; Project, де знаходяться всі матеріал проекту; Toolbar, де знаходиться меню із інструментами. Проект у Unity ділиться на сцени – окремі файли, що містять свій набір об'єктів, скриптів та налаштувань. Основним об'єктом ігрової логіки є ігровий об'єкт – сутність, що включає компоненти. Також Unity надає інтегровані послуги для залучення, утримання та монетизації гравців.

Переваги:

- 1) зручність використання та простота освоєння;
- 2) якісна документація;
- 3) велика спільнота розробників, які використовують Unity;
- 4) можливість налаштувати та доопрацювати середовище розробки під необхідний проект;
- 5) інтегровані послуги монетизації та аналітики;
- 6) кросплатформність.

1.5 Специфікація вимог

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

- 1) Не менше 1 ГБ оперативної пам'яті;

- 2) Роздільна здатність екрану - 720р;
- 3) Мінімальна щільність пікселів – 200 ppi;
- 4) Швидкість пам'яті на запис та читання - 60 і 120 МБ/с;

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Коректність програмного забезпечення - властивість безпомилкової реалізації необхідного алгоритму. Під коректністю розуміють властивості програми, що свідчать про відсутність у ній помилок, допущених розробником на різних етапах проектування (специфікації, проектування алгоритму та структур даних, кодування). Коректність самої програми розуміють стосовно цілей, поставлених перед її розробкою (тобто це відносна властивість) [9].

Стійкість — властивість здійснювати необхідне перетворення інформації за збереження вихідних рішень програми у межах допусків, встановлених специфікацією [9].

Відновлюваність — властивість програмного забезпечення, що характеризує можливість пристосовуватися до виявлення помилок та їх усунення [9].

Надійність можна представити сукупністю наступних характеристик:

- 1) цілісністю програмного засобу (здатністю його до захисту від відмов);
- 2) живучістю (здатністю до вхідного контролю даних та їх перевірки під час роботи);
- 3) завершеністю (бездефектністю готового програмного засобу, характеристикою якості його тестування);
- 4) працездатністю (здатністю програмного засобу відновлення своїх можливостей після збоїв).

Висновки до розділу 1

Мобільні застосунки представляють майбутнє інформаційних технологій. Щороку науково-технічний прогрес продовжує дивувати. Потужність та обчислювальна потужність сучасних мобільних пристроїв наздоганяють і місцями навіть перевершують комп'ютери та ноутбуки, та загальна компактність не залишає шансів стаціонарному ПК.

Було проаналізовано ігри аналогічного жанру і виявлено певні переваги, які просунули застосунок в топ жанру клікер. Однією з переваг стала кросплатформність та генерація рівнів. Слід також зазначити такі недоліки як: штучний інтелект, локалізація та кількість реклами.

2 ОГЛЯД ЕТАПІВ СТВОРЕННЯ ЗАСТОСУНКУ

2.1 Основні вимоги до технічного забезпечення

Після аналізу ігрових двигунів було прийнято рішення для розробки проєкту у Unity. Далі буде представлено більш детальний розбір його архітектури.

Відеоігри, створені на Unity, підтримують OpenGL та DirectX. Крім того, цей двигун використовує простий інтерфейс Drag&Drop. складається з різних вікон, що дозволяє вносити корективи ігри у самому редакторі [9].

У кожному проєкті Unity має бути як мінімум одна сцена - окремий файл, який включає ігрові об'єкти, скрипти і налаштування. У черзі об'єкти можуть містити різні набори компонентів, із якими взаємодіють скрипти. Крім того, кожен об'єкт має ім'я, шар, на якому він знаходиться, відображається і тег. Слід зазначити, що двигун припускає наявність кількох об'єктів з такою самою назвою.



Рисунок 2.1 – Структура проєкту на Unity

Важливо, щоб кожен об'єкт був компонентом Transform та зберігав інформацію про координати свого положення на сцені, повороті та розмір об'єкта за трьома осями. Ще одним важливим компонентом Unity є Mesh – рендер, який робить видиму об'єктну модель.

Редактор Unity також підтримує написання та редагування шейдерів, має компонент створення анімації. Слід зазначити, що анімацію також можливо попередньо створити в будь-якому 3D-редакторі, та імпортувати разом з моделлю. Крім того, моделі, скрипти, матеріали та звуки можуть бути упаковані у формат unityassets, після чого їх можна легко перенести. Цей формат використовується в Unity Asset Store, в якому розробники представляють свої активи і заробляють на цьому гроші.

Двигун Unity також підтримує C# JavaScript для написання сценаріїв а також кілька мов шейдерів зі спеціальною надбудовою Shader Lab.

Мова програмування C# - це об'єктно-орієнтована мова програмування спільного призначення, розроблений Microsoft спільно з платформою .Net. Вона відноситься до мов з C-подібним синтаксисом, та найбільше нагадує C++ і Java. Розроблено за допомогою C# різноманітне програмне забезпечення: веб-сайти, веб-застосунки, офісні програми, десктопні та мобільні програми, ігри тощо. Також слід зазначити, що гра на двигуні Unity підтримує цю мову програмування.

Можна виділити такі переваги цієї мови:

- 1) компонентно-орієнтований підхід до програмування, що сприяє гнучкості та повторне використання деяких частин програмного забезпечення;
- 2) мова претендує реальну об'єктну орієнтацію, тобто сутність претендує на роль об'єкта;
- 3) система набору тексту уніфікована;
- 4) мова більше орієнтована на безпеку коду, порівняно з C++;
- 5) розширене подієво-орієнтоване програмування.

Також можна виділити кілька недоліків C#:

- 1) продуктивність щодо низька;
- 2) мало нових концептуальних ідей;
- 3) складний синтаксис.

Microsoft Visual Studio – це інтегроване середовище розробки (IDE) створений компанією Microsoft. Цей продукт використовує платформи для розробки програмного забезпечення, такого як Windows API, Windows Presentation Foundation, Windows Forms та Microsoft Silverlight[10].

Visual Studio Tools – це безкоштовне розширення для Visual Studio. Це перетворює продукт Microsoft на дуже потужний інструмент для мультиплатформенної технології на Unity, тому що у двигуна немає редактору коду. Це розширення використовує функції редагування коду, налагодження та продуктивності Microsoft Visual Studio для створення скриптів на мові C#, а також їх налагодження з використанням потужних можливостей Visual Studio[11].

Крім того, Visual Studio Tools має дуже глибоку інтеграцію із редактором Unity, тому розробник витрачає менше кліків на завершення простих завдань. Це забезпечує підвищену продуктивність під час розробки ігри. Також можна виділити такі сильні сторони продукту:

- 1) використання технології автозаповнення IntelliSense, яка записує ім'я функції під час введення його початкових букв;
- 2) потужні можливості рефакторингу (зміна внутрішньої структури коду для полегшення його розуміння, але без зміни зовнішньої поведінки самої системи);
- 3) швидке налагодження для Unity;
- 4) широкий вибір налаштувань робочого середовища.

Blender — програмний пакет для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, рендерінгу, обробки відео. До версії 2.80 містив рушій Blender Game Engine для створення відеоігор. Пакет є вільним програмним забезпеченням та розповсюджується під ліцензією GNU GPL[12].

Особливостями пакету є малий розмір, висока швидкість рендерингу, наявність версій для багатьох операційних систем — FreeBSD, GNU/Linux, Mac OS X, SGI Irix 6.5, Sun Solaris 2.8 (sparc), Microsoft Windows, SkyOS, MorphOS та Pocket PC. Пакет має такі функції, як симуляція динаміки твердих тіл (Rigid Body), рідин (Liquid simulation) та м'яких тіл (Soft body), редагування матеріалів і геометрії за принципом вузлів (Nodes), велику кількість легко доступних розширень, написаних мовою Python[12].

У цьому програмному пакеті є функція прив'язування (Link) та прикріплення (Append) до зовнішніх файлів. Прив'язування дає змогу створити посилання на інший файл, не включаючи його до .blend файлу. Прикріплення ж дозволяє перенести з іншого файлу, зокрема й .blend, лише потрібні елементи, як-от окрему модель чи матеріал. При відкритті .blend файлу він показується в такому випадку як низка папок з підпапками та файлами [12].

2.2 Вибір фреймворку

Фреймворк - готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми. Відповідно — програмний фреймворк може містити в собі також допоміжні програми, деякі бібліотеки коду, скрипти та загалом все, що полегшує створення та поєднання різних компонентів великого програмного забезпечення чи швидке створення готового і не обов'язково об'ємного програмного продукту. Побудова кінцевого продукту відбувається, зазвичай, на базі єдиного API[13].

Одна з головних переваг, при використанні каркасних застосунків, полягає в тому, що такі програми мають стандартну структуру. Каркаси застосунків стали популярними з появою елементів інтерфейсу, які мали тенденцію до реалізації стандартної структури для застосунків. З їх

використанням стало набагато простіше створювати засоби для автоматичного створення графічних інтерфейсів, оскільки структура внутрішньої реалізації коду програми стала відома заздалегідь. Для забезпечення каркаса, зазвичай, використовують підходи об'єктно-орієнтованого програмування, наприклад, частини програми можуть успадковуватися від базових класів фреймворку[13].

Перед тим як визначитися, який із фреймворків буде використано у роботі, Unity або UnrealEngine, було проведено порівняльну характеристику. У табл. 2.1. наведено порівняльну характеристику фреймворків.

Таблиця 2.1 – порівняльна характеристика фреймворків

Критерій	Опис	Результат
Продуктивність	Середній показник FPS (Frames per second) при обробці однакових ігрових сцен	Продуктивність Unreal Engine, в середньому, вища на 30%
Вимоги до ресурсів комп'ютера	Необхідні для коректної роботи рушія, параметри процесора, оперативної пам'яті, графічного адаптера	У середньому, Unity3D вимагає на 60% менше системних ресурсів
Поріг входження	Проблема освоєння двигуна для початківців	Unreal Engine складніший для вивчення, ніж Unity3D

Кінець таблиці 2.1

Основна мова програмування	Мова, яка використовується для програмування ігрової логіки	Unity3D використовує мову C#, яка візуально простіше для початківців, порівняно з C++, використовуваним Unreal Engine. Однак, Unreal Engine має так само систему візуального програмування Blueprint, яка є більш наочною
Кількість платних та безкоштовних додаткових ресурсів	Обсяг платних та безкоштовних товарів, доступних у магазинах додаткового контенту рушіїв	Unity3D має 39263 безкоштовних та 3981 платних ресурсів в офіційному магазині, Unreal Engine – 3056 безкоштовних та 60 платних

У результаті, обидва двигуни є потужними інструментами розробки. Якість графіки у обох двигунів приблизно однакове. Unity3D має нижчий поріг входження для нових розробників, він менш вимогливий до ресурсів, його бібліотека додаткових ресурсів значно більша, а мова програмування C# має більш зрозумілу структуру, ніж C++, використовуваний Unreal Engine.

Після проведеного аналізу було зроблено вибір на користь Unity, який добре підійде для проектної діяльності, коли використання ресурсів комп'ютера критично, а також, коли не потрібно заглиблюватися в алгоритміку.

2.3 Вибір мови програмування

Спочатку відеоігри створювалися для персональних комп'ютерів, простих консолей, згодом мобільні версії стали пріоритетом розробників. За історію розвитку програмування людство створило понад 700 самостійних та унікальних мов програмування: від реально використовуваних до розроблених «for fun»[14].

На теперешній час існує три мови програмування, які часто можна зустріти при розробці мобільних ігор для Android [14]:

- 1) C++;
- 2) C#;
- 3) Java.

Вибір конкретної мови залежить від кваліфікації розробника та потреб проекту. C++ – доля досвідчених програмістів, оскільки вважається однією з найскладніших у освоєнні. Він універсальний, демонструє швидку роботу, активно підтримується спільнотою програмістів, тому затребуваний фахівцями [14].

Розробка C# враховувала недоліки попередників: C++, Delphi, Modula, Smalltalk. Мова відмінно підходить движку Unity, що пропонує набір необхідних інструментів та готові пресети для створення гри без глибоких знань програмування [14].

Java легко освоїти за наявності базових знань про принципи програмування та функціонування МП. Мова спочатку позбавлений прив'язки до певної платформи, тому вважається універсальним. Але деякі движки (наприклад, Unity) не працюють з Java через несумісність [14].

