

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Ігровий додаток "War of Ages: Battle for Castle"»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології
проектування»

Виконавець роботи: студент групи ІТ.м – 91 Кузьменко Владислав Віталійович

**Кваліфікаційна робота магістра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2020 р.

Науковий керівник

(підпис)

к.т.н., доц., Федотова Н. А.
(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.
(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2020

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ
Зав. секцією ІТП

_____ В. В. Шендрик
«___» _____ 2020 р.

ЗАВДАННЯ на кваліфікаційну роботу магістра студентіві

Кузьменко Владислав Віталійович

1 Тема роботи Ігровий додаток "War of Ages: Battle for Castle"

затверджена наказом по університету від «26» листопада 2020 р. №1824-III

2 Термін здачі студентом закінченого проекту «07» грудня 2020 р.

3 Вхідні дані до проекту

Актуальність індустрії комп'ютерних ігор, моделі існуючих ігрових додатків.

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

Вступ, аналіз предметної області ігрових додатків, моделювання процесу створення ігрового додатку, моделювання варіантів використання ігрового додатку, розробка ігрових правил, ігрової механіки та ігрової концепції, розробка ігрових об'єктів війська, загонів та гравців, розробка програмної реалізації ігрової механіки згідно описаних правил, висновки, список використаної літератури

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Зображення логотипів додатків, що є аналогами «War of Ages: Battle for Castle» та їх ігрового процесу, зображення військових загонів з ігрового додатку «Sid Meier's Civilization V», зображення текстур ігрового додатку «Minecraft», зображення контекстної діаграми А-0 та її декомпозицій, діаграма варіантів використання ігрового додатку, зображення об'єктної моделі ігрового додатку «War of Ages: Battle for Castle у вигляді UML-діаграми, зображення сцен ігрового додатку «War of Ages: Battle for Castle».

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)Завдання прийняв до виконання _____
(підпис)**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Етап ініціювання	01.06.20-22.06.20	
2.	Етап планування	23.06.20-30.06.20	
3.	Розробка інтелектуальної складової ігрового додатка	01.07.20-31.07.20	
4.	Розробка моделей ігрового додатка	01.08.20-13.08.20	
5.	Комп'ютерне моделювання та програмна реалізація ігрового додатка	14.08.20-13.11.20	
6.	Проведення тестування готового продукту	14.11.20-21.11.20	
7.	Аналіз результатів	22.11.20-29.11.20	
8.	Оформлення документації	30.11.20-05.12.20	

Магістрант _____
(підпис)Кузьменко В.В.Керівник роботи _____
(підпис)ктн. доц. Федотова Н.А.

РЕФЕРАТ

Тема роботи: «Ігровий додаток "War of Ages: Battle for Castle"».

Пояснювальна записка містить розділи: «Вступ», «Аналіз предметної області ігрових додатків», «Постановка задачі та методи дослідження ігрових додатків», «Проектування ігрового додатку», «Розробка ігрового додатку», «Висновок», «Список літератури», та два додатки. Вона включає 97 сторінок, 11 таблиць, 65 ілюстрацій і 36 літературних та інтернет-джерел.

В розділі «Аналіз предметної області ігрових додатків» був виконаний опис проекту створення ігрових додатків, огляд існуючих аналогів майбутнього ігрового додатку та зроблено висновки щодо проведеної аналітичної роботи.

В розділі «Постановка задачі та методи дослідження ігрових додатків» були визначені мета та задачі проекту, розроблені вимоги, було обрано апаратне та програмне забезпечення і проведене планування робіт щодо проектування та реалізації майбутнього ігрового додатку.

В розділі «Проектування ігрового додатку» були змодельовані такі діаграми як: контекстна діаграма IDEF0 процесу використання ігрового додатка та Use Case діаграма варіантів використання з гравцем у ролі головного актора.

В розділі «Розробка ігрового додатка» описуються створені правила гри, ігрова концепція та необхідні ігрові механіки, описується покроковий процес створення ігрового додатка «War of Ages: Battle for Castle», а саме: розробка файлів локалізації, розробка спрайтів та розробка програмного сегменту додатку.

Результатом роботи над проектом є готовий проект ігрового додатку «War of Ages: Battle for Castle».

Ключові слова: ігровий додаток, геймдев, логіко-стратегічна гра, C++, SFML.

ЗМІСТ

ВСТУП	7
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІГРОВИХ ДОДАТКІВ	9
1.1 Огляд існуючих аналогів ігрового додатку	10
1.2 Висновки до розділу 1	17
2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ ІГРОВИХ ДОДАТКІВ	18
2.1 Мета та задачі дослідження ігрових додатків	18
2.2 Вибір програмного забезпечення для розробки ігрового додатку	19
2.3 Планування робіт з проектування та реалізації ігрового-дodatку	22
3. ПРОЕКТУВАННЯ ІГРОВОГО ДОДАТКУ	23
3.1 Контекстна діаграма нотації А-0 (IDEF0) процесу використання ігрового додатку	23
3.2 Діаграма варіантів використання ігрового додатку	29
4 РОЗРОБКА ІГРОВОГО ДОДАТКА	32
4.1 Етапи розробки ігрового додатку	32
4.2 Розробка графічних файлів	33
4.3 Головні класи об'єктної моделі ігрового додатку	36
4.4 Другорядні класи об'єктної моделі ігрового додатку	39
4.5 Реалізація війська та армій у ігровому додатку	42

4.6 Майбутній розвиток та наступні версії ігрового додатка	48
ВИСНОВОК.....	50
СПИСОК ЛІТЕРАТУРИ.....	52
ДОДАТОК А.....	56
ДОДАТОК Б.	65
ДОДАТОК В.....	96

ВСТУП

Актуальність. У сучасному світі створення ігрових додатків є одним з найбільших сегментів індустрії розваг. Масштаби ігрової індустрії можна зіставити наприклад з кіноіндустрією.

Актуальність дипломної роботи зумовлена бажанням розвивати власний унікальний програмний продукт, що розрахований на багаторічну підтримку та потенцій розвиток кіберспортивного напрямку.

Тема роботи. Ігровий додаток "War of Ages: Battle for Castle".

Мета роботи є розроблення загальної концепції та правил гри стратегічного ігрового додатку «War of Ages: Battle for Castle» і його послідувача програмна реалізація для платформи Windows.

Предметом дослідження проекту ігрового додатку є процес розробки інформаційної складової додатка згідно сучасних тенденцій та популярних напрямів у геймдеві.

Об'єктом дослідження додатку є математична модель процесу ігрового суперництва гравців заради досягнення перемоги та базована на ній реалізація багато користувальницького режиму гри.

Практичним значенням даного магістерського дипломного проекту є дві функції: розважальна та змагальна. Тобто під час гри можна не тільки з задоволенням проводити свій час, а й змагатись з іншими гравцями в умінні логічно та стратегічно мислити. Також додаток здатен навчити швидко приймати рішення та продумувати свою тактику на декілька хвилин вперед.

Задачами проекту є аналіз тенденції на ринку ігрових додатків, визначення оптимальних методи розробки, необхідного програмного забезпечення, реалізація

базису додатку, механіки гри та основного функціонала, реалізація багато користувальницького програмного функціоналу для майбутнього серверного підключення гравців, реалізація засобу збору післяматчевої статистики та стеження за рейтингом гравців.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІГРОВИХ ДОДАТКІВ

За останні чотири роки ріст індустрії розробки розважальних комп'ютерних ігор не те що відстав від росту кіно, а навіть істотно його випереджав. Найякіснішими іграми вважаються продукти ААА-якості. Це неформальний термін, що означає клас високо бюджетних відеоігор. На один фільм-блокбастер у рік приходиться 2-3 ААА-відеоігри [1-4].

Якщо проаналізувати ступінь впливу на користувачів та їх залучення у інтерактивну середу, яку надають ігрові додатки, то цей сегмент вже давно виділяється у порівнянні з іншими видами розваг. Навіть так званий короткометражний 5-D фільм зі своїми окулярами розширеної реальності, рухами крісел та імітацією природи за допомогою кондиціонерів, вентиляторів і водяних кранів не може порівнятись з грою, сюжет та атмосфера якої може не відпускати гравця від себе десятки або навіть сотні годин [4-9].

Деякі відеоігри можуть бути цікавими навіть протягом тисяч годин. Зазвичай, це ігри, у яких відбувається суперництво між гравцями. Наприклад: хто кращий стратег, хто кращий будівник, хто кращий стрілець і так далі. Частіше за все ці ігри представляють собою кіберспортивні дисципліни, чим ще більше заманюють нових гравців. У таких відеоіграх на перемогу впливає лише вміння грати і нічого більше, тому іноді необхідно витратити сотні годин для того, щоб бути конкурентоспроможним у мережевій грі та на різноманітних турнірах [10-11].

Існує велика кількість жанрів ігрових додатків. Серед найпопулярніших можна виділити шутери, тобто стрілялки від першого лиця, стратегії, як у реальному часі, так і покрокові та аркади. Якщо останні є найпопулярнішим жанром на мобільних пристроях та ігрових приставках, то у перші два важко грати не маючи миші та

клавіатури. Кожен жанр відеоігор вимагає від гравця певного занурення та зосередженості. Шутери у більшій мірі вимагають від гравця швидкої реакції, покрокові глобальні стратегії потребують глибини мисленевих процесів, а стратегії у реальному часі комбінують потребу у швидкій реакції та вмінні продумувати свої дії на декілька хвилин вперед.

Одним із найпопулярніших жанром-відкриттям 2018-2019 років є жанр автобатлерів або автошахів. Він представляє собою неглобальну покрокову стратегію зі збором свого невеликого війська, що буде само битися з супротивником без участі гравця. Тобто геймплейно гравець приймає участь лише у вербуванні та позиціонуванні свого загону на невеликому полі битви. Все інше гра прораховує за нього і йому залишається лише спостерігати. Даний ігровий жанр має дуже низький поріг входження адже в ньому не потрібно ані дуже швидко приймати важкі рішення ані тримати у голові безліч чисел. Достатньо перед кожним боєм просто покращувати свою армію і перемога у партії не заставить себе довго чекати.

Саме через фактор популярності та низького входження було вирішено створити власний ігровий додаток у цьому жанрі.

1.1 Огляд існуючих аналогів ігрового додатку

Розробкою відеоігор або оперуванням багато користувальницьких відеоігор займаються, як правило, не самі розробники, а видавництва. При цьому видавці або оператори локалізують ігри, взаємодіють з власниками платформ, проводять маркетингові компанії та забезпечують технічну і інформаційну підтримку додаткам, що випускаються. Але є і компанії, що самі займаються розробкою ігор та самі їх

випускають. Це такі гіганти як Valve [12], Bethesda [13], Blizzard [14] та Firaxis [15]. Кожна гра цих розробників є або AAA-проектом, або навіть видається з позначкою «шедевр» у магазинах відео-ігор.

1.2.1 Відеогра «Hearthstone» від компанії розробника Blizzard Entertainment та видавця Activision.

Hearthstone – колекційна карткова онлайн-гра, що була створена по мотивам багато користувальницької гри World of Warcraft (рис.1.1) [16].



Рисунок 1.1 – Відеогра «Hearthstone»

У грі містяться «паки», так звані набори карток, відкриваючи які користувачі поповнюють свої колекції. Також гравці можуть створювати власні колоди з тих карток, що вже мають та змагатися за допомогою них з іншими гравцями у стандартному режимі гри, що поділяється на «Звичайна гра» та «Гра на рейтинг».

Ще один режим, що представлений у грі це «Поля битв» (рис 1.2). На відміну від інших ігрових режимів в ньому непотрібно збирати набір стандартної колоди, що складається з 30 карток. Натомість гравцям треба обрати одного з героїв на початку гри, створити потужний загін з завербованих у таверні істот та змагатися у битвах доки не залишиться лише один переможець [17].



Рисунок 1.2 – Режим гри «Поля битв»

Саме останній режим гри був одним із натхненників даного магістерського дипломного проекту. Головною проблемою цієї гри є низький вплив уміння грати на перемогу. Більш ніж 50% впливу має звичайна генерація випадкових чисел. Випадкові істоти у таверні, випадкові герої, випадкові сутички між супротивними гравцями та випадкові удари під час битви. Це значно відштовхує, адже навіть новачок, що перший раз зайшов у гру зможе здолати досвідченого, якщо йому просто пощастить.

1.2.2 Серія відеоігор «Sid Meier's Civilization» від компанії розробника Firaxis Games та видавця 2K Games.

Цивілізація – серія комп'ютерних розважальних відеоігор у жанрі покрокової стратегії (рис. 1.3) [18]. Додатково у іграх цієї серії є поєднання особливостей таких жанрів: глобальна стратегія, варгейм (жанр стратегічних ігор, що відображають у собі реальні та вигадані конфлікти [19]) і 4X (жанр стратегічних ігор, у яких гравець відіграє роль повного правителя та займається вивченням, розширенням, експлуатацією та винищенням [20]).