Між мовами програмування є основні фактори які розділяють їх між собою [14]:

- 1) можливість синхронізації – рідкісний проєкт пишеться однією мовою. Часто розробник використовує різні мови програмування, наприклад, використовує для обробки інформації C, створює логіку на Java, frontend опрацьовує на JavaScript
- 2) оптимізація – деякі мови самостійно збирають сміття у коді, інші вимагають участі програміста для успішного результату;
- 3) платформа – для Apple – Objective-C (у минулому) та Swift (в теперішньому), для Android – Java, Kotlin, Python та деякі інші. За наявності планів працювати лише з однією платформою, не варто розпорошуватися на невідповідні мови.

Найчастіше мобільні проєкти збираються з C# та JavaScript, що ідеально працюють разом. Спарка дозволяє проєктувати різноманітні жанри: головоломки, платформери, аркади, повноцінні RPG, шутери. Вибір C# пояснюють переваги:

- 1) широкий функціонал, який вирішує більшість завдань;
- 2) зрозуміла структура синтаксису, що спрощує освоєння мови;
- 3) сумісність із двигуном Unity (хітом мобільного геймдева);
- 4) наявність виняткової бібліотеки, попередньо завантаженої до бази.

Оскільки передбачається створення мобільного застосунку на Unity, найкращий вибір мови програмування остається C#.

2.4 Вибір програми для розробки дизайну та моделей застосунку

Візуальна складова всіх ігор – це моделі та дизайн, практично все, що ви бачите в грі, це фантазія художників та розробників.

Під час розробки застосунку потрібно також продумати дизайн, на який користувач буде звертати увагу на 90% всього часу використання застосунку.

Дизайн та моделі – той фактор, який відрізняє мобільні гри однакового жанру. Тому потрібно приділити достатньо уваги, щоб користувач зупинився саме на цьому стилі даного жанру мобільної гри.

Cinema 4D або скорочено C4D — програмний пакет для створення тривимірної графіки та анімації, розроблений Maxon. Cinema 4D є універсальною комплексною програмою для створення і редагування тривимірних ефектів і об'єктів [16].

Основна програма містить інструменти для моделювання, текстурювання, рендеру та анімації. Основою для створення об'єктів слугують примітиви на кшталт сфери чи площини, поділені на полігони. Об'єкти, як цілком, так за виділеними полігонами, можуть змінюватися базовими перетвореннями, такими як обертання, зміна розміру, та просунутими — скручування, тиснення, перетворення за формулою тощо. Програма надає ряд деформаторів і генераторів складних об'єктів, наприклад, ландшафтів. Особливістю Cinema 4D є інструмент «ніж» для ручного розділення більшого полігона на менші. Програма також дає змогу малювати полігональні стрічки, прив'язувати одні полігони до інших, перетворювати грані на дуги. Інструмент MoGraph дозволяє автоматично створити з базового об'єкта чи їх групи складний об'єкт перетвореннями на кшталт клонування чи симетричного копіювання [16].

Cinema 4D надає гнучку і систему створення матеріалів з параметричними шейдерами для швидкого визначення поверхні тривимірної моделі. Матеріали передбачають такі властивості, як текстура, відбиття світла, світіння, прозорість, рельєфне текстурювання тощо [16].

Багатопрохідний рендеринг забезпечує відтворення кольору, тіней, відображень, розмиття. Підтримується експорт тривимірних моделей програми, такі як Adobe Photoshop, Adobe After Effects, Final Cut Pro. Крім

основного рендера, Cinema 4D може працювати зі сторонніми рендерами, що вбудовуються до програми, або із зовнішніми [16].

Blender — програмний пакет для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, рендерінгу, після-обробки відео. Один із плюсів цього програмного пакету, те що він безкоштовний та надає художникам всі інструменти, що їм необхідні[12].

Характерними особливостями пакету Blender є його невеликий розмір та підтримка багатьох популярних операційних систем. Він підтримує роботу з багатьма геометричними примітивами — базовими полігональними моделями (куб, сфера, циліндр тощо), кривими Безьє, поверхнями NURBS, metaballs, векторними шрифтами. Шляхом їх перетину та зміни розташування й розмірів окремих полігонів створюються всі інші, складніші, об'єкти. Є функція малювання довільних кривих нарисним олівцем (Grease Pencil)[12].

Програмний пакет можна розширювати доповненнями, котрі упаковуються в архів .zip або являють собою файли .py (Python). Такі доповнення дозволяють, наприклад, генерувати складні моделі, додавати нові інструменти. Python використовується також як засіб імпорту/експорту файлів (наприклад COLLADA), автоматизації завдань [12].

Autodesk 3ds Max - потужна програма для 3D моделювання найбільше затребувана у промисловому виробництві, дизайні інтер'єрів, а також в ігровій та кіноіндустрії. Створення реалістичних моделей будь-якого рівня деталізації, побудова найскладніших масштабних сцен із тисячами об'єктів.

Оскільки із зазначеними інструментами моделювання, найкраще для створення моделей та дизайну гри підходить Blender. Він має більше переваг та має вбудований потужний рендер.

Висновки до розділу 2

У ході роботи над розділом 2 було розглянуто основні вимоги до технічного забезпечення. Було проаналізовано мови програмування та інструменти для створення фронтенду.

Було проаналізовано та обрано певний набір інструментарію для виконання проекту, а саме – мова програмування C#, рушій гри Unity3D та дизайну Blender.

Проаналізовано структуру проектів на Unity. Визначено що кожен об'єкт повинен бути компонентом Transform та зберігати інформацію про координати свого положення на сцені, повороті та розмір об'єкта за трьома осями.

У зв'язку зі сформованою метою та визначенням переліку завдань було обрано необхідні інструменти виконання майбутніх етапів розробки застосунку.

3 ПРОЄКТУВАННЯ СИСТЕМИ

3.1 Структурно-функціональне моделювання реалізації застосунку

Функціональна модель IDEF0 відображає функції та структуру системи, а також потоки матеріальних об'єктів та інформації, які перетворюються за допомогою цих функцій. Дана методологія означає створення системи ієрархічних діаграм, тобто одиничних описань її фрагментів. Спочатку створюється контекстна діаграма, тобто повний опис системи та взаємодії її з навколишнім середовищем. Далі проводиться декомпозиція системи, тобто розбиття її на окремі підсистеми та їх окремий опис. Після чого ці системи розбиваються на більш дрібніші, до досягнення потрібного рівня деталізації[17].

Кожна IDEF0-діаграма повинна складатися з блоків робіт та стрілок. Роботи – це функції або процеси, які мають чіткі результати та відбуваються протягом певного часу. Стрілки описують взаємодію робіт між собою або з зовнішнім світом.

Для створення даної моделі використовувався програмний продукт Erwin Process Modeler, який розроблений компанією Computer Associates. На рис. 3.1 зображена контекстна діаграма реалізації ігрового застосунку.

- 1) Ціль – розробити ігровий застосунок.
- 2) Точка зору – розробник.
- 3) Предметна область – ігровий застосунок.

Кафедра інженерії програмного забезпечення
Мобільна гра-клікер на базі Unity

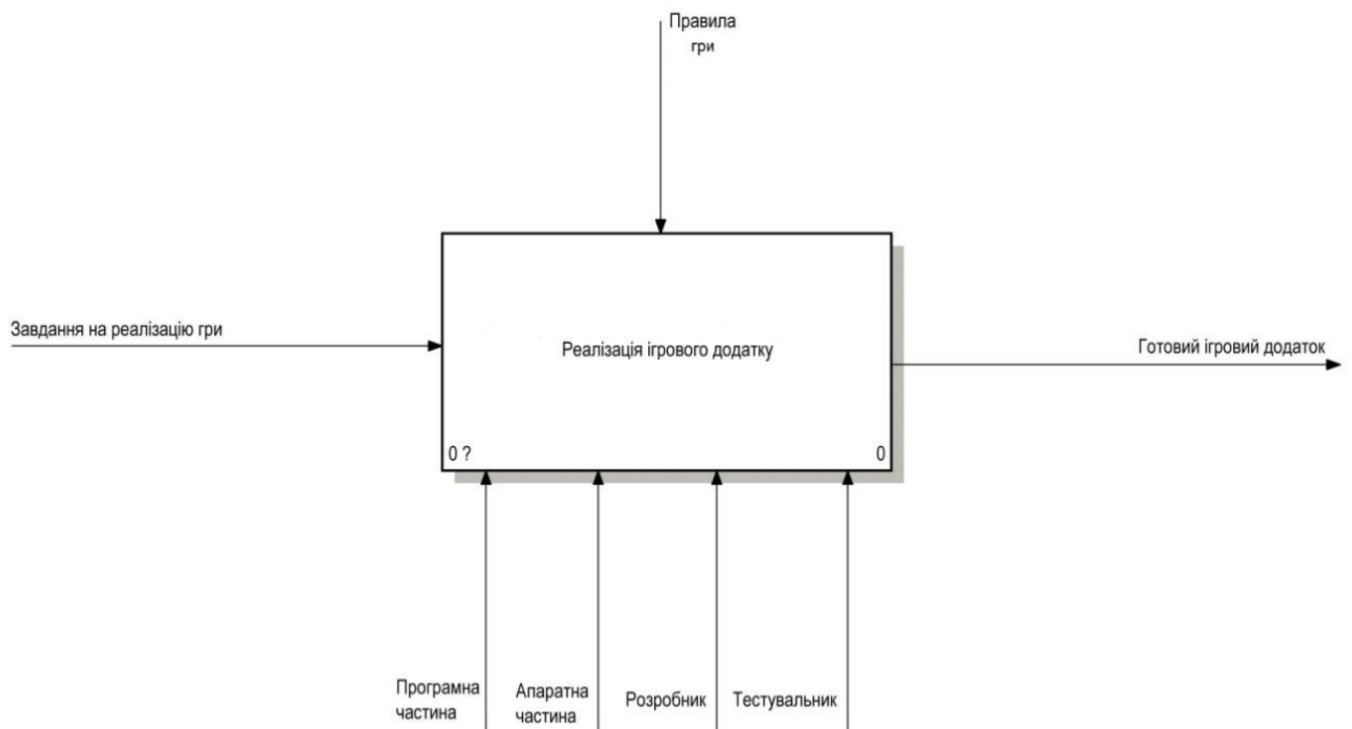


Рисунок 3.1 – Контекстна діаграма IDEF0

Процес декомпозиції був розбитий на чотири підрівнів, а саме:

Реалізація базових механічних дій, реалізація створення графіки або дизайну та анімації, реалізація проекту, реалізація тестування. Це обумовлено тим, що реалізація складності проекту поділяється на чотири етапи:

- 1) реалізація базових механічних функцій;
- 2) створення графіки;
- 3) реалізація проекту;
- 4) Виконання тестування.

На рис. 3.2 зображена декомпозиція діаграми реалізації ігрового застосунку.

Кафедра інженерії програмного забезпечення
Мобільна гра-клікер на базі Unity

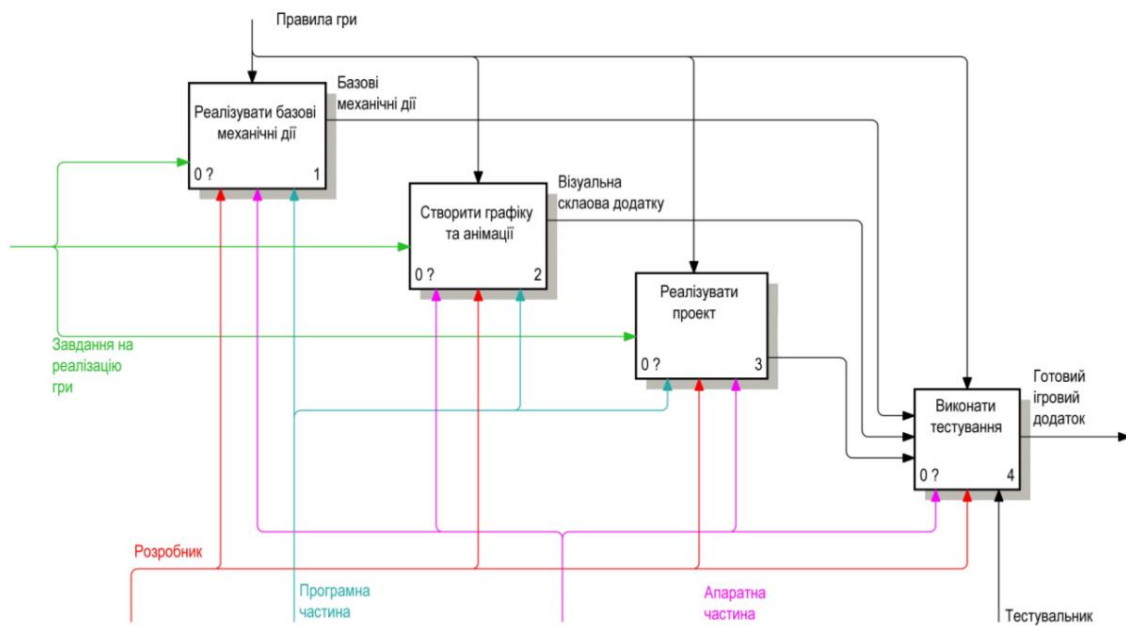


Рисунок 3.2 – Діаграма декомпозиції IDEF0

IDEF3 – це методологія моделювання, яка графічно описує інформаційні потоки взаємин між процесами, що обробляють інформацію, та об'єктами, які є їх частиною. Дана методологія дає можливість описати ситуацію, коли процеси виконуються послідовно, та описати об'єкти, що беруть участь в одному процесі. Кожна IDEF3 діаграма містить: роботи, зв'язки, об'єкти посилення та перехрестя [18].

Роботи виставлені у вигляді прямокутних блоків, кожен із яких має власний ідентифікаційний номер. Їх можна розкласти на більш докладний аналіз. Зв'язки показують відносини між роботами, всі вони існують односпрямований і може бути спрямований будь-куди. IDEF3 може існує три типи зв'язків: тимчасовий пріоритет, нечітке ставлення та об'єкт потік. Перехрестя використовується для відображення логіки руху потоки між творами, вони дозволяють вказати події, які мають чи можуть мати місце для виконання наступної роботи.

IDEF3 діаграму процесу розробки ігрового застосунку та її декомпозицію, можна побачити на рис. 3.3 – 3.4.



Рисунок 3.3 – Контекстна діаграма IDEF3

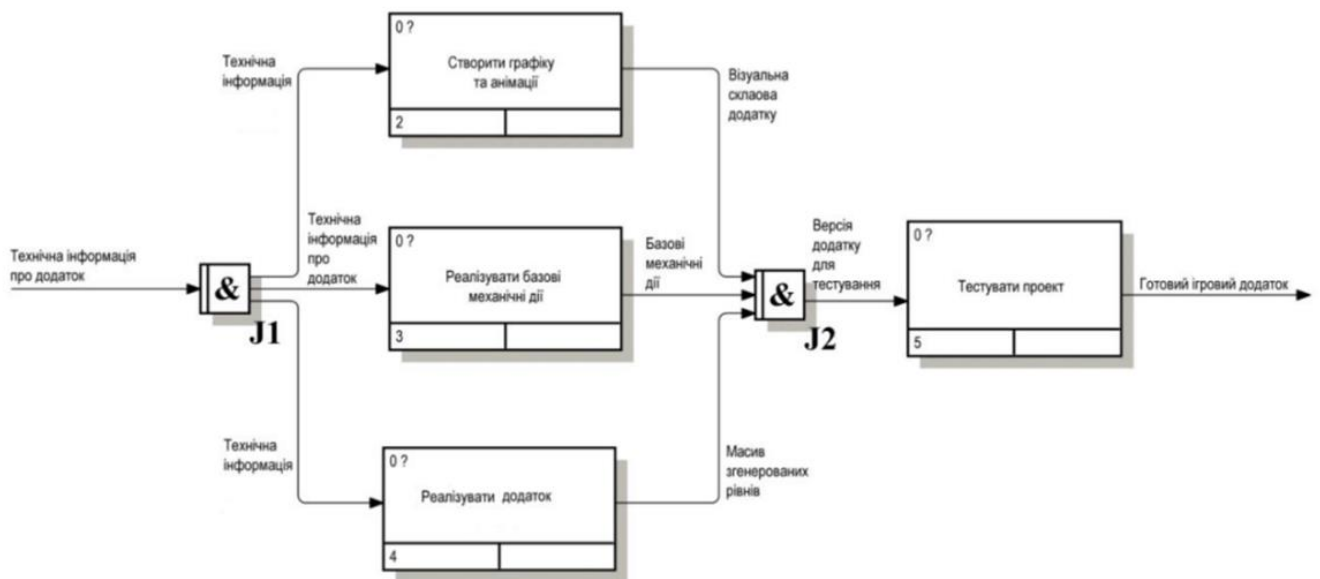


Рисунок 3.4 – Діаграма декомпозиції IDEF3

Процес тестування та відлагодження були поділені на два підрівні, а саме: тестування та відлагодження. Це пов'язано з тим, що процес пошуку помилок та їх. Виправлення поділено на два етапи:

- 1) Перший етап: пошук помилок.
- 2) Другий етап: виправлення знайдених помилок, якщо вони є.

При описі першого етапу були сформовані такі дані:

- 1) Вхідні дані: файли альфа-версії гри.
- 2) Вихідні дані: звіт про помилки, реліз гри.
- 3) Управління: вимоги до гри, планування розробки.

4) Механізми: команда проекту, програмні засоби, апаратне забезпечення.

Вигляд процесу «Тестування та відладка» після декомпозиції на 2 етапи наведений на рис. 3.5.

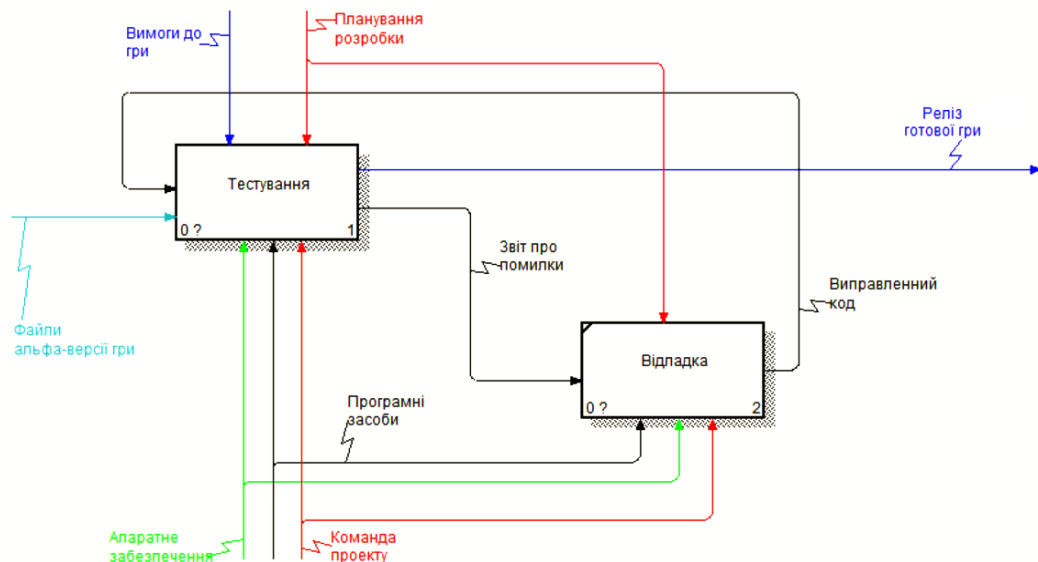


Рисунок 3.5 – Декомпозиція процесу «Тестування та відладка»

3.2 Моделювання діаграми варіантів застосування

Було побудовано діаграму варіантів використання, яку представлено на рис. 3.6. Єдиним актором системи є користувач. Варіанти використання ігрової програми:

- 1) отримати гроші за натискання - можливість отримати гроші за натискання на ігрове поле;
- 2) вікно досягнень – панель елементів, де описуються умови отримання досягнень та поточний результат гравця. У вікні досягнень є можливість забрати заслугу при досягненні умов;
- 3) забрати нагороду – отримати нагороду за виконані умови досягнення;

- 4) переключити панель покупок – можливість переключитись на іншу панель покупок для доступу до інших товарів;
- 5) здійснити покупку - можливість придбати покращення та здібності за ігрові гроші;
- 6) застосувати здатність – можливість застосування придбаних раніше здібностей збільшення доходу;
- 7) отримати бонус – можливість отримати бонуси за перегляд реклами;
- 8) меню установок – відкрити панель керування настройками гри;
- 9) вимкнути звук – вимкнути відтворення звуків у грі;
- 10) вимкнути музику – вимкнути відтворення музики у грі.

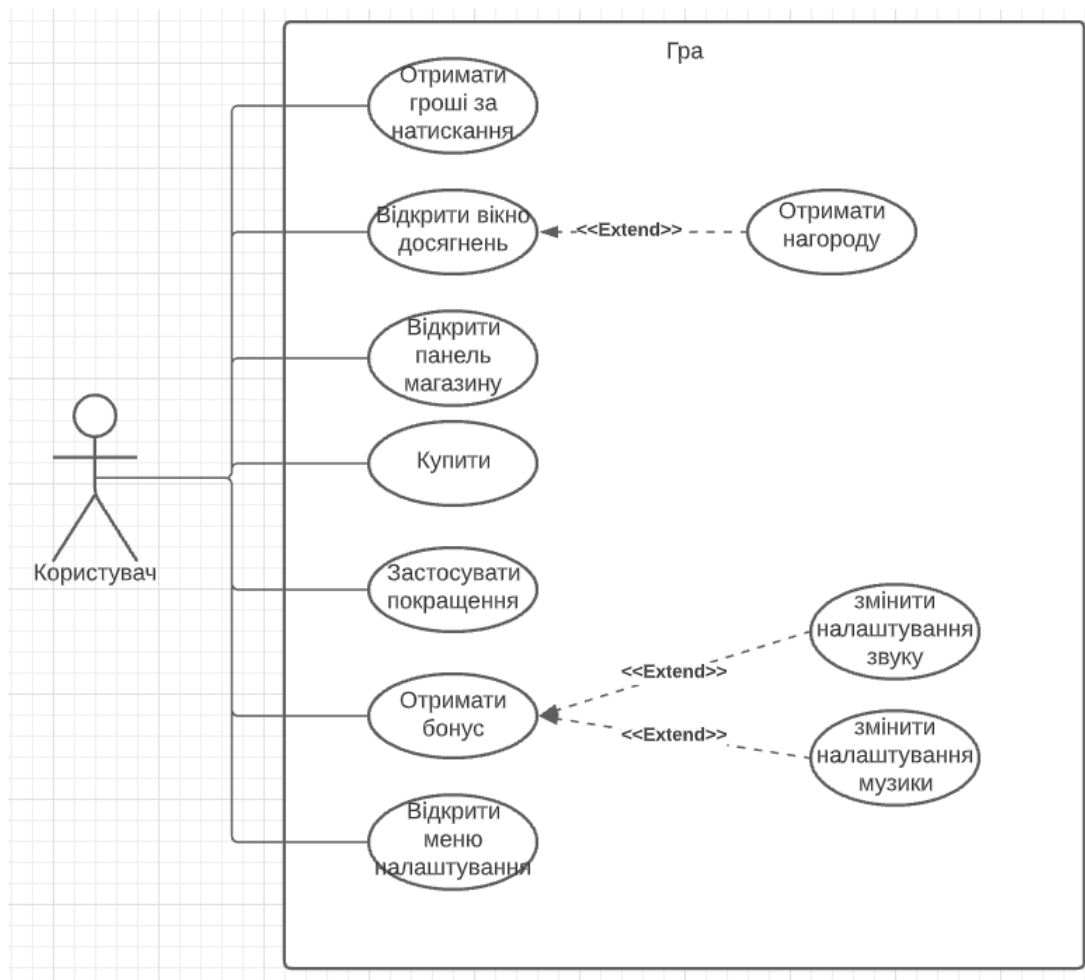


Рисунок 3.6 – Діаграма варіантів застосування

3.3 Розробка макету інтерфейсу застосунка

Було розроблено макет для візуалізації інтерфейсу системи. Це засіб занурення користувача у продукт та розподілу з ним сенсорних ланцюгів для виконання певних дій.