Sid Meier's CIVILIZATION™

Рисунок 1.3 – Серія ігор «Civilization»

На даний момент у цій серії 7 повноцінних ігор, з яких шість ігор про розвиток людства від печерних людей до робототехніки та міжгалактичних подорожей і одна про розвиток міжконтинентальної імперії (рис 1.4). Остання гра цієї серії – Sid Meier's Civilization VI, що увібрала в себе всі найкращі риси попередніх ігор.



Рисунок 1.4 – Відеогра «Civilization VI»

Ця серія ігор не єдина у світі, що відображає розвиток з давніх віків і до майбутнього, але конкретно в ній це представлено таким різноманітним, цікавим та дуже детальним способом. Тобто гравець не просто розвиває своє військо від дикунів до космічних кораблів, весь світ, природа та інші гравці роблять це разом з ним.

1.2.3 Відеогра «Dota UNDERLORDS» від компанії розробника та видавця Valve.

Dota Underlords – стратегічна гра-автобатлер у всесвіті Dota 2 (рис. 1.5), з участю тих самих героїв, що і у оригінальній грі.



Рисунок 1.5 – Відеогра «Dota UNDERLORDS»

У матчі одночасно змагаються 8 гравців. Так само як і у полях битв гри Hearthstone перемогу отримує лише один гравець, але головна ідея гри значно відрізняється. Замість того, щоб збирати карти істот, у яких зазвичай є лише показники здоров'я та шкоди у цій грі необхідно збирати героїв, що мають більше ніж 30 унікальних показників (рис. 1.6).



Рисунок 1.6 – Ігровий процес у «Dota UNDERLORDS»

Головний мінус гри – це кількість часу, що витрачається на один матч. Зазвичай це 30 хвилин, але у також присутній турбо-режим, коли партія триває максимум 10 хвилин [21].

Проаналізувавши описані вище аналоги було складено порівнювальну таблицю. Так як кожна з описаних вище ігор – це багатомільйонний AAA-проект, то і критерії були обрані більш загальні. Порівняння було проведено за такими критеріями:

1. Наявність російської, англійської та української мов,
2. Можливість адекватної роботи на бюджетних варіантах комп'ютерної конфігурації,
3. Можливість грати у ту саму версію гри зі свого смартфона,
4. Можливість самостійної зміни системних файлів для створення модифікацій гри,
5. Open Source.

Таблиця 1.1 – Порівняння аналогів

Критерій	Hearthstone Battlegrounds	Sid Meier's Civilization VI	Dota UNDERLORDS	War of Ages: Battle for Castle (мій додаток)
Наявність російської, англійської та української мов	-	-	+	+
Можливість адекватної роботи на бюджетних варіантах комп'ютерної конфігурації	-	-	+	+
Можливість грати у ту саму версію гри зі свого смартфона	+	-	+	-
Можливість самостійної зміни системних файлів для створення модифікацій гри	-	+	-	+
Open Source	-	-	-	+

1.2 Висновки до розділу 1

Ігрові додатки вже давно є частиною нашого повсякденного життя. Хтось воліє грати на приставках до телевізора, хтось любить ноутбуки, а хтось не може грати ні на чому крім свого класного персонального комп'ютеру.

Бюджет сучасних ігрових додатків давно переступив навіть бюджет фільмових блокбастерів. Сюжет фільму зазвичай пишуть дві-три людини, а над сюжетом гри зазвичай працюють цілі команди з двадцятьох людей. І це не дивно, так як новітні технології здатні навіть із фільму зробити інтерактивну гру. Не відстає також і кіберспорт. Щомісячні турніри з величезними грошовими винагородами та сотні мільйонів фанатів по всьому світу. І як після такої популярності у людей не бажати розвивати власний продукт.

2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ ІГРОВИХ ДОДАТКІВ

2.1 Мета та задачі дослідження ігрових додатків

Виконавши аналіз зазначеної предметної області, проаналізувавши аналоги, сформувавши мету, предмет та об'єкт дослідження ми можемо сформулювати задачі даного проекту:

- проаналізувати сучасні тенденції на ринку розробки ігрових додатків і сформулювати концепцію та правила гри;
- визначити оптимальні методи розробки, доцільне програмне забезпечення, та на їх основі реалізувати базис додатку, механіку гри та основний функціонал;
- реалізувати багато користувальницький програмний функціонал для майбутнього серверного підключення гравців до ігрового лобі;
- реалізувати засіб для збору післяматчевої статистики для налагодження ігрового балансу;
- реалізувати засіб для стеження за внутрішнім розвитком гравців, а саме автоматично призначати їм відповідний ELO-рейтинг для більш збалансованого підбору багато користувальницьких партій.

Для вдалої та коректної реалізації додатку необхідно визначити перелік необхідних вимог. Основними вимогами до розробки додатку є:

- реалізація функціоналу додатку згідно встановленим правилам;
- персональні налаштування користувача;
- можливість пройти навчання з роз'ясненням основ гри;

- проведення всіх розрахунків кожної партії на сервері для запобігання ігрового шахрайства;
- наявність статистики кожного гравця;
- наявність рейтингу, що відповідає міжнародній системі рейтингового оцінювання Elo у кожного гравця, який буде змінюватись після зіграної багато користувальницької партії.

2.2 Вибір програмного забезпечення для розробки ігрового додатку

Для програмної реалізації ігрового додатка існують декілька спеціалізованих ігрових рушіїв, наприклад Unity3D та Unreal Engine 4, але в цьому проекті я буду використовувати власний. Моя відеогра буде написана на чистій C++ з використанням бібліотеки SFML так як не існує жодних інших альтернатив цій швидкій та продуктивній мові програмування [22]. Для створення двовимірних графічних об'єктів зазвичай використовують засоби роботи з растровою графікою. Найпопулярніший такий засіб – це Adobe Photoshop.

SFML – бібліотека C++ для простої та швидкої роботи з графікою та мультимедіа.

Unity3D – це крос-платформовий засіб, який використовують для створення ігрових додатків. Для написання коду на цьому рушії використовуються компілятори таких мов програмування як C# та JavaScript. Також присутня можливість одночасного перегляду, відладки та доопрацювання теперішнього проекту [23].

C# – крос-платформна мова програмування. У рушії Unity3D унаслідкується від «Mono Behavior».

Adobe Photoshop – найпопулярніший редактор растрової комп'ютерної графіки від компанії розробника програмного забезпечення Adobe [24].

Unreal Engine 4 – ігровий рушій для створення ігрових додатків. За основу рушія використовується компілятор мови програмування C++ [25].

Для оцінки кожного з варіантів ігрових рушіїв доцільно буде скласти порівняльну таблицю їх можливостей. Порівняння відбулося за такими критеріями:

1. Компілятор мови програмування,
2. Швидкість роботи,
3. Необхідність встановлення додаткових засобів,
4. Перевантаженість зайвими засобами, які лише заважають при розробці,
5. Продуктивність на ПК зі слабкою конфігурацією.

Таблиця 2.1 – Порівняння рушіїв гри

Критерій	Unity3D	Unreal Engine 4	Власний рушій
Компілятор мови програмування	C# / JavaScript	C++ / JavaScript	C++ SFML
Швидкість роботи	Середня	Нижче ніж у Unity3D	Найшвидша
Необхідність встановлення додаткових засобів	Unity3D як зайвий додатковий засіб та його модулі	Unreal Engine 4 як зайвий додатковий засіб та його модулі	Лише засоби для роботи з C++
Перевантаженість зайвими засобами, які лише заважають при розробці	Оціночні 10-20 ГБ та 200-300 МБ кожен проект	Оціночні 40-50 ГБ та 2-3 ГБ кожен наступний проект	14 МБ встановлення C++ необхідної версії, 24 МБ розмір бібліотеки та до 100 МБ кожен проект
Продуктивність на ПК зі слабкою конфігурацією	Невеликі зависання при тестуванні продукту, але в цілому працювати можливо	Дуже важко працювати навіть на ПК з середньою конфігурацією	Можна працювати навіть на найслабших ПК

2.3 Планування робіт з проектування та реалізації ігрового- додатку

Після формування мети та визначеного переліку задач, що треба виконати необхідно спланувати список робіт задля виконання дипломного проекту. Для цього необхідно створити:

- 1) структурну декомпозицію робіт для організації командної роботи, тобто WBS;
- 2) організаційну структуру проекту для відображення учасників та їх безпосередніх керівників, тобто OBS;
- 3) діаграму Ганта для розподілу людських ресурсів та часу згідно запланованим роботам;
- 4) матрицю відповідальності для дотримання структури виконання поставлених робіт.

Також необхідно прорахувати можливі ризики, а саме вірогідності і причини виникнення та визначити методи для їх усунення.

Усі вони наведені у Додатку А.

3. ПРОЕКТУВАННЯ ІГРОВОГО ДОДАТКУ

Після завершення аналізу предметної області даного магістерського проекту, визначення його мети та задач, сформованих функціональних вимог та обраного програмного забезпечення доцільно перейти до проектування ігрового додатку.

Для створення майбутньої оптимальної програмної реалізації слід зайнятися проектуванням, тобто процесом вирішення задач та їх раціонального планування.

3.1 Контекстна діаграма нотації A-0 (IDEF0) процесу використання ігрового додатку

Побудова контекстної діаграми A-0 – перший крок до успішного проектування ігрового додатку. Така діаграма складається з чотирьох головних елементів, котрі відповідають за свою частину впливу на виконання робіт: вхідні дані, управління, механізми та вихідні дані [32-33].

Після проведеного аналізу стосовно головних елементів даної контекстної діаграми «Процес ігрової партії у додатку "War of Ages: Battle for Castle"», був сформований такий перелік даних:

- Вхідні дані: модель ігрового поля, модель ігрових загонів, модель серверних прорахунків та графічні матеріали.
- Вихідні дані: результат партії.
- Управління: методологія створення додатку з бібліотекою SFML, правила гри та вимоги до гри.

- Механізми: ігровий додаток, апаратне забезпечення та програмне забезпечення.

На рис.3.1 представлена контекстна діаграма процесу використання А-0, що була змодельована за допомогою програми Erwin Process Modeler на основі цих даних.



Рисунок 3.1 – Контекстна діаграма А-0

Так як дана діаграма містить у собі тільки загальну інформацію про систему у додатку, то необхідно виконати декомпозицію.

Декомпозиція – це метод розподілу складної системи на її більш прості складові елементи. Вона дозволяє більш детально проаналізувати логічні зв'язки у послідовності виконання робіт задля подальшої програмної реалізації майбутнього результату проекту.

Діаграма А-0 була декомпозована на три підрівні, а саме: відображення меню, процедура ігрової партії та підведення підсумків партії. Пов'язано з тривалістю процесу ігрової партії, що умовно поділяється на три етапи:

- першим етапом є відображення меню, тобто місце, де гравці зможуть виконати необхідні їм налаштування та приєднатися до партії;
- другим етапом є процедура ігрової партії, тобто обслуговування партії від початку та до її кінця з підтримкою усіх ігрових механік та багато користувальницької взаємодії;
- третім етапом є підведення підсумків партії, тобто запис та аналіз вихідних результатів партії.

Під час опису першого етапу ігрової партії були сформовані такі показники:

- вхідні дані: графічні матеріали;
- вихідні дані: створення ігрового лобі;
- управління: методологія створення додатку з бібліотекою SFML та вимоги до гри;
- механізми: ігровий додаток, апаратне забезпечення та програмне забезпечення.

Під час опису другого етапу ігрової партії були сформовані такі показники:

- вхідні дані: створення ігрового лобі, графічні матеріали, модель серверних прорахунків, модель ігрового поля та модель ігрових загонів;
- вихідні дані: процедура завершення партії;
- управління: методологія створення додатку з бібліотекою SFML, правила гри та вимоги до гри;
- механізми: ігровий додаток, апаратне забезпечення та програмне забезпечення.

Під час опису третього етапу ігрової партії були сформовані такі показники:

- вхідні дані: процедура завершення партії, графічні матеріали та модель серверних прорахунків;
- вихідні дані: результат партії;

- управління: методологія створення додатку з бібліотекою SFML та вимоги до гри;
- механізми: ігровий додаток, апаратне забезпечення та програмне забезпечення.

На рис.3.2 наведений вигляд контекстної діаграми після її декомпозиції на три ігрові етапи.

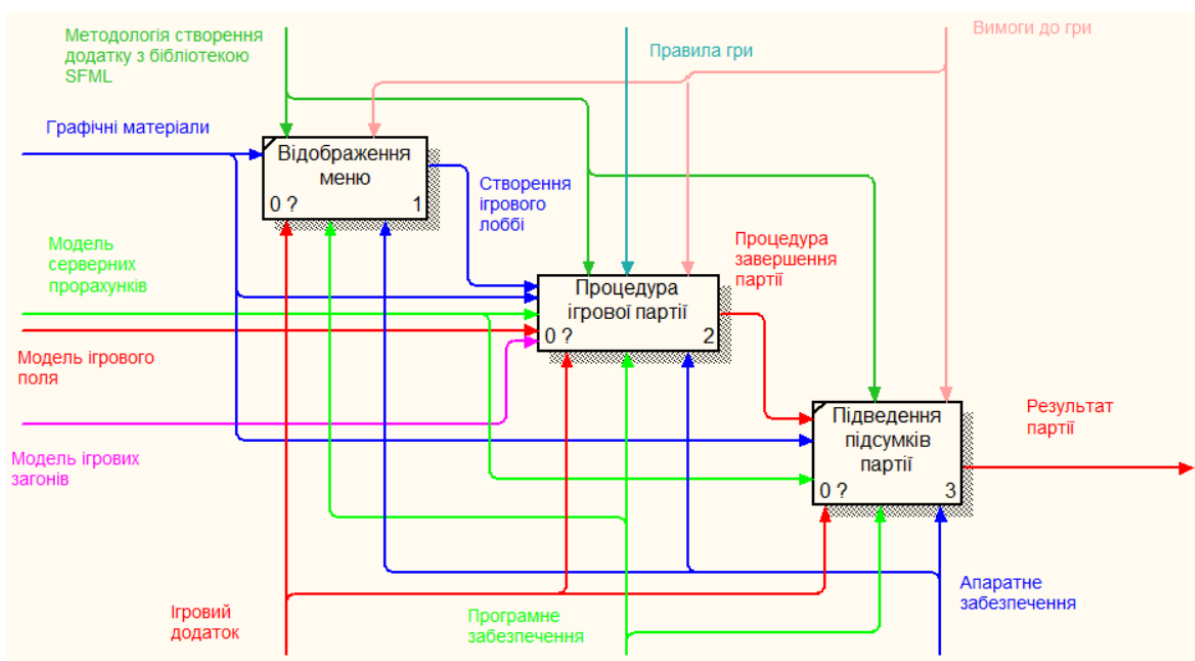


Рисунок 3.2 – Декомпозиція контекстної діаграми А-0

Після розбиття на три ігрові процеси даної діаграми А-0 слід виконати декомпозицію другого процесу за умови, що кожен попередній результат є вхідними даними для наступних.

Процес «Процедура ігрової партії» був декомпований на п'ять релевантних підрівнів, а саме: підключення гравців у лобі, перевірка кількості гравців, вербування загонів військ, процедура бою між супротивними гравцями та вибування гравців, що програли. Це зумовлено процесом відтворення ігрової партії, що умовно поділяється на п'ять функціональних етапів:

- першим етапом є підключення гравців до ігрового лобі;
- другим етапом є перевірка кількості гравців, щоб було не більше восьми

для початку партії та не менше одного для її завершення;

- третім етапом є вербуванням гравцями загонів військ;
- четвертим етапом є бій між кожним гравцем;
- п'ятий етап: гравці, що зазнали фатальної шкоди вибувають з партії.

Під час опису першого ігрового етапу були сформовані такі показники:

- вхідні дані: графічні матеріали, створення ігрового лобі, модель серверних прорахунків;
- вихідні дані: готовність гравців;
- управління: вимоги до гри, методологія створення додатку з бібліотекою SFML;
- механізми: ігровий додаток, апаратне забезпечення та програмне забезпечення.

Під час опису другого ігрового етапу були сформовані такі показники:

- вхідні дані: графічні матеріали, модель серверних прорахунків, готовність гравців та кількість вибулих гравців;
- вихідні дані: перехід у фазу вербування;
- управління: вимоги до гри, методологія створення додатку з бібліотекою SFML;
- механізми: ігровий додаток, апаратне забезпечення та програмне забезпечення.

Під час опису третього ігрового етапу були сформовані такі показники:

- вхідні дані: графічні матеріали, модель серверних прорахунків, модель ігрового поля та модель ігрових загонів;
- вихідні дані: перехід у фазу бою;

- управління: вимоги до гри, методологія створення додатку з бібліотекою SFML та правила гри;

- механізми: ігровий додаток, апаратне забезпечення та програмне забезпечення.

Під час опису четвертого ігрового етапу були сформовані такі показники:

- вхідні дані: графічні матеріали, модель серверних прорахунків, модель ігрового поля та модель ігрових загонів;

- вихідні дані: повідомлення про кількість шкоди гравцям;

- управління: вимоги до гри, методологія створення додатку з бібліотекою SFML та правила гри;

- механізми: ігровий додаток, апаратне забезпечення та програмне забезпечення.

Під час опису п'ятого ігрового етапу були сформовані такі показники:

- вхідні дані: графічні матеріали, модель серверних прорахунків;

- вихідні дані: кількість вибухливих гравців;

- управління: вимоги до гри, методологія створення додатку з бібліотекою SFML;

- механізми: ігровий додаток, апаратне забезпечення та програмне забезпечення.

На рис.3.3 наведено вигляд процесу «Процедура ігрової партії» після розбиття на п'ять етапів.

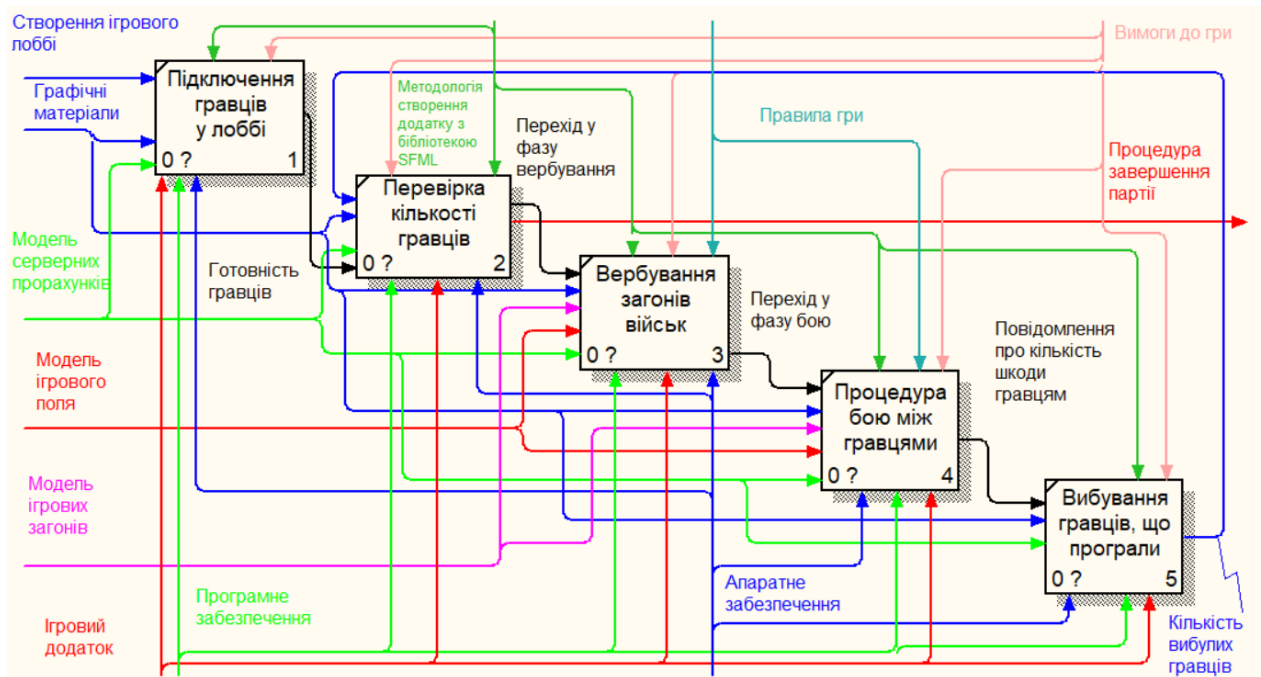


Рисунок 3.3 – Декомпозиція процесу «Процедура ігрової партії»

3.2 Діаграма варіантів використання ігрового додатку

Діаграма варіантів використання є наступним етапом у проектуванні майбутнього продукту. Вона повинна відображати залежність та взаємозв'язки між одним або декілька акторами та різноманітними варіантами як вони зможуть використовувати продукт. За допомогою неї можна спостерігати та відстежувати, які саме дії чи послідовності дій актора мають або призведуть до виконання певних процесів [34-35].

Для реалізації Use Case діаграми був визначений Гравець як єдиний актор, котрий сам буде обирати, що йому необхідно, щоб відбувалося у додатку. Після визначення ключової особи, що буде взаємодіяти з визначеною системою, треба

визначити перелік потенційних варіантів використання. Варіанти використання ігрового додатку:

- Змінити налаштування додатку;
- Отримати довідку згідно гри;
- Завершити роботи додатку;
- Проходження одиночного режиму навчання;
- Підключення до багато користувальницького лобі.

Деякі з варіантів використання мають продовження.

Варіант використання «Змінити налаштування додатку» має два варіанти використання:

- змінити налаштування звуку;
- обрати мову.

Варіанти «Проходження одиночного режиму навчання» та «Підключення до багато користувальницького лобі» мають спільне продовження «Ігровий процес стандартної партії», що у свою чергу перенаправить до процесу «Показати результат гри».

На рис.3.4 наведена розроблена діаграма варіантів використання, що була створена на основі визначених варіантів використання та інформації про актора.

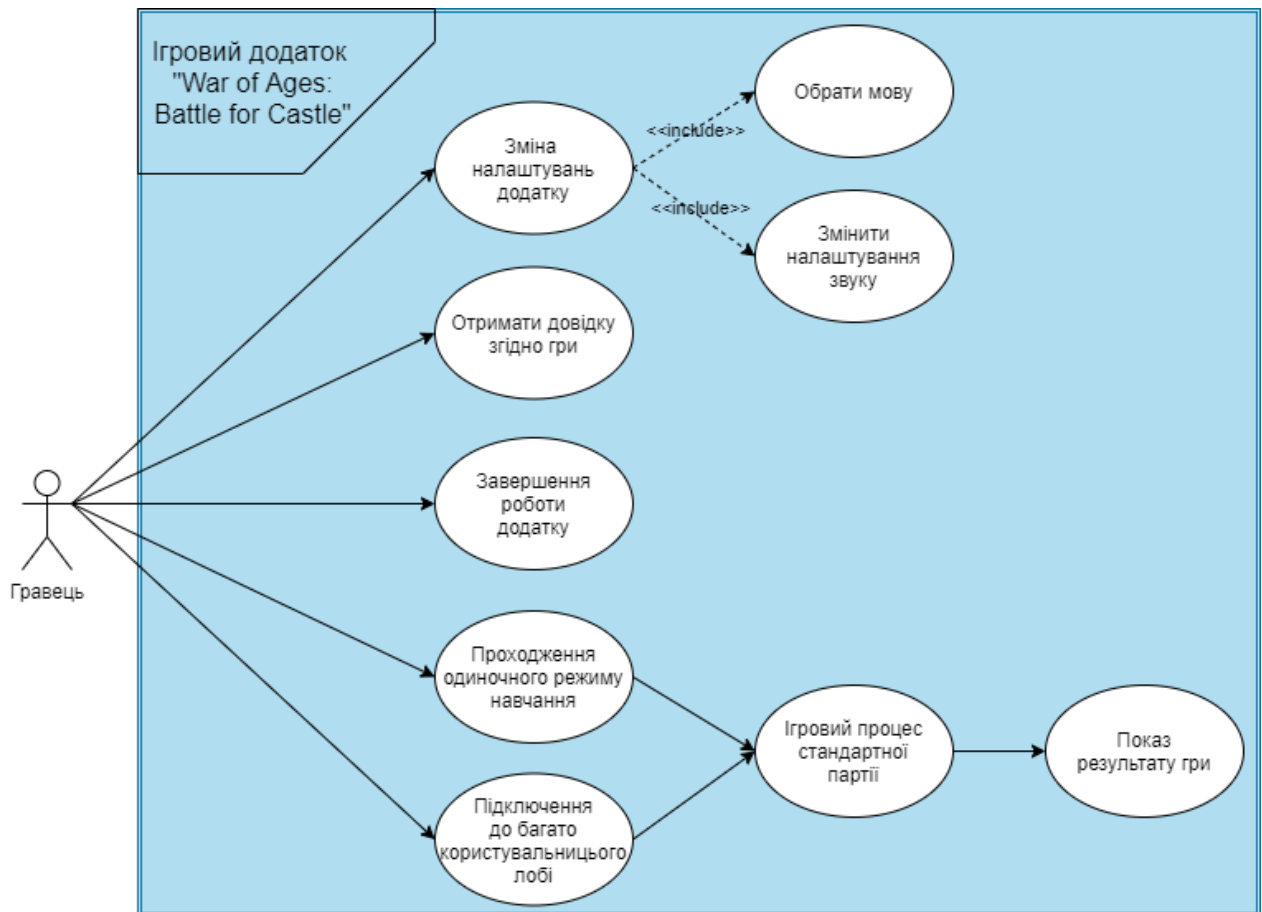


Рисунок 3.4 – Діаграма варіантів використання ігрового додатку

4 РОЗРОБКА ІГРОВОГО ДОДАТКА

4.1 Етапи розробки ігрового додатку

Розробка додатку була поділена на 8 умовних етапів, для кожного з яких був використаний певний засіб реалізації. У таблиці 4.1 зазначені усі етапи з відповідним їм засобом реалізації.