Основні частини інтерфейсу:

- 1) панель покупок, розташована нижче за середину екрана;
- 2) ігрове поле, розташоване у верхній половині екрана.

Нижче панелі покупок розміщуються три кнопки перемикання панелей покупок, кожна відповідає за відображення своєї панелі. При натисканні на кнопку отримання бонусу спливає панель з діалогом, який пропонує подивитися рекламу за нагороду. Між ігровим полем та панеллю покупок розташовується панель здібностей, які гравець може використовувати для збільшення доходу. При натисканні на ігрове поле гравець отримує певну кількість монет. Налаштування звуку відкривають панель, на якій з'являється можливість вимкнути звук або музику у грі.

Лічильник грошей відображає кількість монет, а також кількість кристалів у гравця. Натискання на кнопку панелі досягнень призводить до відкриття відповідної панелі, де гравець може отримати кристали за виконання певних вимог. У грі немає головного меню, що дозволяє відразу залучити гравця до ігрового процесу.



Рисунок 3.7 – Приклад ігрової сцени

Висновки до розділу 3

У ході реалізації розділу 3 були розроблені сценарії використання системи. Було розроблено контекстна діаграма IDEF0, діаграма декомпозиції IDEF0, контекстна діаграма IDEF3, діаграма декомпозиції IDEF3, декомпозиція процесу «Тестування та відлагодження», діаграма варіантів застосування та приклад ігрової сцени.

Було розглянуто структурно-функціональне моделювання реалізації застосунку, моделювання діаграми варіантів застосування, розробка макету інтерфейсу застосунку.

4 РОЗРОБКА КОМПОНЕНТІВ ЗАСТОСУНКУ

4.1 Файлова структура

Файлова структура (рис.4.1) – це схема розміщення його основних розділів. Це план-схема, яка показує, як будується застосунок, логічна зв'язка усіх елементів.

В результаті роботи над ігровим програмою була розроблена файлова структура.

У директорії **Plugins** містяться плагіни для роботи з внутрішньоігровим магазином та рекламою.

У директорії **Fonts** містяться текстові шрифти, що використовуються в ігровий застосунок.

У директорії **Prefabs** є шаблони ігрових об'єктів.

У директорії **Images** містяться зображення для всіх об'єктів на ігрове поле і анімація, а також зображення для інтерфейсу.

У директорії **Scripts** містяться скрипти, що використовуються в ігровому застосунок.

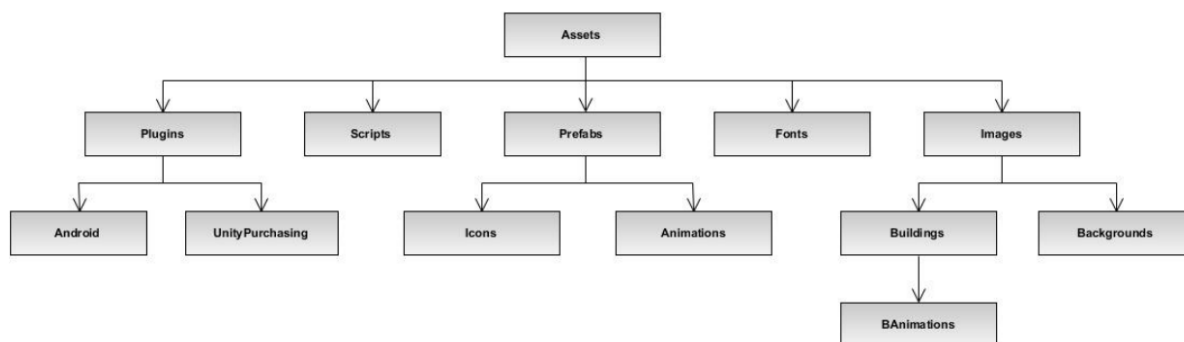


Рисунок 4.1 – Файлова структура

4.2 Опис алгоритмів

Один із найважливіших аспектів гри – отримання монет за натискання на ігрове поле. У цей момент запускається перевірка на включену здатність

«колекційна монета», яка дає шанс отримати збільшений у десять разів дохід від натискання.

Після запуску, гру необхідно повернути в той же стан, коли гравець залишав її. Для цього зчитується збереження, відбувається приховування вже придбаних товарів. Також вираховується дохід, який гравець отримав за час відсутності та відображається гравцю.

Слід зазначити, що для кожного проекту потрібен свій алгоритм залежно від бажаного результату та поставленої цілі, а кожен тип контенту вимагає свого власного підходу. Алгоритми наведено на рис. 4.2 та на рис. 4.3.

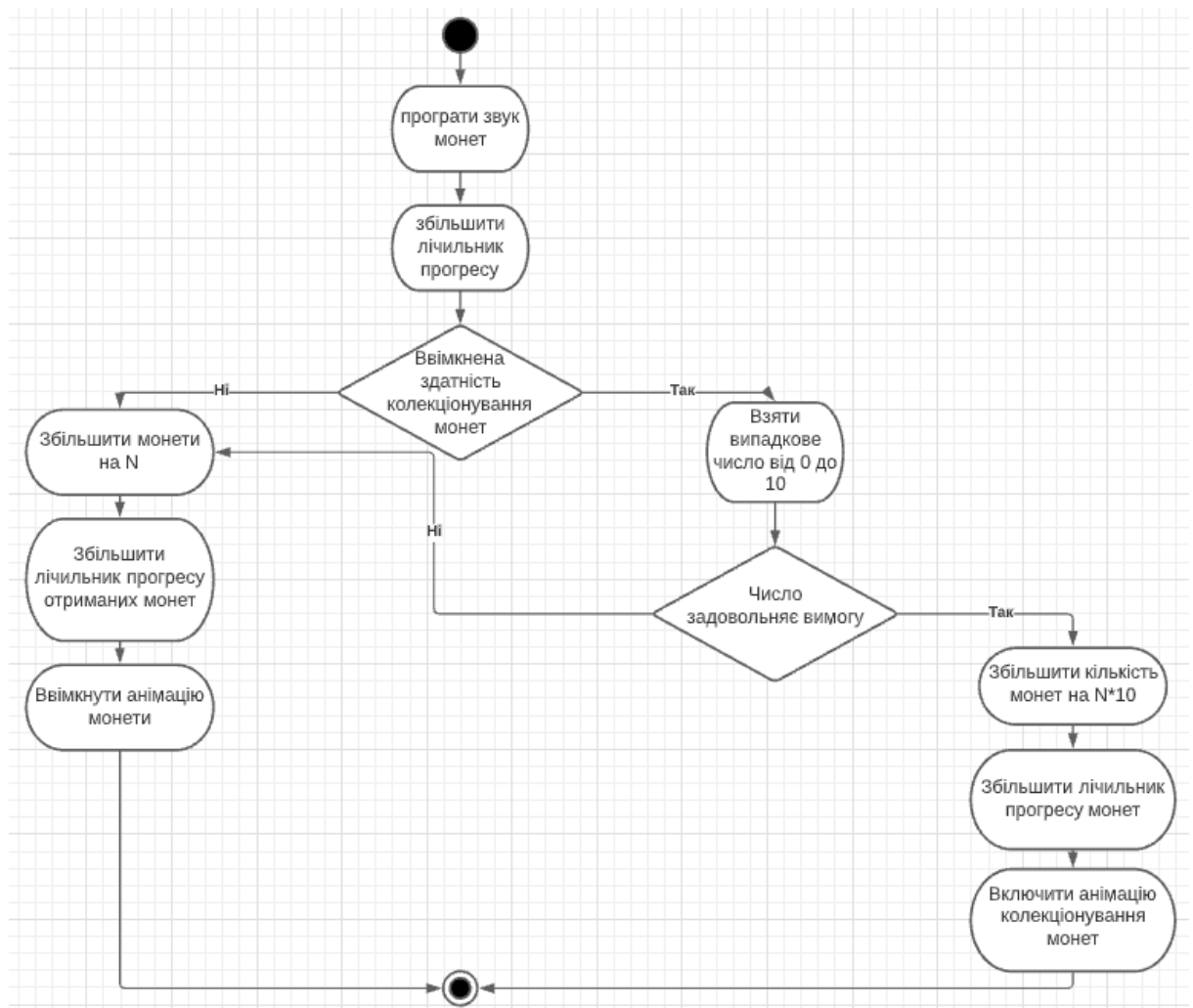


Рисунок 4.2 – блок-схема алгоритму після натискання на ігрове поле

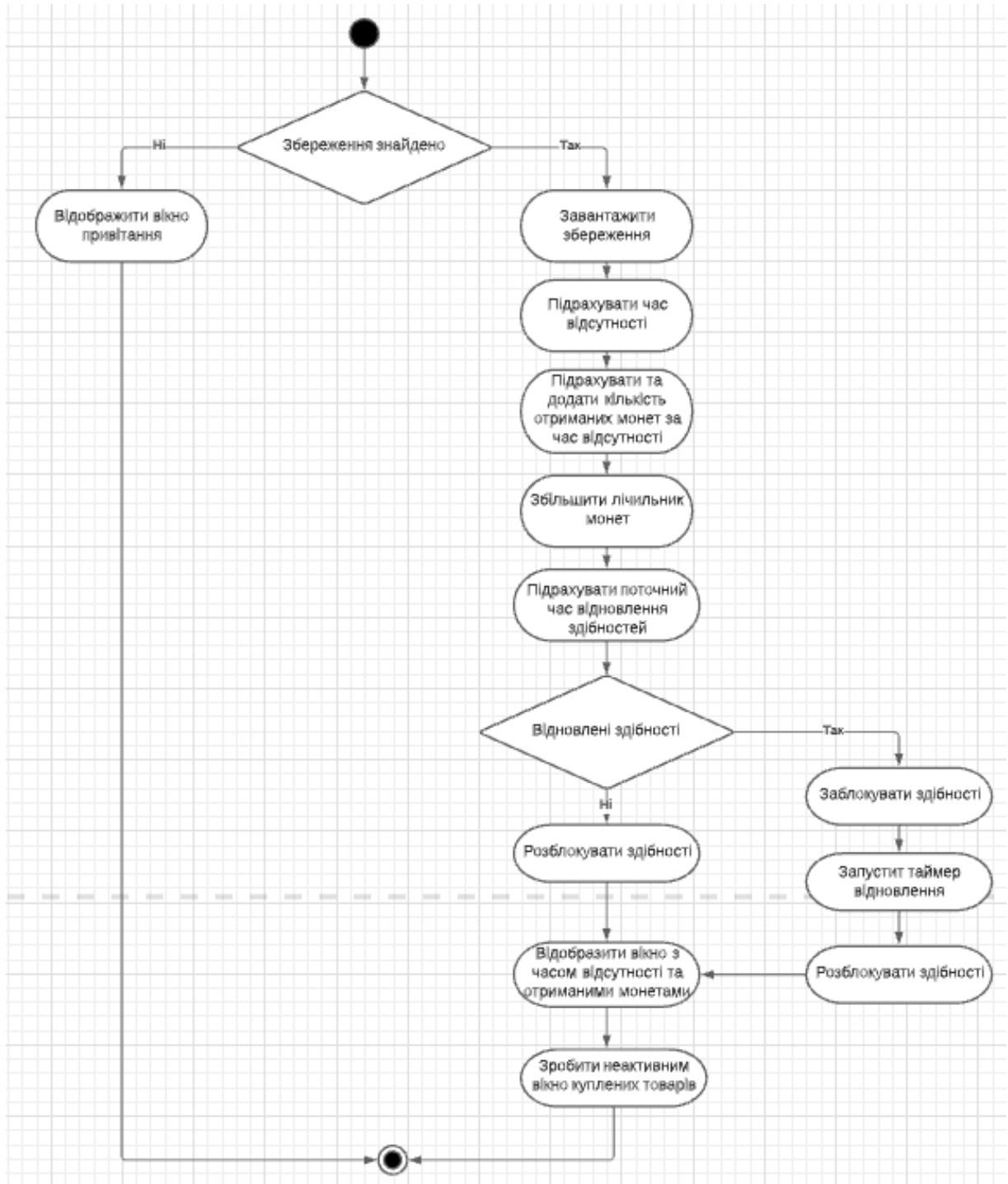


Рисунок 4.3 – блок-схема алгоритму після запуску гри

4.3 Реалізація користувацького інтерфейсу

У розробленому ігровому застосунку інтерфейс користувача реалізується за допомогою Unity UI. Ця система досить зручна у використанні вона має Drag&Drop інтерфейс, який дозволяє швидко створити необхідну кнопку, налаштувати її параметри, а також призначити необхідний скрипт і налаштувати дизайн.