Таблиця 4.1 Етапи розробки та їх засоби реалізації

Номер етапу	Етап розробки ігрового додатку	Засіб реалізації
1	Розробка правил та концепції ігрового додатку	–
2	Розробка текстур та спрайтів.	Adobe Photoshop
3	Розробка ієрархічної об'єктної моделі залежностей для програмної реалізації.	–
4	Розробка класів-перерахувань (enum classes) для збереження та порівняння характеристик об'єктів.	C++
5	Адаптація та обробка розроблених графічних об'єктів у програмному середовищі	Бібліотека SFML
6	Програмна реалізація описаної об'єктної моделі з використанням класів-перерахувань (enum classes).	C++
7	Реалізація основної ігрової механіки.	C++
8	Реалізація збору статистики та стеження за внутрішнім розвитком гравця.	C++

4.2 Розробка графічних файлів

Для ігрового додатку були розроблені такі растрові файли: текстура усіх загонів, текстура усіх епох, растрове зображення папірусу та растрові зображення прапорів для мов локалізації.

4.2.1 Текстура усіх військових загонів

Дана текстура містить у собі всі спрайти доступних у грі військових загонів (рис. 4.1). Кожен її окремий елемент є власністю компанії Firaxis Games та використовується лише для демонстрації та некомерційної діяльності, якою зараз виступає дипломний проект. Дана текстура була скомпільована у одне зображення для програмного виділення та вирізання конкретних частин і подальшого присвоєння до ігрових покажчиків. Розмір текстури 1800 на 800 пікселів.



Рисунок 4.1 – Текстура з усіма доступними військовими загонами

4.2.2 Текстура усіх історичних епох

Дана текстура містить у собі всі спрайти, що відображають кожен епоху з представлених у грі та відповідають найпопулярнішому матеріалу згідно історії (рис. 4.2), а саме: трава для нейтральної території, кругляк для епохи Стародавнього світу, напівокислена мідь для Класичної ери, камінна цегла для епохи Середньовіччя, звичайна цегла для епохи Відродження, вугілля для епохи Промислового ера, залізо для епохи Нового часу, уран та свинець для Епохи Атому, золото для Інформаційної ери та діаманти для епохи Майбутнє. Кожен її окремий елемент є власністю компанії Mojang та використовується лише для демонстрації та некомерційної діяльності, якою зараз виступає дипломний проект. Кожен елемент є тридцятишестикратно помноженим результатом стартового зображення. Дана текстура була скомпільована у одне зображення для програмного виділення та вирізання конкретних частин і подальшого присвоєння до зображення заднього фону. Розмір текстури 1200 на 120 пікселів.

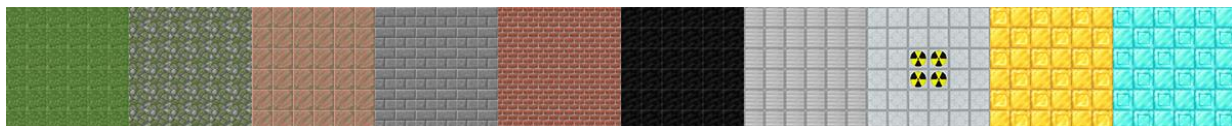


Рисунок 4.2 – Текстура з усіх історичних епох

4.2.3 Растрове зображення папірусу

Це зображення представляє собою папірус (рис. 4.3). У проекті воно використовується у якості заднього фону для пунктів меню, а також для заднього фону при відображенні текстової інформації щодо характеристик війська. Початковий розмір зображення 340 на 510 пікселів. Зазвичай деформується за допомогою Scale.

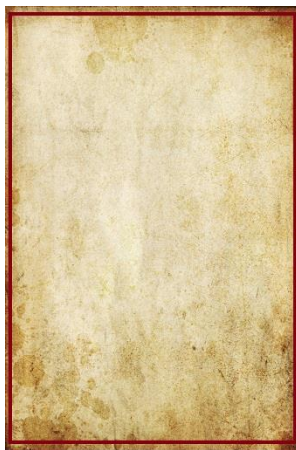


Рисунок 4.3 – Растрове зображення папірусу

4.2.4 Растрові зображення національних прапорів

Ці зображення представляють собою національні прапори України (рис. 4.4), Сполучених Штатів Америки (рис. 4.5) та Росії (рис. 4.6). У проекті вони використовуються як тривіальні розпізнавальні об'єкти для визначення мови додатку у налаштуваннях. Розмір кожного зображення 256 на 256 пікселів.



Рисунок 4.4 – Растрове зображення прапору України



Рисунок 4.5 – Растрове зображення прапора Сполучених Штатів Америки



Рисунок 4.6 – Растрове зображення прапора Росії

4.3 Головні класи об'єктної моделі ігрового додатку

Для ефективної оптимальної реалізації додатку з мінімумом послідуочого рефакторингу необхідно спочатку спроектувати його. Тобто виділити всі необхідні класи, що будуть приймати участь у ігровому процесі та працювати саме з ними.

Перший такий клас – це клас Army, що є батьківським класом для всіх можливих пасивних учасників ігрового процесу, тобто – ігрових об'єктів. Нащадками цього класу будуть клас військових загонів Squad та потенційний майбутній клас укріплень Walls. На рисунку 4.7 зображені взаємозв'язки між цими класами.

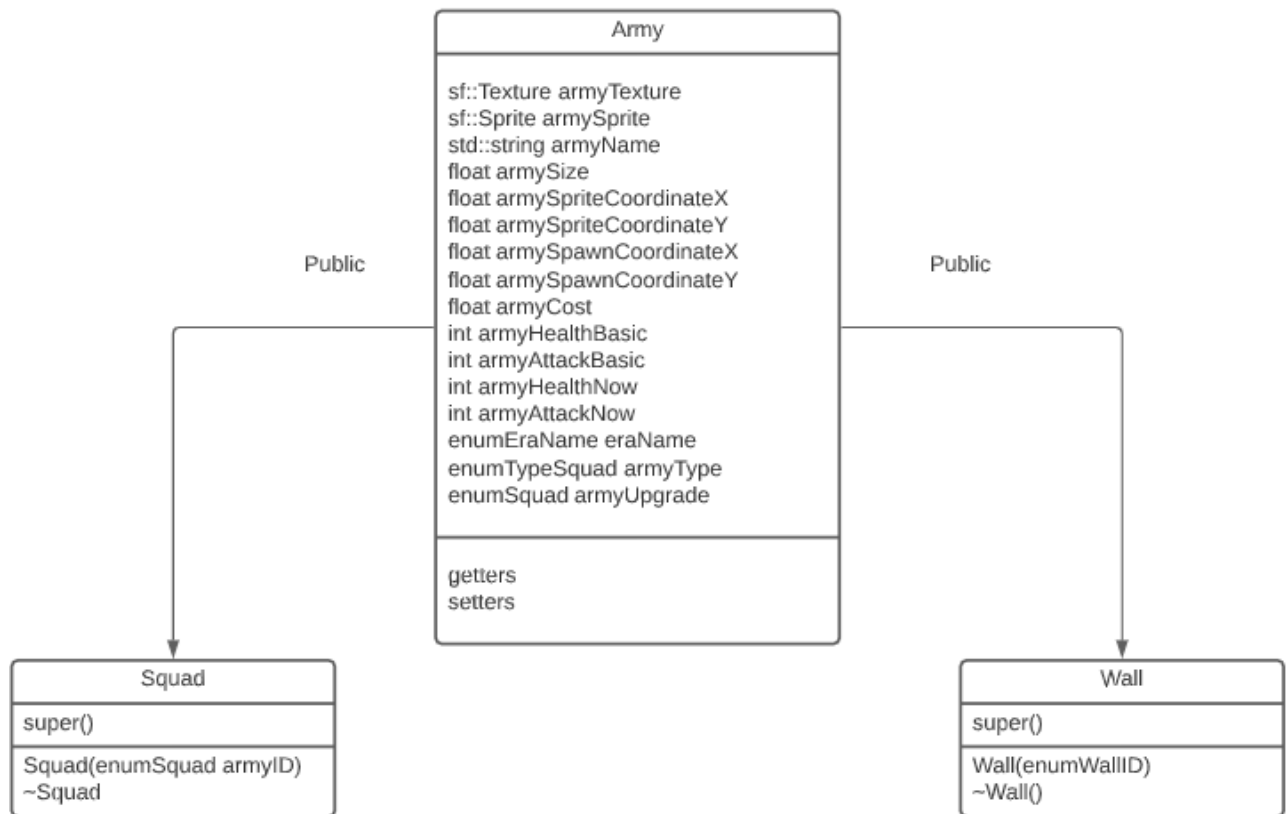


Рисунок 4.7 – Взаємозв’язки між батьківським класом Army та його нащадками

Дані класи мають такі атрибути: `armyTexture`, що відповідає за загальну текстуру війська, `armySprite`, що відповідає за спрайт загону, `armyName`, що відповідає за назву загону, `armySize = 200`, що відповідає за квадратний розмір спрайту на загальній текстурі загону, `armySpriteCoordinateX`, `armySpriteCoordinateY`, `armySpawnCoordinateX`, `armySpawnCoordinateY`, що відповідають за розміщення спрайту на текстурі та вікні, `armyCost`, що відповідає за ціну загону у балах дій, `armyHealthBasic`, `armyAttackBasic`, `armyHealthNow`, `armyAttackNow`, що відповідають за базові та поточні характеристики загонів, `eraName`, `armyType`, що відповідають за приналежність загону то епохи та типу та `armyUpgrade`, що відповідає за загін, у який буде покращено військо.

Другий такий клас – це клас `Player`, що визначає усіх активних учасників ігрового процесу, тобто – живих гравців.

У кожного гравця буде свій замок з 50 балами життя, тому треба буде створити 8 окремих об'єктів цього класу та призначити кожному гравцю по одному. Кількість балів дій та кількість комірок на лаві запасних будуть визначатися в залежності від рівня епохи. Також будуть необхідні метод для нанесення шкоди, тобто зменшення балів життя замку, метод для зміни ери та метод отримання епохи замку. На рисунку 4.8 зображена схематична діаграма цього класу.

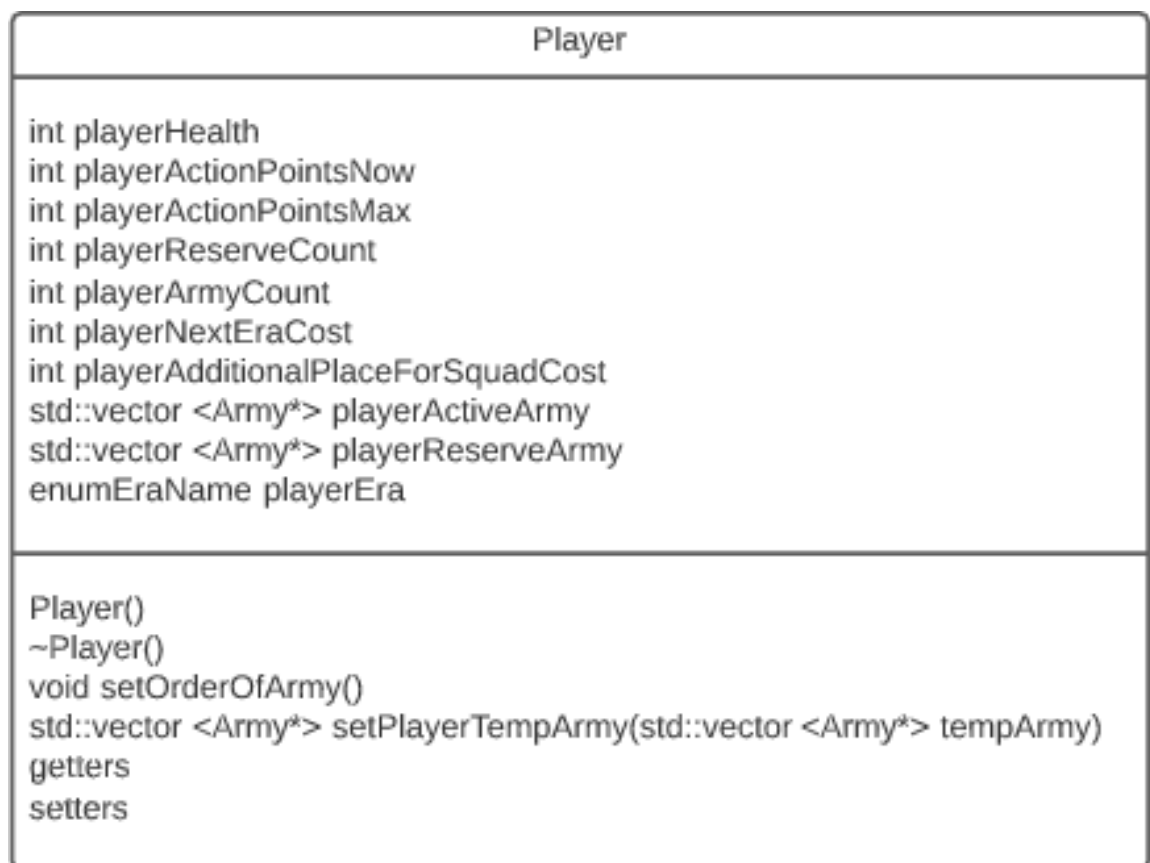


Рисунок 4.8 – Клас `Player` з його атрибутами та методами

Даний клас має такі атрибути: `playerHealth`, що відповідає за кількість балів життя у замка гравця, `playerActionPointsNow`, `playerActionPointsMax`, що відповідають за поточну та максимальну кількість балів дій у гравця,

`playerReserveCount`, `playerArmyCount`, що відповідають за максимальну кількість загонів у армії та резерві гравця, `playerNextEraCost`, `playerAdditionalPlaceForSquadCost`, що відповідають за ціну у балах дій для переходу гравця у наступну епоху та розширення кількості доступних комірок, `playerActiveArmy`, `playerReserveArmy`, що відповідають за поточну основу та резервну армії гравця, `playerEra`, що відповідає за епоху, у якій зараз знаходиться гравець.

4.4 Другорядні класи об'єктної моделі ігрового додатку

Усі інші класи не є основними для роботи додатку та їх екземпляри не будуть створюватись у `main`-функції ігрового додатку, але вони зазвичай є елементами головних класів.

4.4.1 Класи-перерахування

У додатку присутні три класи-перерахування, що були створені згідно ігрових правил та концепції ігрового додатка. Перший такий клас `enumSquad` (рис. 4.9) відповідає за всі доступні військові загони у додатку. Другий `enumEraName` (рис. 4.10) відповідає за список всіх доступних ігрових епох, а третій `enumTypeSquad` (рис. 4.11) за список усіх типів військ згідно їх особливостей.

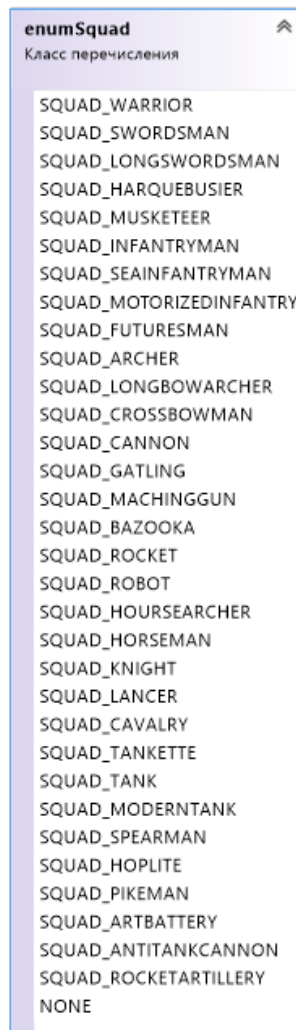


Рисунок 4.9 – Клас Player з його атрибутами та методами

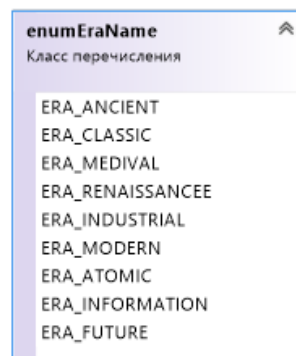


Рисунок 4.10 – Клас Player з його атрибутами та методами

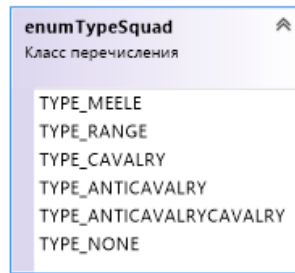


Рисунок 4.11 – Клас Player з його атрибутами та методами

4.4.2 Файли локалізації

У додатку присутні три файли, що відповідають за його локалізацію: StringsInfo.h, що містить у собі переведені на три мови усі рядки, що відносяться до інформації про військові загони, StringsMenu.h, що містить у собі переведені на три мови усі рядки, що відносяться до ігрового меню та LanguageState.h, що містить у собі єдину змінну-стан, яка відповідає за поточну обрану мову у ігровому додатку.

4.4.3 Файл карти

У додатку присутній файл map.h, що відповідає за генерацію візуального заднього фону, який залежить від теперішньої епохи конкретного гравця. Дана карта реалізована за допомогою двомірної квадратної матриці з трьома термами: нейтральна територія (завжди текстура трави), територія першого гравця (завжди відповідає епохи того гравця, під чиїм керуванням зараз знаходиться гра) та територія другого гравця, що відповідає епосі його теперішнього супротивника.

4.4.4 Файл меню

У додатку присутній файл Menu.h, що відповідає за малювання ігрового меню у вікні, а також за його пункти, тобто пункт налаштування та пункт допомоги. Даний файл є стартовим під час старту взаємодії з додатком і саме від дій у ньому залежить подальша робота ігрового додатку.

Розроблена діаграма класів зображена на рисунку 4.12 та передає схематичне відображення включення елементів одних класів до інших за допомогою стрілок.

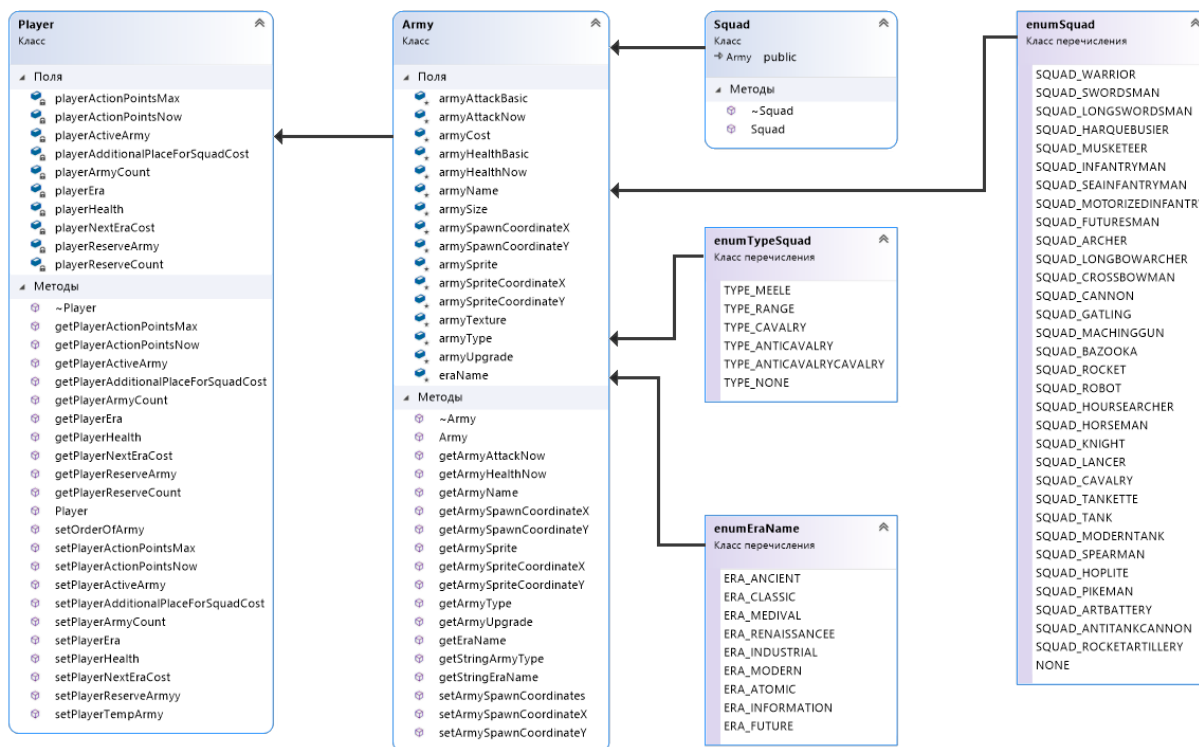


Рисунок 4.12 – Схематична діаграма класів

4.5 Реалізація війська та армій у ігровому додатку

За створення кожного загону відповідав конструктор класу Squad. З вхідних параметрів він має лише порядковий номер (або ID) класу перерахувань військових загонів і вже базуючись на ньому створювався повноцінний військовий загін. У додатку В описаний фрагмент конструктору створення військового загону, що відображає процес створення Воїну з епохи Стародавнього світу. Процес створення

будь якого іншого військового загону відрізняється лише точковим налаштуванням ігрових параметрів.

Для відтворення загального набору (з якого описаним у правилах псевдо випадковим способом будуть пропонуватись війська та укріплення) необхідно спочатку додати всі війська до нього. Кількісно загальний набір відповідно до епох описаний у таблиці 4.2.

Таблиця 4.2 – Кількість військ кожної епохи

Назва епохи	Кількість представників кожного з унікальних типів
Стародавній світ	16
Античність	16
Середньовіччя	16
Відродження	16
Промислова ера	16
Новий час	12
Епоха атому	8
Інформація ера	8
Майбутнє	6

Якщо порахувати, то загальний набір усіх військових загонів складається з 412 одиниць. Тобто для маніпулювання усіма пасивними ігровими об'єктами необхідно створити 412 покажчиків. Хоч вони і були створені від покажчика одного класу – все ж це досить великий об'єм даних, не кажучи вже про засоби їх об'єднання та групування. Всі ці покажчики поділяються на 9 груп, тобто військові заони кожної з

дев'яти епох. Тому було вирішено створити 9 векторів показників, що відповідають кожній епосі та один вектор векторів для зручнішого користування.

Найбільшою проблемою при створенні такої кількості показників є проблема часу та проблема ресурсів. Якщо лінійно почергово створювати кожен показник, то це займає близько однієї хвилини часу. Для уникання почергового виконання програми у програмуванні існують потоки. Це засоби, що дозволяють при можливостях процесору комп'ютера збільшити навантаження на кожне ядро, але при цьому прискорити виконання необхідних задач. Таке програмування називається асинхронним або паралельним коли кожний рядок коду замість того, щоб чекати поки виконається його попередник – виконується одразу одночасно з іншими частинами програми. Але це можливо тоді, коли процесор є багатоядерним інакше жодного приросту у продуктивності не відбудеться. Так як на дворі 2020 рік, то 99% усіх процесорів є багато ядерними і навіть якщо замість необхідних для виконання коду 8 процесорів буде лише два, то навіть приросту продуктивності у 2 рази буде достатньо. Тому було вирішено використовувати потоки. Завдяки ним час скоротився у 5 разів з хвилини при одному потоці до 12 секунд при 9. На рисунку 4.13 зазначена потреба програми у ресурсах при максимальному навантаженні від роботи потоків створення та додавання до векторів показників загонів та їх покрокове зниження до кінця часу використання потоків.

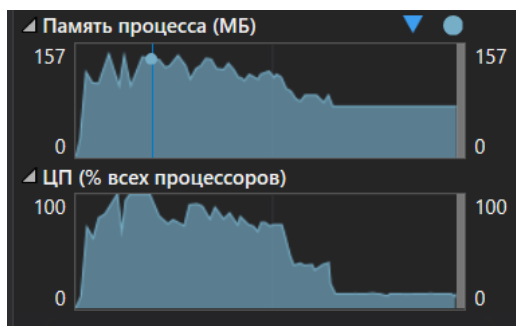


Рисунок 4.13 – Проблематика часу та ресурсів при використанні асинхронного програмування

Для графічного динамічного відображення кількості балів життя та нападу кожного загону використовуються засоби бібліотеки SMFL, а саме по два об'єкта тексту, що приймають значення з класу Squad та два об'єкти векторних кіл, що візуально виділяють текстові значення. Положення кожного з описаних об'єктів залежить від визначених у класі координат спрайту у вікні додатку. Типовий вигляд кожного військового загону з максимальними їх значеннями балів життя та нападу зазначені на рисунку 4.14.



Рисунок 4.14 – Згенеровані загони з фігурними позначеннями балів життя та шкоди

У користувацькому графічному інтерфейсі якщо гравець наводитиме курсор миші на будь який військовий загін (не важливо ворожий чи власний), то йому буде виведена вся базова інформація про нього, а саме: назва загону, його тип, епоха та опис пасивних властивостей. На рисунках 4.15, 4.16 та 4.17 зображені такі згенеровані приклади.