В основі Unity UI лежать два компоненти:

- 1) Canvas – є контейнером для всіх елементів UI та визначає режим рендеру;
- 2) Rect Transform – цей компонент дозволяє задати розташування та розміри ігрового об'єкту;

Всі ігрові об'єкти є дочірніми по відношенню до Canvas. У розробленому ігровому застосунку було використано такі типи об'єктів:

- 1) UI.Text - використовувалися для зображення тексту в ігровій програмі. До тексту були використані шрифти з папки Fonts.
- 2) UI.Button – використовувалися для відкриття панелей та покупки покращень, також із цього об'єкта складається основне ігрове поле.
- 3) UI.Panel – ці об'єкти використовувалися для спливаючих вікон, а також для панелей покупок. На об'єктах Panel розташовані інші об'єкти інтерфейсу користувача.
- 4) UI.Image - використовувалися для розміщення спрайтів на ігровому полі.
- 5) UI.Slider – використовувалися створення смуги прогресу.

4.4 Функціонал застосунку

Було розроблено дизайн для кнопки за допомогою інструменту Blender. На рис. 4.4-4.7 зображено етапи розробки.

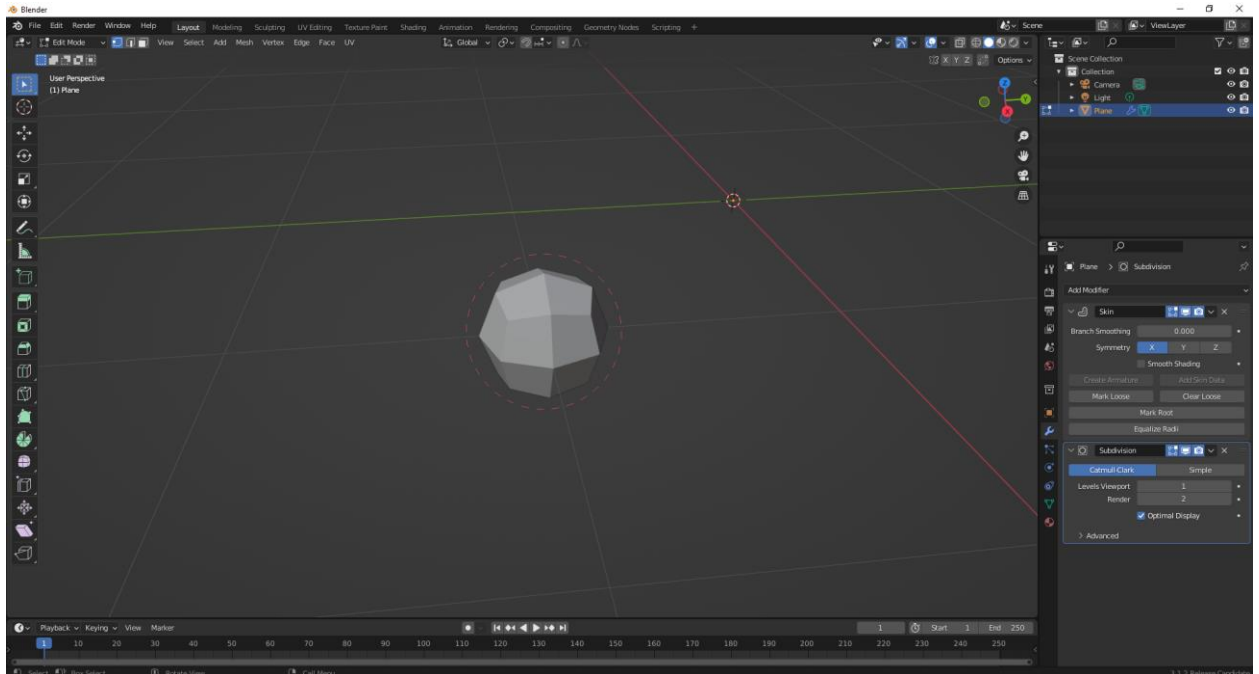


Рисунок 4.4 – Створення об'єкту деформації

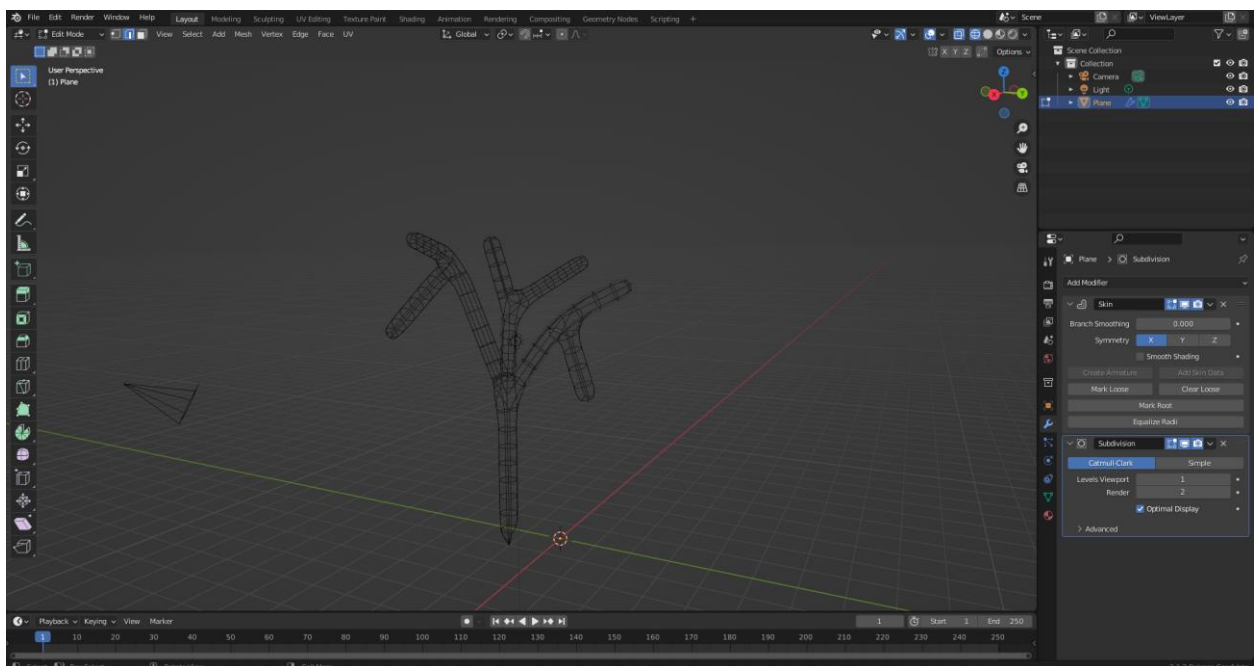


Рисунок 4.5 – Деформація полігонів

Кафедра інженерії програмного забезпечення Мобільна гра-клікер на базі Unity

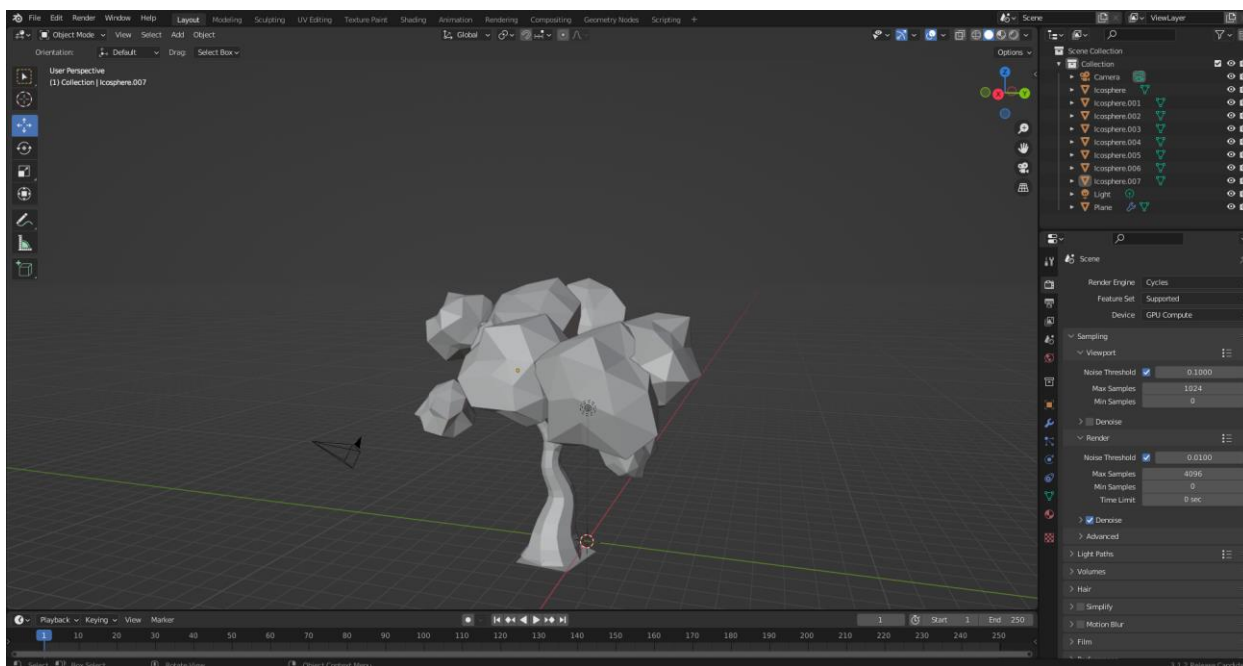


Рисунок 4.6 – Результат деформації полігонів

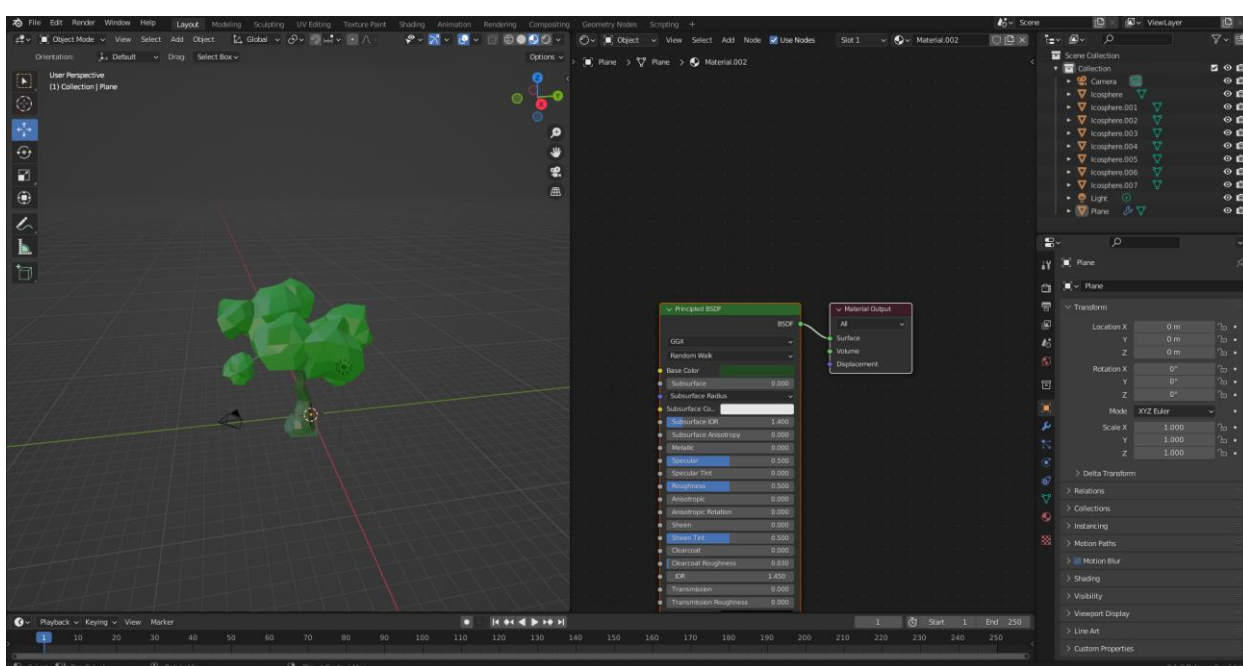


Рисунок 4.7 – Текстурування моделі

Модель було використано в якості ігрової кнопки. При натисканні на яку відбувається взаємодія між користувачем та застосунком. На рис. 4.8-4.11 продемонстровано кінцевий вигляд застосунку та результат взаємодії із застосунком.

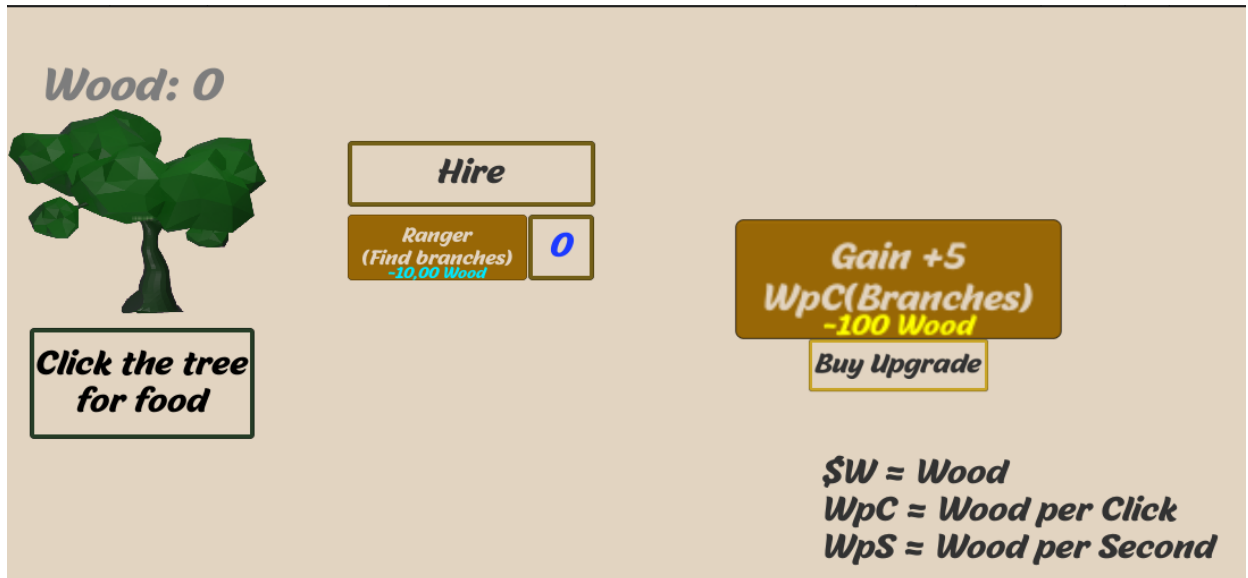


Рисунок 4.8 – Кінцевий дизайн застосунку

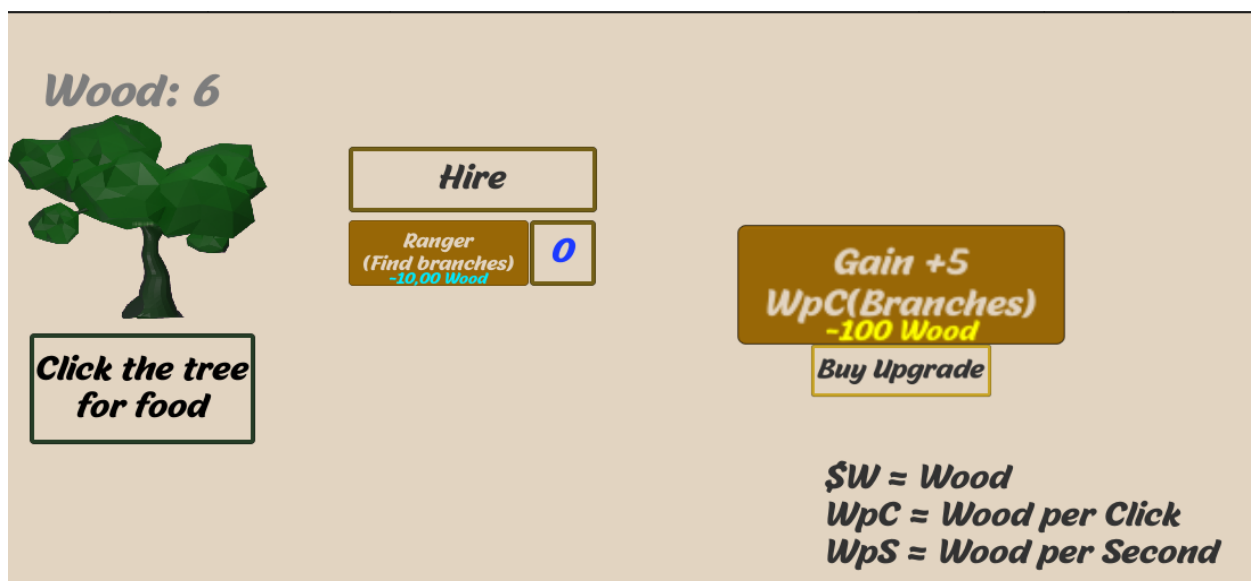


Рисунок 4.9 – Взаємодія із застосунком після натискання на кнопку

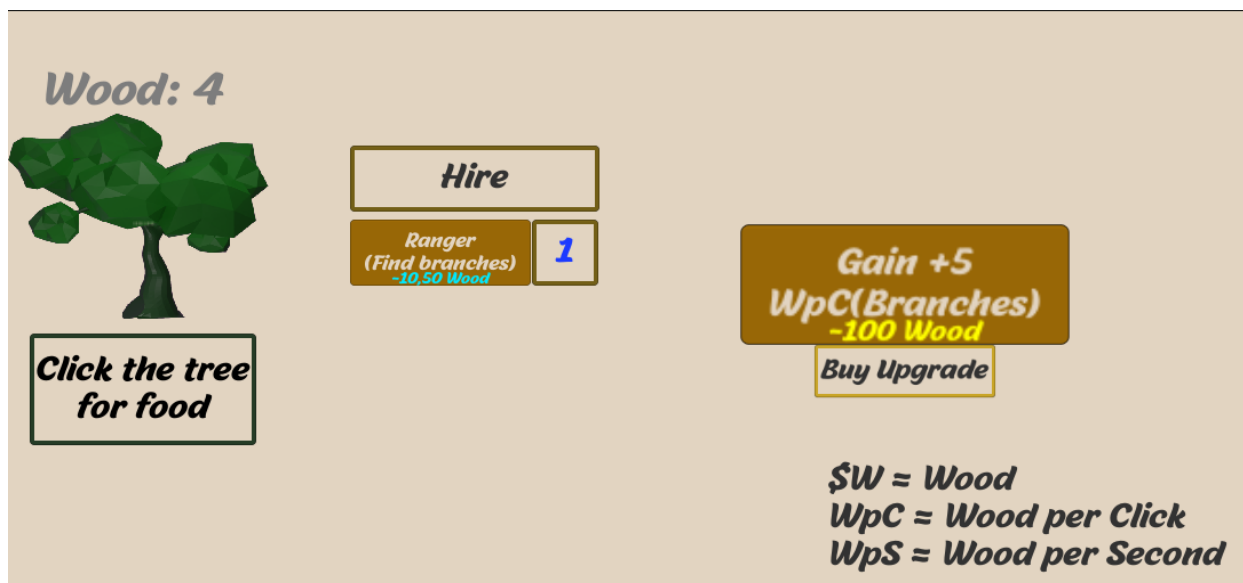


Рисунок 4.10 – Взаємодія із кнопкою магазину



Рисунок 4.11 – Взаємодія із кнопкою магазину

4.5 Тестування

Результати тестування наведені в табл. 4.1.

Таблиця 4.1 - Тестування

№	Назва тесту	Дія тестувальника	Очікуваний результат	Проходження тестування
1	Запуск застосунку	Натискання на запуск застосунку	Запуск застосунку	Так
2	Збільшення ігрової валюти	Натискання на кнопку у вигляді дерева	Збільшення ігрової валюти	Так
3	Покупка за ігрову валюту	Натискання на кнопку купівлі за ігрову валюту	Збільшення пасивного здобуття ігрової валюти	Так
4	Покупка за ігрову валюту	Натискання на кнопку купівлі за ігрову валюту	Збільшення здобуття ігрової валюти після натискання на кнопку у вигляді дерева	Так

Висновки до розділу 4

Під час виконання розділу 4 було розроблено файлову структуру, алгоритм гри, користувацький інтерфейс. Було протестовано функціонал застосунку.

Файлова структура містить чотири директорії з відповідним функціоналом. Цей функціонал будує логічний зв'язок застосунку.

Алгоритми наведені в двох блок-схемах, що описують структуру директорій і її функціонал.

Інтерфейс було реалізовано за допомогою інструмента UnityUI. Було використано елементи, що відповідають за певний функціонал для налаштування дизайну.

Функціонал застосунку було протестовано, та результати були наведені в таблиці.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра був проведений аналіз предметної області. В аналітичній частині були описані вимоги до розробки ігрового Android-застосунку і обґрунтований вибір програмного забезпечення для реалізації завдання. Завдяки аналізу засобів реалізації було обрано певний набір інструментарію для виконання проекту, а саме – мова програмування C#, рушій гри Unity3D та Blender в якості розробки дизайну.

Для планування робіт були створені діаграма контекстна діаграма IDEF0, контекстна діаграма IDEF3, діаграма варіантів застосування, макет ігрової схеми, блок-схеми, діаграма декомпозицій.

Для передбачення помилок при створенні Android-застосунку були побудовані контекстна діаграма IDEF0 та діаграма варіантів використання.

Процес реалізації застосунку був розділений на 3 етапи:

- 1) Етап підготовки перед розробкою застосунку, в якому визначалися інструменти та мова програмування;
- 2) Етап підготовки за допомогою діаграм та макету застосунку;
- 3) Етап реалізації програмного коду та скриптів і сцен за допомогою Unity.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методика розробки URL: <https://lektsii.net/2-50017.html>. (дата звернення: 10.06.2022)
2. Розробка мобільних додатків від А до Я URL: <https://dan-it.com.ua/blog/razrabotka-mobilnyh-prilozhenij-ot-a-do-ja-polnyj-gajd/> (дата звернення: 10.06.2022)
3. Мануал розробки під Android. URL: <https://docs.unity3d.com/ru/530/Manual/android-GettingStarted.html> (дата звернення: 10.06.2022)
4. Розробка додатків на UNITY під ос Android. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2017/06/125-1.pdf> (дата звернення 10.06.2022)
5. Актуальність розробки мобільних додатків під Android. URL: <https://htmler.ru/2020/12/29/aktualnost-razrabotki-mobilnyh-prilozhenij-pod-android/> (дата звернення: 10.06.2022)
6. Опис аналога Tap Tycoon. URL: <https://play.google.com/store/search?q=tap+tycoon&c=apps&hl=ru&gl=US> (дата звернення: 10.06.2022)
7. Опис аналога Tap Titans. URL: <https://play.google.com/store/search?q=tap+titans&c=apps&hl=ru&gl=US> (дата звернення 10.06.2022)
8. Опис аналога Make More!. URL: <https://play.google.com/store/search?q=make%20more!&c=apps&hl=ru&gl=US> (дата звернення 10.06.2022)
9. Лекції по Unity. URL: https://zinref.ru/000_uchebniki/02800_logika/011_lekcii_raznie_25/941.htm (дата звернення: 10.06.2022)