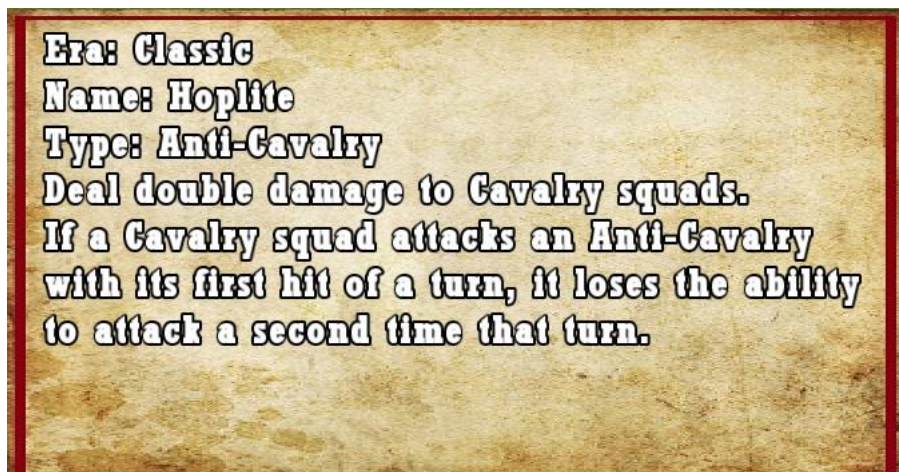


Рисунок 4.15 – Згенерована інформація про загін «Гоплітів»

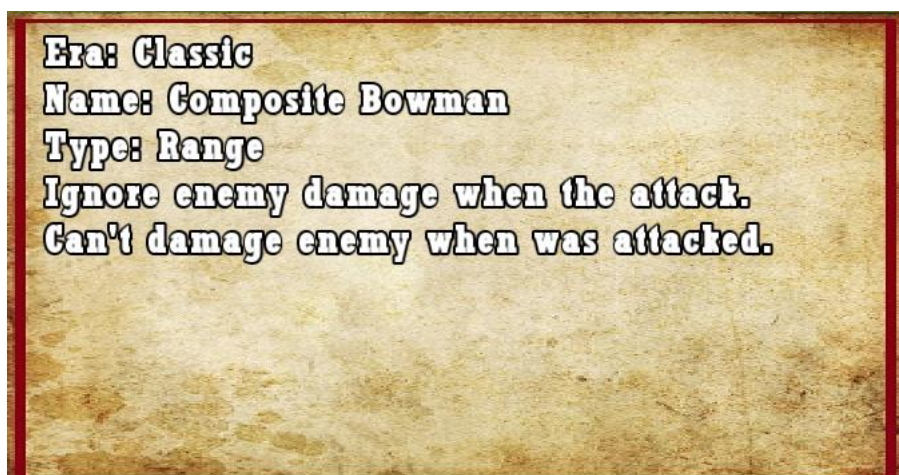


Рисунок 4.16 – Згенерована інформація про загін «Довголучників»

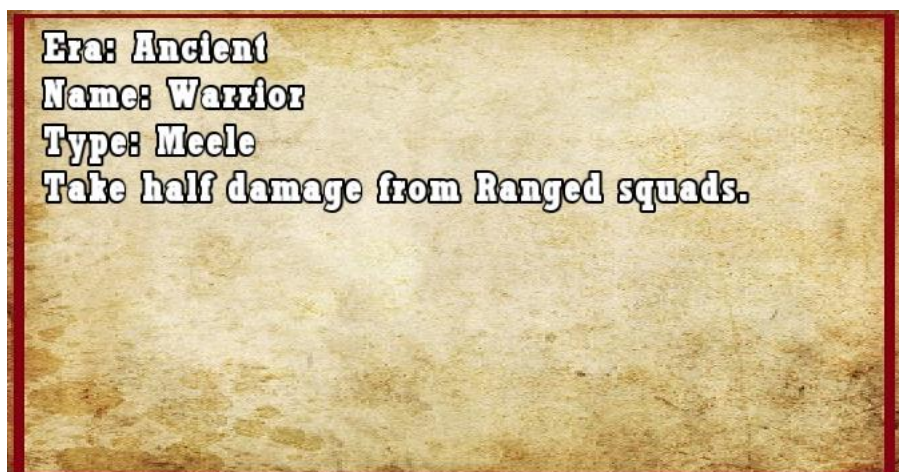


Рисунок 4.17 – Згенерована інформація про загін «Воїнів»

Нижче на рисунку 4.18 зображений приклад згенерованого поля битви між гравцями з військами конкретної епохи, яка відображається на їх території.



Рисунок 4.18 – Згенеровані армія епохи Стародавнього світу (знизу) проти армії епохи Класичної ери (зверху)

Так як максимальна кількість загонів у армії кожного гравця 7, то для зручного відображення усього війська були описані точні координати генерації загонів, що

залежать від їх теперішньої кількості. Відступи між загонами встановлені у 250 пікселів (для стандартного розміру екрану, а саме Full HD 1920/1080 пікселів).

Повний код проекту ігрового додатка «War of Ages: Battle for Castle» знаходиться за посиланням <https://github.com/KuzmenkoVladyslav/WOABFC>.

За допомогою засобів GitHub можна відстежити процес розробки даного додатку шляхом спостереження за commits та детальним описом, з яким вони були запушені до проекту.

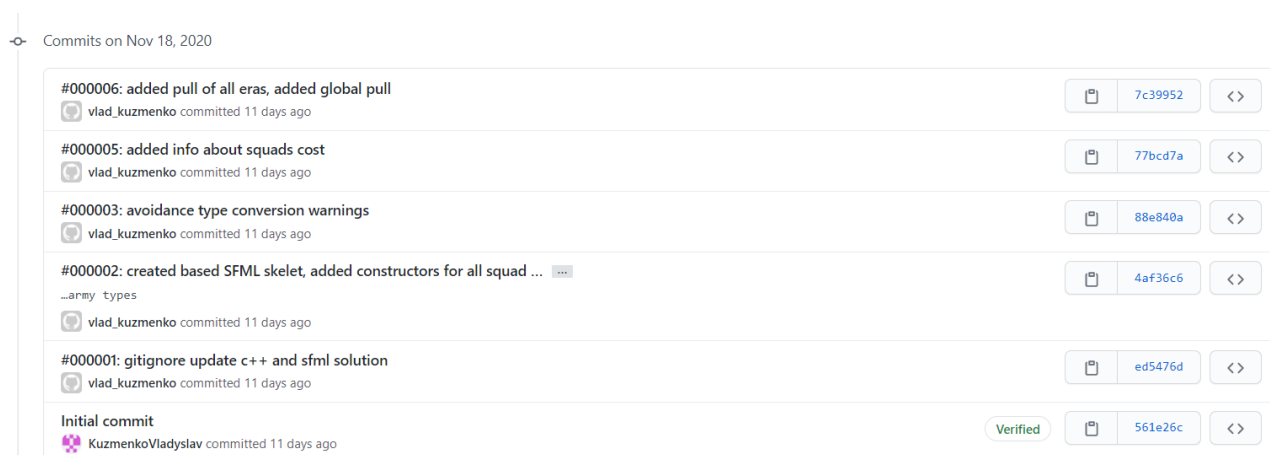


Рисунок 4.19 – Приклад перших commits та їх опису у GitHub репозиторії

4.6 Майбутній розвиток та наступні версії ігрового додатка

Першочерговим планом для розвитку є реалізація серверного підключення на власний або орендований сервер для одночасного ігрового процесу між усіма учасниками партії.

Серед інших планів можна виділити на покращення ігрового досвіду та, у цілому, геймплею, а саме:

- 1) Додати новий тип війська: укріплення, що будуть відволікати на себе ворожі загони доки не будуть знищені та не матимуть змоги нападати самотужки.
- 2) Реалізувати механіку полководців: гравець на початку партії обирає собі пасивний бонус полководця, що, наприклад, буде покращувати власну армію або погіршувати ворожу. Наприклад: всі союзні стрільці отримують +1 до шкоди або всі ворожі кавалеристи втрачають можливість атакувати двічі.
- 3) Розвивати та розповсюджувати ігровий додаток за допомогою платформ Steam та Epic Games Store для популяризації кіберспортивного напрямку.

ВИСНОВОК

Для виконання дипломної роботи магістерського проекту був проведений аналіз предметної області ігрових додатків, а саме було вивчено актуальну позицію ігрової індустрії у 2020 році та детально розібрані AAA-проекти трьох провідних світових розробників та видавців розважальних комп'ютерних ігор. Після цього були відібрані найкращі сторони кожного з проекту для компіляції та використання у власній грі. Не зважаючи на це мій проект не є плагіатом, тому що та концепція, яку він містить у собі є унікальною.

В аналітично-теоретичній частині дипломного магістерського проекту були описані задачі, методи та вимоги до розробки ігрового додатка і обґрунтований вибір програм та пристроїв для реалізації поставлених задач.

Після аналізу обраних засобів реалізації та програмних засобів було визначено такий набір інструментів для виконання магістерської роботи: мова програмування C++ з бібліотекою SFML для написання ігрового рушію, а також Adobe Photoshop CS6 як засіб для роботи з растровою графікою для створення спрайтів і зображень.

Після сформованої мети та визначення переліку задач, що треба виконати було сплановано список робіт задля виконання дипломного проекту. Для цього були створені структурна декомпозиція робіт для організації командної роботи, організаційна структура проекту для відображення учасників та їх безпосередніх керівників, тобто WBS і OBS, діаграма Ганта для розподілу людських ресурсів та часу згідно запланованим роботам і матриця відповідальності для дотримання структури виконання поставлених робіт. Були прораховані можливі ризики, а саме вірогідності і причини виникнення та визначені методи для їх усунення.

Було проведено проектування дипломного проекту ігрового додатка. Для передбачення помилок під час процесу забезпечення додатку були побудовані контекстна діаграма A-0 і її декомпозиція та діаграма варіантів використання (Use Case).

Процес реалізації ігрового додатка був розділений на 4 загальних етапи:

- процес інтелектуальної розробки або мозкового штурму для генерації оптимальних ігрових правил;
- процес розробки вторинних частин ігрового додатка, а саме файлів локалізації та растрових зображень;
- процес будування сцен ігрового додатка;
- процес написання програмного коду для реалізації описаних ігрових правил, пов'язання створених файлів локалізації та зображень з розробленими сценами.

Були описані плани на майбутнє по розвитку існуючого додатку та його потенційного впровадження.

СПИСОК ЛІТЕРАТУРИ

1. Уточкин В.А. Игровая индустрия: геймдев (gamedev) [Электронный ресурс] / Уточкин В.А. // Центр развития компетенций в бизнес-информатике высшей школы бизнеса. – 2020. – Режим доступа до ресурсу: <https://hsbi.hse.ru/articles/igrovaya-industriya-geymdev/>.
2. Clinton Keith. Agile Game Development: Build, Play, Repeat (2nd Edition) (Addison-Wesley Signature Series (Cohn)) 2nd Edition / Clinton Keith., 2020. – 576 с.
3. Gabor Szauer. Hands-On C++ Game Animation Programming: Learn modern animation techniques from theory to implementation with C++ and OpenGL / Gabor Szauer., 2020. – 368 с.
4. Claudio Torres. Ethereum Game Development Projects: Build your own line of blockchain-based crypto-strategic games / Claudio Torres., 2020. – 447 с.
5. Michael Dawson. Beginning C++ Through Game Programming / Michael Dawson., 2014. – 352 с.
6. Tarn Adams. Procedural Storytelling in Game Design 1st Edition / Tarn Adams, Tanya X. Short., 2019. – 408 с.
7. John Epstein. Hololens and Game Programming Kindle Edition / John Epstein., 2020. – 115 с.
8. Victor Brusca. Video Game UDP Client/Server Design and Implementation: With a cross platform, networked, example game / Victor Brusca, Brian Ree., 2020. – 843 с.
9. Michael Dawson. Beginning C++ Through Game Programming / Michael Dawson., 2020. – 432 с.
10. Fletcher Dunn. 3D Math Primer for Graphics and Game Development / Fletcher Dunn., 2011. – 846 с.

11. Robert Nystrom. Game Programming Patterns / Robert Nystrom., 2014. – 354 с.
12. Valve Corporation [Электронный ресурс] // Valve. – 2020. – Режим доступа до ресурсу: <https://www.valvesoftware.com/ru/>.
13. Bethesda Corporation [Электронный ресурс] // Microsoft. – 2020. – Режим доступа до ресурсу: <https://bethesda.net/ru/dashboard>.
14. Blizzard Corporation [Электронный ресурс] // Activision. – 2020. – Режим доступа до ресурсу: <https://www.blizzard.com/ru-ru/>.
15. Firaxis Games Developer [Электронный ресурс] // Firaxis Games. – 2020. – Режим доступа до ресурсу: <https://firaxis.com/>.
16. Hearthstone — стратегічна карткова гра [Электронный ресурс] // Blizzard Entertainment. – 2020. – Режим доступа до ресурсу: <https://playhearthstone.com/ru-ru>.
17. Hearthstone Battlegrounds [Электронный ресурс] // Blizzard Entertainment. – 2020. – Режим доступа до ресурсу: <https://playhearthstone.com/ru-ru/news/23156373>.
18. Civilization VI [Электронный ресурс] // 2k Games and Firaxis Games. – 2020. – Режим доступа до ресурсу: <https://www.civilization.com/>.
19. War games shed light on real-world strategies [Электронный ресурс] // theconversation. – 2019. – Режим доступа до ресурсу: <https://theconversation.com/war-games-shed-light-on-real-world-strategies-113631>.
20. The Curious Case of 4X Games, Efficiency Engines, and Missing Strategic Gambits [Электронный ресурс] // big-game-theory.com. – 2020. – Режим доступа до ресурсу: <http://www.big-game-theory.com/2020/01/the-curious-case-of-4x-games-efficiency.html>.
21. Dota Underlords [Электронный ресурс] // Valve. – 2020. – Режим доступа до ресурсу: <https://www.underlords.com/>.
22. Simple and Fast Multimedia Library [Электронный ресурс] // Laurent Gomila. – 2020. – Режим доступа до ресурсу: <https://www.sfml-dev.org/>.