10. Windows Presentation Foundation. URL: <https://dic.academic.ru/dic.nsf/ruwiki/132037> (дата звернення: 11.06.2022)
11. Розробка простих консольних додатків за допомогою Microsoft Visual Studio. URL: <https://infopedia.su/13xa2dc.html> (дата звернення: 11.06.2022)
12. Blender – програмний пакет для створення тривимірної комп’ютерної графіки. URL: <https://uk.wikipedia.org/wiki/Blender> (дата звернення: 11.06.2022)
13. Програмний каркас, фреймворк. URL: https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9_%D0%BA%D0%B0%D1%80%D0%BA%D0%B0%D1%81 (дата звернення: 11.06.2022)
14. На яких мовах програмування пишуть мобільні ігри. URL: <https://vokigames.com/na-kakih-yazykah-programmirovaniya-pishut-mobilnye-igry/> (дата звернення: 13.06.2022)
15. Найкраща програма для 3D моделювання. URL: <https://videasmile.ru/post/897-luchshaya-programma-dlya-3d-modelirovaniya> (дата звернення: 11.06.2022)
16. Cinema4D – програмний пакет для створення тривимірної графіки. URL: https://uk.wikipedia.org/wiki/Cinema_4D (дата звернення: 11.06.2022)
17. Побудова контекстної діаграми. URL: https://studopedia.com.ua/1_162872_pobudova-kontekstnoi-diagrami.html (дата звернення: 12.06.2022)
18. Розробка інформаційної системи. URL: <https://www.webkursovik.ru/kartgotrab.asp?id=98038> (Дата звернення: 12.06.2022)

ДОДАТОК А. HireUpdate

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class HireUpdate : MonoBehaviour
{
    /*
     * All Hire Objects
     */

    /* Ranger */
    // GameObject for rangers
    public GameObject rangerGainOB;
    public GameObject rangerCostOB;
    // Script for rangerNumText and rangerCostText
    RangerNumText rangerNumScript;
    RangerCostText rangerCostScript;

    /*
     * WoodUpdate Object Reference
     */
    public GameObject woodUp;
    WoodUpdate woodUpScript;

    // Start is called before the first frame update
    void Start()
    {
```

```
/* Use Gameobject to initialise empty script objects */

//Ranger
rangerNumScript =
rangerGainOB.GetComponent<RangerNumText>();
rangerCostScript = rangerCostOB.GetComponent<RangerCostText>();

//Wood
woodUpScript = woodUp.GetComponent<WoodUpdate>();
}

private float nextActionTime = 0.0f;
public float period = 1.0f; // period = 1 second

void Update()
{
    // Update Automatic Wood Gain by Calling Function
    if (Time.time > nextActionTime)
    {
        nextActionTime += period;
        // Adds WoodperSecond every Second
        woodUpScript.UpdateWood(calculateWpS());
    }
}

//Function that gets all WpS from all "Hire"NumText
public double calculateWpS()
```

```

{
    // Note: Update this function as more Hired GameObjects added
    double total = rangerNumScript.GetRangerWpS();

    return total;
}

//Function that updates a hire based on input
public void Hiring(string hireMethod)
{

    //Update Wood when Hiring (Wood Cost)
    woodUpScript.UpdateWood(findHiringMethod(hireMethod));

}

//Function that finds hire method and hires
public double findHiringMethod(string hireM)
{
    // Ranger
    if (hireM == "Ranger" || hireM == "ranger")
    {
        double currentRangerCost = rangerCostScript.GetRangerCost();
        if (woodUpScript.enoughWood(currentRangerCost*-1))
        {
            //Update HireNum
            rangerNumScript.SetRangerNum(rangerNumScript.GetRangerNu
m() + 1);

```

```
//Update New Hiring Cost
rangerCostScript.SetRangerCost(currentrangerCost * 1.05);

return currentrangerCost;
}
}

return 0;
}

}
```

ДОДАТОК Б. CollectUpdate

```
using UnityEngine;

using UnityEngine.UI;

using System;

public class CollectUpdate : MonoBehaviour
{
    /*
     * All Collect Objects
     */

    /* Branch */
    //GameObject for Branch
    public GameObject branch;
    //create BranchText script object
    BranchText branchScript;

    /*
     * WoodUpdate Object Reference
     */
    public GameObject woodUp;
    WoodUpdate woodUpScript;

    // Start is called before the first frame update
```

```
void Start()
{
    //Use Gameobject to initialise empty script objects
    branchScript = branch.GetComponent<BranchText>();

    woodUpScript = woodUp.GetComponent<WoodUpdate>();
}

/*
 * Get called when collect-buttons are pressed
 */
public void updateCollectedWood(string collectMethod)
{
    woodUpScript.UpdateWood(getWoodChange(collectMethod));
}

/*
 * Find the correct Value that wood is changed by
 * based on passed string
 */
public double getWoodChange(string collectMethod)
```

```
{  
  
    // Change BranchPerSec  
    if (collectMethod == "Branch" || collectMethod == "branch")  
    {  
        return branchScript.GetBranch_WpClick();  
    }  
  