23. Unity, платформа разработки [Электронный ресурс] // Unity Technologies. – 2020. – Режим доступа до ресурсу: <https://unity.com/ru>.
24. Редактор растрової графіки Adobe Photoshop [Электронный ресурс] // Adobe. – 2020. – Режим доступа до ресурсу: <https://www.adobe.com/ua/products/photoshop.html>.
25. Make something Unreal [Электронный ресурс] // Epic Games. – 2020. – Режим доступа до ресурсу: <https://www.unrealengine.com/en-US/>.
26. What is SMART in Project Management? [Электронный ресурс] // wrike. – 2020. – Режим доступа до ресурсу: <https://www.wrike.com/project-management-guide/faq/what-is-smart-in-project-management/>.
27. Work Breakdown Structure, WBS [Электронный ресурс] // uplandsoftware. – 2020. – Режим доступа до ресурсу: <https://uplandsoftware.com/psa/resources/glossary/work-breakdown-structure-wbs/>.
28. Organization Breakdown Structure (OBS) [Электронный ресурс] // uplandsoftware. – 2020. – Режим доступа до ресурсу: <https://uplandsoftware.com/psa/resources/glossary/organization-breakdown-structure-obs/>.
29. The RACI matrix: Your blueprint for project success [Электронный ресурс] // CIO United States. – 2018. – Режим доступа до ресурсу: <https://www.cio.com/article/2395825/project-management-how-to-design-a-successful-raci-project-plan.html>.
30. What Is a Gantt Chart? [Электронный ресурс] // teamgantt. – 2020. – Режим доступа до ресурсу: <https://www.teamgantt.com/what-is-a-gantt-chart>.
31. The Risk Management Process in Project Management [Электронный ресурс] // projectmanager. – 2017. – Режим доступа до ресурсу: <https://www.projectmanager.com/blog/risk-management-process-steps>.

32. Project management in manufacturing using IDEF0 [Електронний ресурс] // sciencedirect. – 2020. – Режим доступу до ресурсу: <https://www.sciencedirect.com/science/article/abs/pii/026378639190041S>.
33. Functional and Information Modeling of Production Using IDEF Methods [Електронний ресурс] // Journal of Mechanical Engineering. – 2019. – Режим доступу до ресурсу: <http://www.macs.hw.ac.uk/~yjc32/project/ref-IDEF0/IDEF0%20family.pdf>.
34. Use cases what every project manager should know [Електронний ресурс] // Project Management Institute. – 2020. – Режим доступу до ресурсу: <https://www.pmi.org/learning/library/use-cases-project-manager-know-8262>.
35. USE CASE DIAGRAMS: A PM'S VIEW [Електронний ресурс] // projectsmart. – 2019. – Режим доступу до ресурсу: <https://www.projectsmart.co.uk/use-case-diagrams-a-pm-view.php>.
36. ІМА 2020. Інформатика, Автоматика, Математика / матеріали та програма Науково-Технічної Конференції / – Суми: Сумський державний університет, 2020. – С. 174.

ДОДАТОК А.

ПЛАНУВАННЯ РОБІТ

Ідентифікація мети реалізації ігрового додатку. Для конкретизації мети дипломного проекту на основі певних показників був використаний метод SMART, що визначає такі показники постановки: конкретність, вимірюваність, досяжність, реалістичність та обмеженість у часі (табл. А.1) [26].

Таблиця А.1 Деталізація мети методом SMART

S (Конкретність)	Розробити багато користувальницький ігровий додаток у жанрі стратегії для кіберспортивного суперництва.
M (Вимірюваність)	Створити функціональні можливості для виконання правил гри, ігрових механік та взаємодії гравців між собою.
A (Досяжність)	Бюджет проекту складає 528 200 грн.
R (Доцільність)	Індустрія відеоігор є актуальною та приносить високі прибутки.
T (Обмеженість в часі)	Проект буде виконано та здано у експлуатацію вчасно, що підтверджується календарним планом проекту.

Планування змісту структури робіт ІТ-проекту. Структурна декомпозиція робіт (work breakdown structure, WBS) - у менеджменті та системотехніці є орієнтованою на повне завершення проекту та детермінованою декомпозицією проекту на менші частки. Є ключовою часткою робіт по проекту, яка організовує командну роботу [27]. Структурна декомпозиція робіт проекту представлена на рис. А.1.

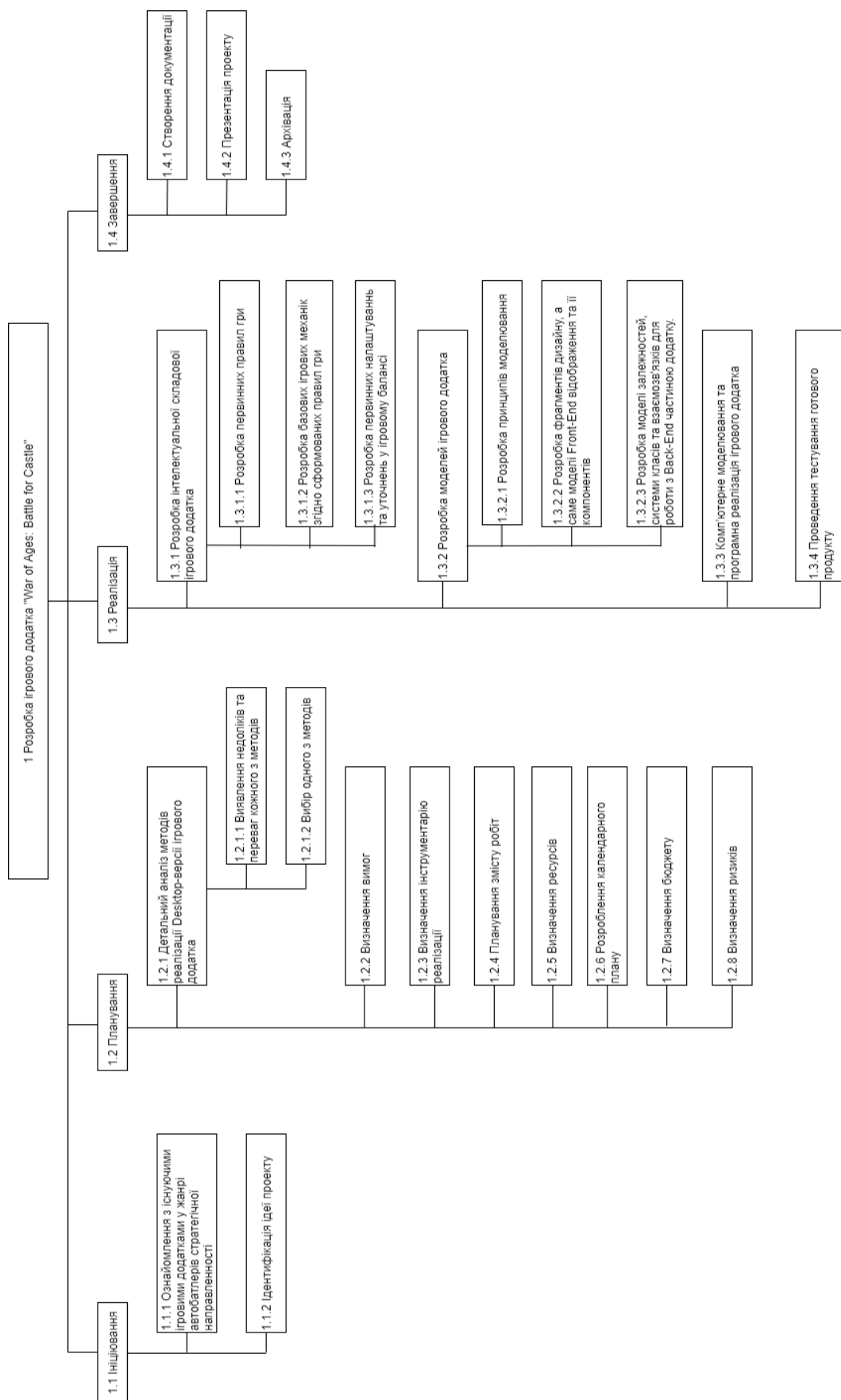


Рисунок А.1 – WBS структура проекту

Організаційна структура проекту (OBS). Організаційна структура проекту (OBS) – це графічне відображення учасників проекту та їхніх відповідальних осіб, що були залучені до реалізації проекту. На верхньому рівні OBS проекту знаходиться керівник та команда управління проектом; на наступному рівні – виконавці [28]. Останнім рівнем OBS-структури є відповідальні особи виконавців. OBS структура представлена на рисунку А.2.

Побудова матриці відповідальності. Матриця відповідальності (Responsibility Assignment Matrix) забезпечує опис і узгодження структури відповідальності за виконання пакетів робіт. Вона являє собою форму опису розподілу відповідальності за реалізацію робіт проекту із зазначенням ролі кожного з виконавців [29]. На рисунку А.3 показано матрицю відповідальності проекту.

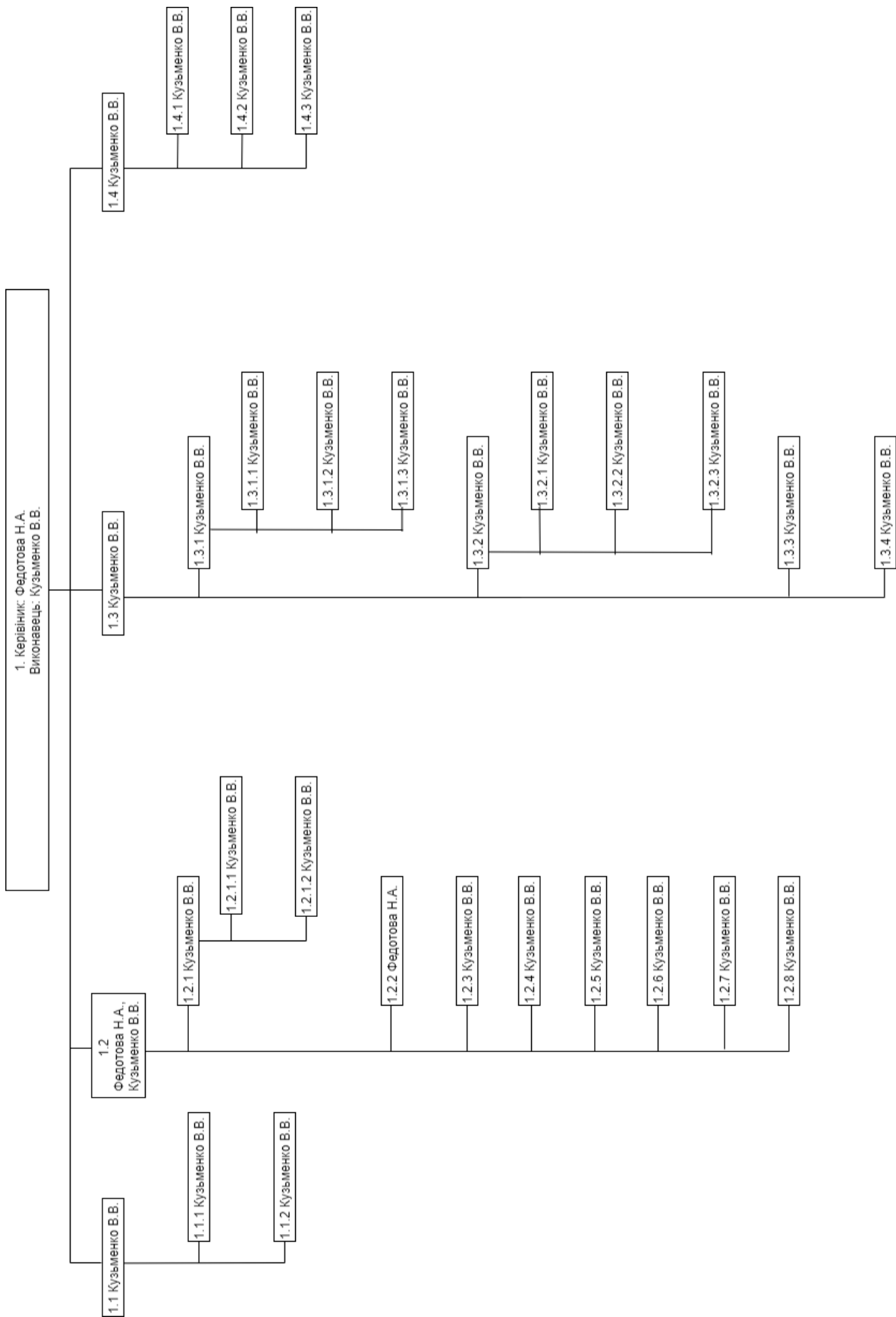


Рисунок А.2 – OBS структура

	СР1.1		СР1.2								СР1.3						СР1.4			
	СР1.1.1	СР1.1.2	СР1.2.1	СР1.2.2	СР1.2.3	СР1.2.4	СР1.2.5	СР1.2.6	СР1.2.7	СР1.2.8	СР1.3.1		СР1.3.2		СР1.3.3	СР1.3.4	СР1.4.1	СР1.4.2	СР1.4.3	
			СР1.2.1.1	СР1.2.1.2							СР1.3.1.1	СР1.3.1.2	СР1.3.1.3	СР1.3.2.1	СР1.3.2.2	СР1.3.2.3				
			+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Кузьменко В.В.																				
Федотова Н.А.				+																