    return Int32.Parse(collectMethod);  
}  
  
}
```

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

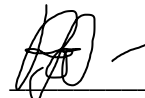
МОБІЛЬНА ГРА-КЛІКЕР НА БАЗІ UNITY

СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408.21810821

Студент



Р. А. Парахненко

підпис

«__» _____ 2022 р.

Консультант кан.тех.наук., доцент

_____ **Г. В. Горбань**

підпис

«__» _____ 2022 р.

Миколаїв – 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	3
ВСТУП	4
ОХОРОНА ПРАЦІ.....	5
1 Нормативна база охорони праці	5
2 Основні вимоги безпеки під час роботи з електронними пристроями.....	6
3 Мінімальні вимоги безпеки до екранних пристроїв.....	6
4 Загальні обов'язки роботодавців	8
5 Вимоги до приміщення в якому використовуються екранні пристрої	10
Висновок до розділу.....	14
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ	15

ПЕРЕЛІК СКОРОЧЕНЬ

МОЗ – міністерство охорони здоров'я

ДСН – державні санітарні норми

ДСанПН – державні санітарні правила та норми

МЮУ – Міністерство юстиції України

ЕОМ – електронна обчислювальна машина

ВДТ – візуально дисплейний термінал

ПП – приватне підприємство

ВСТУП

Сучасні люди проводять величезну кількість часу, спілкуючись за допомогою мобільних телефонів, але лише деякі розуміють, як саме вони працюють і як впливають на наш організм. Протягом багатьох років вчені намагаються відповісти на ці питання, але досі не було опубліковано жодного переконливого дослідження, яке б поставило крапку в цих дебатах. Одна з найголовніших проблем це мобільна залежність. Мобільна залежність впливає на психічний та фізичний стан людини. Йдеться не лише про зниження зору від постійного контакту з електронним дисплеєм. Неконтрольоване використання смартфона насамперед пов'язане з малорухливим способом життя — ігри в мережі не потребують жодних фізичних навантажень. Так під загрозою опиняється нормальна вага тіла залежного [1].

З іншого боку, працює ціла індустрія для забезпечення потреб людей залежних від мобільних ігор і в процесі розроблення таких мобільних застосунків, що забезпечується фахівцями здійснюється вплив на самих розробників застосунку. Тому актуальному є питання охорони праці працівників і у розділі присвяченому охорони праці, доцільно розглянуто вимоги до безпеки праці таких спеціалістів.

ОХОРОНА ПРАЦІ

1 Нормативна база охорони праці

В Основному Законі - Конституції України (ст. 43) зазначено: "Кожен має право на належні, безпечні й здорові умови праці, на заробітну плату, не нижчу від визначеної законом"; "Використання праці жінок і неповнолітніх на небезпечних для їхнього здоров'я роботах забороняється". "Кожен, хто працює, має право на відпочинок" (ст. 45). Це право забезпечується наданням днів щотижневого відпочинку, а також щорічної оплачуваної відпустки, встановленням скороченого робочого дня щодо окремих професій і виробництв, скороченої тривалості роботи в нічний час. Громадяни мають право на соціальний захист (ст. 46), що включає право на забезпечення їх у разі повної, часткової або тимчасової втрати працездатності, втрати годувальника, безробіття з незалежних від них обставин, а також у старості та інших випадках, передбачених законом[2].

Зазначені права реалізуються шляхом виконання вимог, викладених у Кодексі законів про працю, а також Законах: "Про охорону праці", "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності", "Про охорону здоров'я", "Про пожежну безпеку", "Про забезпечення санітарного та епідеміологічного благополуччя населення", "Про використання ядерної енергії та радіаційний захист", "Про охорону навколишнього природного середовища", "Про колективні договори і угоди", "Про дорожній рух", "Про поводження з радіоактивними відходами". Положення цих Законів конкретизуються у відповідних правилах, стандартах, нормах, інструкціях та інших нормативно-правових актах, перелік яких наведений в "Державному реєстрі нормативних актів з охорони праці"[2].

2 Основні вимоги безпеки під час роботи з електронними пристроями

Згідно з НПАОП 0.00-1.28-10 «Про затвердження правил охорони праці під час експлуатації електронно-обчислювальних машин»[4] до роботи з комп'ютерною технікою не слід допускати персонал, що не пройшли інструктаж та перевірку знань з охорони праці, пожежної безпеки та правил експлуатації ЕОМ.

Комп'ютерна техніка працює від мережі змінного струму 220В, шкідливий вплив якого повинна враховувати охорона праці. Згідно з ГОСТ 12.1.030-81 ССБТ «Електробезпека. Захисне заземлення. Занулення» [3] для запобігання виникнення небезпеки протікання струму через неструмоведучі металеві частини корпусу здійснюється підключення до пристрою захисного. Захисне заземлення – це з'єднання металевих не струмоведучих частин електроустановок із землею через заземлюючі провідники та заземлювачі для створення між цими частинами і землею малого опору. Захисне заземлення 42 включає у свої конструкції металічні труб діаметром 25-50 мм, довжиною 2-3 м, що вбито на 4-6 м одна від одної і з'єднано між собою металічною трубою, остання проходить в приміщення, з'єднується з внутрішнім контуром заземлення.

Також класифікують такі небезпечні фактори, які потрібно уникати: фізичні, хімічні, біологічні та психофізіологічні фактори.

3 Мінімальні вимоги безпеки до екранних пристроїв

При облаштуванні робочого місця екранними пристроями, працівнику необхідно вибирати таке обладнання, яке не створює зайвого шуму і не виділяє зайвого тепла та відповідає прилад наведеним далі вимогам[5]:

- Екранні пристрої не повинні бути джерелом ризику для працівників;

Кафедра інженерії програмного забезпечення
Мобільна гра-клікер на базі Unity

- Усе випромінювання, за винятком видимої частини електромагнітного спектра, має бути зведене до незначного рівня з погляду безпеки і охорони здоров'я працівників;
- Символи на екранних пристроях мають бути чіткими, відповідного розміру. Між символами і рядками символів має бути належна відстань;
- Зображення на екрані має бути стабільним, без миготінь або інших видів нестабільності;
- Яскравість та/або контрастність символів має легко регулюватися працівником під час роботи з екранними пристроями, а також швидко адаптуватися до навколишніх умов;
- Вибираючи екрани, слід надавати перевагу таким екранам, які легко та вільно повертаються і нахиляються відповідно до потреби працівника;
- Після закінчення роботи екранні пристрої слід відключати від електричної мережі;
- Щодня перед початком роботи необхідно очищати екранні пристрої від пилу та інших забруднень;
- Робочий стіл або робоча поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, допускати гнучкість під час розміщення екрана, клавіатури, документів і відповідного устаткування;
- Робоче крісло має бути стійким і дозволяти працівнику з екранними пристроями легко рухатися та займати зручне положення;
- Поверхня клавіатури має бути матовою, щоб уникнути віддзеркалювання. Розташування клавiш і самі клавiші мають полегшувати роботу із клавіатурою. Позначення клавiш повинно бути достатньо контрастним і розбірливим.

4 Загальні обов'язки роботодавців щодо забезпечення здорових та безпечних умов праці

Кожен роботодавець повинен виконувати наступні обов'язки[5]:

- Роботодавець зобов'язаний проінформувати про умови праці та наявність небезпечних та шкідливих виробничих факторів (хімічних, фізичних, біологічних, психофізичних), які виникають при роботі з екранними пристроями та що вони ще не усунені, а також можливі наслідки впливаючи на здоров'я працівників відповідно до вимог статті 5 Закону України «Про охорону праці»;
- Роботодавець першочергово забезпечує навчання і перевірку знань персоналу з охорони праці до початку роботи з екранними пристроями, а також у випадках модернізації та організації роботи обладнання;
- Роботодавець повинен вжити відповідних заходів, щоб забезпечити відповідність робочого місця працівника цим вимогам;
- При організації робочого місця працівника екранними пристроями важливо вибирати пристрої, які не створюють зайвого шуму і не перегріваються. Рівень шуму на робочих місцях людей, які працюють з екранним обладнанням, повинен відповідати виробничим вимогам шуму, ультразвуку та інфрачервоного випромінювання ДСН 3.3.6.037-99, затвердженим постановою Головного державного санітарного лікаря України від 1 грудня 1999 року №37;
- Роботодавець зобов'язаний за власний рахунок забезпечити проведення медичних оглядів відповідно до вимог працівників певних категорій, зазначених МОЗ України від 21 травня 2007 року №246, зареєстрованих в МІОУ 23 липня 2007 року №846/14113. На підставі результатів цих оглядів роботодавець повинен забезпечити виконання відповідних оздоровчих заходів у разі необхідності;

- У разі потреби роботодавець зобов'язаний провести лабораторні дослідження умов праці працівників для виявлення небезпечних і шкідливих факторів середовища, важкості та напруженості трудового процесу (виявлення ризиків, пов'язаних з порушенням зору, інвалідністю, стресом). Після цього роботодавець повинен вжити щодо усунення знайдених ризиків відповідно до статті 13 Закону України «Про охорону праці».

5 Вимоги до приміщення в якому використовуються екранні пристрої

Робочі місця, на яких працівники працюють і використовують персональні комп'ютери або ноутбуки, повинні відповідати низці вимог, які повинні перевірятися відповідними службами щодо дотримання вимог охорони праці та техніки безпеки. Нижче наведено деякі вимоги документа про правила охорони праці під час роботи на підприємствах[7]:

- Вимоги стосовно освітлення, оптимальних умов мікроклімату, ергономічних характеристик основних елементів робочого місця, рівнів шуму, вібрації, електромагнітного, ультрафіолетового та інфрачервоного випромінювання та електростатичного поля викладено у ДСанПіН 3.3.2-007-98;
- Виробничі об'єкти повинні відповідати проектній документації, затвердженій в установленому порядку;
- Електробезпека будівель та приміщень, де розміщені робочі місця операторів, повинна відповідати вимогам Правил безпечної експлуатації електроустановок споживачів, затверджених наказом Держнаглядохоронпраці від 09.01.98 N 4, зареєстрованих у Міністерстві юстиції України 10.02.98 за N 93/2533 (далі - НПАОП 40.1-1.21-98);
- Будівлі та приміщення, де розміщені робочі місця операторів, повинні відповідати вимогам нормативно-технічної та експлуатаційної документації виробника ЕОМ з ВДТ і ПП, чинних санітарних норм, санітарних норм і правил, вказаних у ДСанПіН 3.3.2-007-98, та цих Правил;
- Робочі місця заборонено облаштовувати у підвальних приміщеннях або цокольних приміщеннях будинків;

- Площу та об'єм для одного робочого місця оператора визначають згідно з вимогами ДСанПіН 3.3.2-007-98. Площа має бути не менше 6,0 кв. м, об'єм - не менше 20,0 куб. м;
- Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), мають бути надійно захищені діелектричними щитками або сітками з метою недопущення потрапляння працівника під напругу[7].

Також до виробничих приміщень пред'являються вимоги щодо параметрів мікроклімату та вібрації. Ці вимоги дають змогу знизити вплив негативних факторів на здоров'я працівників

Оптимальні умови мікроклімату встановлюються для постійних робочих місць (таблиця 1)[6].

Температурні показники повітря в робочій зоні не повинні перевищувати нормальних значень оптимальної температури для даної категорії робіт, зазначених у таблиці 1, по висоті і по горизонталі, а також протягом робочої зміни [6].

Показники температури повітря в робочій зоні по висоті та по горизонталі, а також протягом робочої зміни не повинні виходити за межі нормованих величин оптимальної температури для даної категорії робіт, наведених в таблиці. 1 [6].

При виконанні робіт операторського типу, пов'язаних з нервово-емоційним напруженням в кабінетах, пультах і постах керування технологічними процесами, в залах обчислювальної техніки та інших приміщеннях повинні дотримуватися оптимальні умови мікроклімату (температура повітря 22-24 град.С, відносна вологість 60-40%, швидкість руху повітря не більш 0,1 м/сек.) [6].

Таблиця 1 - Норми мікроклімату для приміщень [6]

Пора року	Категорія робіт	Температура повітря, °С, не більше	Відносна вологість повітря, %	Швидкість руху, м/сек.
Холодна	Легка - 1а	22-24	40-60	0,1
	Легка – 1б	21-23	40-60	0,1
	Середньої важкості 2а	19-21	40-60	0,2
	Середньої важкості 2б	17-19	40-60	0,2
	Важка – 3	16-18	40-60	0,3
Тепла	Легка - 1а	23-25	40-60	0,1
	Середньої важкості 2а	21-23	40-60	0,3
	Середньої важкості 2б	20-22	40-60	0,3
	Легка - 1б	22-24	40-60	0,2
	Важка - 3	16-18	40-60	0,4

Допустимі величини мікрокліматичних умов встановлюються у випадках, коли на робочих місцях не можна забезпечити оптимальні величини мікроклімату за технологічними вимогами виробництва, технічною недосяжністю та економічно обгрунтованою недоцільністю [6].

Санітарні норми поширюються на шум. Високий рівень шуму і вібрації робочої зони може бути спричинений роботою таких елементів комп'ютерів, як жорсткий диск, вентилятори блоку живлення, охолодження мікропроцесора, швидкісні CD-ROM (DVD-ROM), механічні сканери, пересувні механічні частини принтера.

Гранично дозволений рівень шуму закріплений у Державних санітарних нормах допустимих рівнів шуму в приміщеннях житлових та громадських будинків і на території житлової забудови (наказ від 22.02.2019 № 463).[5].

Параметри постійного шуму на робочих місцях, що нормуються, є рівнями звукових тисків у октавних смугах з середньо геометричними частотами 31,5; 63; 125; 500; 1000; 2000; 4000; 8000 Гц в децибелах. За часовими характеристиками шуми слід поділяти на[5]:

– постійні, рівень шуму яких за повний робочий день при роботі технологічного обладнання змінюється не більш ніж на 5 дБА при вимірюваннях на часовій характеристиці "повільно" шумоміра по шкалі "А";

– непостійні, рівень шуму яких за повний робочий день при роботі технологічного обладнання змінюється більш ніж на 5 дБА при вимірюваннях за часовою характеристикою "повільно" шумоміра по шкалі "А".

Параметрами непостійного шуму (що коливається в часі та переривається) на робочих місцях, які нормуються, є інтегральний рівень - еквівалентний (по енергії) та максимальний рівень шуму у дБА[5].

Високий рівень шуму і вібрації робочої зони може спровокувати швидку стомленість працюючого, погіршення слуху, нервові розлади. Для зниження рівня шуму в приміщенні необхідно встановити шумопоглиначі.

Висновок до розділу

При написанні кваліфікаційної роботи розглядалися питання охорони праці, такі як аналіз шкідливих та небезпечних виробництв, фактори у виробничому приміщенні, карта умов праці (обґрунтування вибору нормованих значень шкідливих та небезпечних факторів виробництва, оцінка факторів виробництва та трудових процесів, гігієнічна оцінка умов праці, оцінка технічний та організаційний рівень, атестація робочого місця.

Умови праці встановлені стандартом вимог до захисних заходів від ураження електричним струмом, а також виконання цих заходів та їх складових частин в електроустановках будівель і споруд.

Санітарні норми виробничого шуму, мікроклімату та приміщення встановлені стандартом які посилаються на нормативні акти та нормативні документи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Про мобільну залежність. URL: <http://supportme.org.ua/teenage/phone-addiction> (дата звернення: 09.06.2022).
2. Законодавство України з охорони праці. URL: <https://pedcollege.kiev.ua/2009-11-13-19-03-22/563-2011-11-15-13-52-32.html>. (дата звернення: 09.06.2022).
3. . ДСТУ 2293:2014 Охорона праці. Терміни та визначення основних понять. URL: http://web.kpi.kharkov.ua/safetyofliving/wp-content/uploads/sites/171/2017/10/dstu_2293_2014.pdf (дата звернення: 09.06.2022).
4. ДСТУ Б В.2.5-82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. URL: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=65395 (дата звернення: 09.06.2022).
5. Санітарні норми виробничого шуму. URL: <https://zakon.rada.gov.ua/rada/show/va037282-99#Text> (дата звернення: 09.06.2022).
6. Санітарні норми мікроклімату виробничих приміщень. URL: <https://zakon.rada.gov.ua/rada/show/va042282-99#Text> (дата звернення: 09.06.2022).
7. Вимоги до виробничих приміщень. URL: https://ips.ligazon.net/document/view/re17588?an=73&ed=2010_03_26 (дата звернення: 09.06.2022).

ЗВІТ

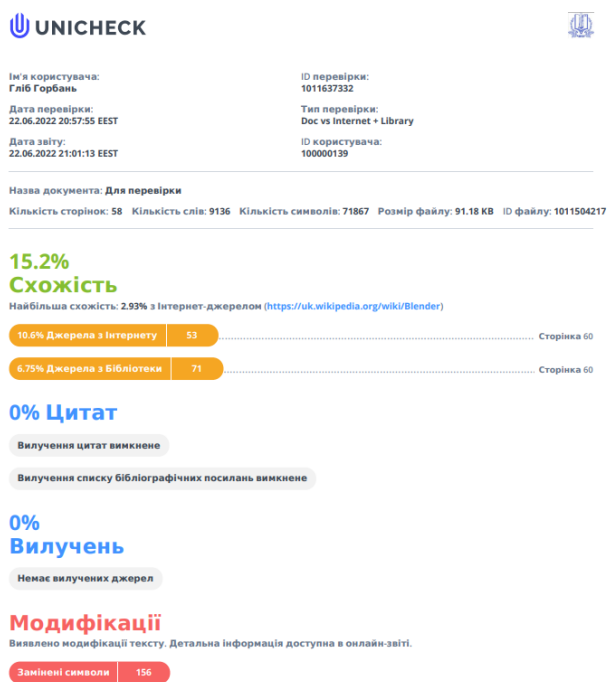
про перевірку на унікальність кваліфікаційної роботи бакалавра на тему:
«Мобільна гра-клікер на базі Unity»

студента спеціальності 121 «Інженерія програмного забезпечення», 408 групи

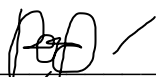
Парахненко Ростислав Андрійович
прізвище, ім'я, по батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck

Результат перевірки тексту кваліфікаційної роботи бакалавра: схожість складає 15.2%.



Студент:


підпис

Р. А. Парахненко
ініціали, прізвище

Керівник:

канд. техн. наук, доцент


підпис ініціали, прізвище

Дата: «__» _____ 2022 р.

ВІДГУК
на кваліфікаційну роботу
студента групи 408 ЧНУ імені Петра Могили
Парахненко Ростислава Андрійовича
«Мобільна гра-клікер на базі Unity»

Кваліфікаційна робота бакалавра здобувачки Парахненко Ростислава Андрійовича присвячена розробки мобільної гри-клікер на базі Unity. У рамках роботи проведено аналіз сучасного стану у ігрових застосунках, здійснено порівняння існуючих застосунків, описано архітектуру розробленого застосунку та оформлено специфікацію вимог.

Тема роботи гарно опрацьована, підібрані вдалі приклади аналогів, результат кваліфікаційної роботи повністю задовольняє поставленим завданням. Здобувачкою самостійно підібрана література та довідникові матеріали, що були уміло та професійно опрацьовані. Під час виконання кваліфікаційної роботи Парахненко Р. А. проявив самостійність, гнучкість мислення та швидке вирішення проблем у критичних ситуаціях, самостійну ініціативу і активність у створенні нових функцій застосунку та роботі з проблемами, що виникали у процесі роботи.

Робота виконана якісно та на високому професійному рівні, а здобувач Парахненко Ростислав Андрійович під час виконання роботи проявив здібності до практичного застосування набутих знань та навичок при реалізації поставлених завдань.

Кваліфікаційна робота «Мобільна гра-клікер на базі Unity» може бути допущена до захисту, а її автор Парахненко Ростислав Андрійович заслуговує присвоєння їй освітньої кваліфікації «Бакалавр з інженерії програмного забезпечення» за спеціальністю 121 «Інженерія програмного забезпечення» галузі знань 12 «Інформаційні технології».

Керівник кваліфікаційної роботи:

канд. техн. наук, доцент кафедри ІПЗ

_____ Г. В. Горбань