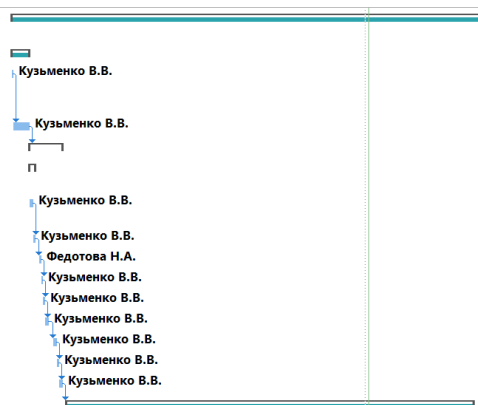
Рисунок А.3 – Матриця відповідальності

Побудова календарного графіка виконання ІТ-проекту. Діаграма Ганта - це вид діаграми, що використовується для планування і контролю виконання проекту. Такий мережевий графік присутній практично у всіх СУ проектами. На діаграмі відображаються завдання і стадії проекту з урахуванням їх часу виконання. Завдання на діаграмі можуть бути залежними один від одного або незалежними [30].

Для побудови діаграми Ганта використовуємо програму MS Office Project 2020 року.

На рисунку А.4 показана діаграма Ганта.

1 Розробка ігрового додатку "War of Ages: Battle for Castle"	175 днів	Пн 27.04.20	Чт 24.12.20	Кузьменко В.В., Федотова Н.А.
1.1 Ініціювання	7 днів	Пн 27.04.20	Вт 05.05.20	Кузьменко В.В.
1.1.1 Ознайомлення з існуючими ігровими додатками у жанрі автобатлерів стратегічної направленості	1 день	Пн 27.04.20	Пн 27.04.20	Кузьменко В.В.
1.1.2 Ідентифікація ідей проекту	6 днів	Вт 28.04.20	Вт 05.05.20	3 Кузьменко В.В.
1.2 Планування	13 днів	Ср 06.05.20	Пт 22.05.20	4 Кузьменко В.В.
1.2.1 Детальний аналіз методів реалізації Desktop-версії ігрового додатка	3 днів	Ср 06.05.20	Пт 08.05.20	Кузьменко В.В.
1.2.1.1 Виявлення недоліків та переваг кожного з методів	2 днів	Ср 06.05.20	Чт 07.05.20	Кузьменко В.В.
1.2.1.2 Вибір одного з методів	1 день	Пт 08.05.20	Пт 08.05.20	7 Кузьменко В.В.
1.2.2 Визначення вимог	1 день	Пн 11.05.20	Пн 11.05.20	8 Федотова Н.А.
1.2.3 Визначення інструментарію реалізації	1 день	Вт 12.05.20	Вт 12.05.20	9 Кузьменко В.В.
1.2.4 Планування змісту робіт	1 день	Ср 13.05.20	Ср 13.05.20	10 Кузьменко В.В.
1.2.5 Визначення ресурсів	2 днів	Чт 14.05.20	Пт 15.05.20	11 Кузьменко В.В.
1.2.6 Розроблення календарного плану	2 днів	Пн 18.05.20	Вт 19.05.20	12 Кузьменко В.В.
1.2.7 Визначення бюджету	1 день	Ср 20.05.20	Ср 20.05.20	13 Кузьменко В.В.
1.2.8 Визначення ризиків	2 днів	Чт 21.05.20	Пт 22.05.20	14 Кузьменко В.В.
1.3 Реалізація	150 днів	Пн 25.05.20	Чт 17.12.20	15 Кузьменко В.В.



1.3.1 Розробка інтелектуальної складової ігрового додатка	45 днів	Пн 25.05.20	Чт 23.07.20	Кузьменко В.В.
1.3.1.1 Розробка первинних правил гри	15 днів	Пн 25.05.20	Пт 12.06.20	Кузьменко В.В.
1.3.1.2 Розробка базових ігрових механік згідно сформованих правил гри	15 днів	Пн 15.06.20	Чт 02.07.20	18 Кузьменко В.В.
1.3.1.3 Розробка первинних налаштувань та уточнень у ігровому балансі	15 днів	Пт 03.07.20	Чт 23.07.20	19 Кузьменко В.В.
1.3.2 Розробка моделей ігрового додатка	45 днів	Пт 24.07.20	Чт 24.09.20	20 Кузьменко В.В.
1.3.2.1 Розробка принципів моделювання	15 днів	Пт 24.07.20	Чт 13.08.20	Кузьменко В.В.
1.3.2.2 Розробка фрагментів дизайну, а саме моделі Front-End відображення та її компонентів	15 днів	Пт 14.08.20	Чт 03.09.20	22 Кузьменко В.В.
1.3.2.3 Розробка моделі залежностей, системи класів та взаємозв'язків для роботи з Back-End частиною додатку.	15 днів	Пт 04.09.20	Чт 24.09.20	23 Кузьменко В.В.
1.3.3 Комп'ютерне моделювання та програмна реалізація ігрового додатка	55 днів	Пт 25.09.20	Чт 10.12.20	24 Кузьменко В.В.
1.3.4 Проведення тестування готового продукту	5 днів	Пт 11.12.20	Чт 17.12.20	25 Кузьменко В.В.
1.4 Завершення	5 днів	Пт 18.12.20	Чт 24.12.20	26 Кузьменко В.В.
1.4.1 Створення документації	2 днів	Пт 18.12.20	Пн 21.12.20	Кузьменко В.В.
1.4.2 Презентація проекту	2 днів	Вт 22.12.20	Ср 23.12.20	28 Кузьменко В.В.
1.4.3 Архівація	1 день	Чт 24.12.20	Чт 24.12.20	29 Кузьменко В.В.



Рисунок А.4 – Діаграма Ганта

Управління ризиками. Виконаємо якісну і кількісну оцінку ризиків роботи. При якісній оцінці визначимо ризики, що потребують швидкого реагування. Така оцінка визначить ступінь важливості ризику і дозволить вибрати спосіб реагування. Кількісна оцінка ризиків буде виконана для більш повної ідентифікації ризиків та ступеня їхнього впливу на виконання проекту. Кількісна і якісна оцінка ризиків можуть використовуватися окремо або разом, залежно від наявного часу і бюджету, необхідності в кількісній або якісній оцінці ризиків [31].

Визначимо ризики:

- R1 – зміна вимог при реалізації проекту;
- R2 – невчасне виконання етапів проекту;
- R3 – виявлення помилок на етапі завершення;
- R4 – виникнення несправностей в апаратному забезпеченні;
- R5 – непередбачувані життєві обставини розробника.

Ймовірність виникнення ризиків:

Таблиця А.2 – Ймовірності виникнення ризиків

Ймовірність виникнення	R1	R2	R3	R4	R5
Дуже низька	+				
Низька		+	+		+
Середня					
Висока				+	
Дуже висока					

Таблиця А.3 – Значимість впливу ризиків

Значимість впливу	R1	R2	R3	R4	R5
мінімальна					
низька					+
середня					
висока	+	+		+	
максимальна			+		

Таблиця А.4 – Втрати при виникненні ризиків

Втрати			R3	R4	
		R2			
	R1				
					R5
	Ймовірність				

Виходячи з цього, було визначено чотири критичних ризики, такі як:

- R1 – зміна вимог при реалізації проекту;
- R2 – невчасне виконання етапів проекту;
- R3 – виявлення помилок на етапі завершення;
- R4 – виникнення несправностей в апаратному забезпеченні.

Таблиця А.5 – Варіанти запобігання та реакції на виникнення ризиків

Ризик	Наслідки та способи уникнення
Зміна вимог при реалізації проекту	Своєчасне визначення вимог проекту з керівником
Невчасне виконання етапів проекту	Розподіл свого часу та ресурсів заздалегідь
Виявлення помилок на етапі завершення	Найкритичніший ризик, у разі його виникнення на виправлення помилок піде багато часу
Виникнення несправностей в апаратному забезпеченні	Сприяє некоректній роботі системі, тому треба своєчасно подбати про справний персональний комп'ютер

ДОДАТОК Б.

ПРАВИЛА ГРИ

Правила гри та ігрова механіка ігрового додатку

Першим етапом розробки було формування правил та концепції гри. Правила поділяються на загальну механіку гри, механіку розвитку своїх володінь, механіку купівлі війська та загальні особливості кожного з військових загонів у відповідності до історичних епох.

Загальна механіка гри

1. Стандартна ігрова партія розрахована на 8 гравців.
2. На початку гри у розпорядженні кожного гравця є свій палац з визначеною та однаковою для всіх кількістю балів життя рівною 50.
3. На кожному ході гравці мають деяку кількість балів дій, що залежить від епохи до якої вони розвинулися. Більш докладно про це у таблиці Б.1.
4. Бали дій можна витратити на купівлю військових загонів, укріплень, розширення комірок для загонів і укріплень та на перехід у наступну еру.
5. Вартість кожної дії, крім купівлі переходу у наступну еру та збільшення кількості комірок – 1 бал дій.
6. Невитрачені бали дій не зберігаються на наступний хід.

Механіка розвитку своїх володінь

1. Усього у грі рахуючи стартову 9 епох. У кожній епосі є свої види військ, що відповідають їй. Список епох та їх бонуси представлені у розділі 4.3.
2. Максимальна вартість переходу у наступну епоху зазначена у таблиці Б.1. Вона зменшується кожного ходу на один.
3. На початку гри та після кожної битви усім гравцям пропонуються на вибір 4 категорії для покупки. Це можуть бути війська будь яких типів або укріплення.
4. У кожній епосі є визначений шанс на знаходження у магазині війська тієї чи іншої епохи.
5. Укріплення завжди відповідають тій епосі, у якій знаходиться гравець.
6. На початку гри кожен гравець у своєму розпорядженні має лише 3 комірки для розміщення військ та укріплень для майбутнього бою.
7. Для збільшення кількості комірок необхідно їх придбати. Кожна додаткова комірка потрапляє у магазин після виходу у парну епоху. Вартість додаткової комірки рівна 5 – КІЛЬКІСТЬ ХОДІВ, У ЯКІ ГРАВЕЦЬ МІГ ПРИДБАТИ ДОДАТКОВУ КОМІРКУ, АЛЕ НЕ ЗРОБИВ ЦЬОГО балів дій.
8. У замку кожного гравця є так звана лава запасних, у якій гравець може залишати війська чи укріплення, які поки що не хоче або не може виставити для битви. Її місткість рівна НОМЕР ЕПОХИ / 2 + 3 комірок на лаві запасних.

Механіка купівлі війська

1. Будь який загін будь якого типу війська будь якої епохи коштує рівно 1 бал дій якщо не вказано інакше.
2. Усі загони військ знаходяться в одному загальному наборі з якого потім потрапляють у магазини всіх гравців. Тобто якщо Гравець 1 придбає один загін

конкретного війська, то доки він його не демобілізує він не буде доступним для інших гравців. Наприклад це не значить, що він забере у всіх гравців усіх Лучників епохи Стародавнього світу, це значить, що він забере лише 1 з усіх доступних екземплярів цієї епохи.

3. Продаж загону війська неможливий, можна тільки демобілізувати його, тобто прибрати зі свого війська та лави запасних. Витрачені при купівлі бали дій не повертаються.

4. Придбані 2 однакових загони перетворюються у 1 загін війська наступної епохи, але тільки якщо гравець вже перейшов в неї і тільки якщо не сказано інакше. Наприклад, два загони Пікінерів або два загони Лицарів з епохи Середньовіччя не об'єднуються для покращення, а ось Пікінер та Лицар стануть Уланом (якщо відкрита ера Відродження, звісно).

Особливості війська

1. Війська бувають чотирьох типів: ближнього бою, анти кавалерійські, кавалерія та стрільці.

2. Кожен загін має параметри шкоди, здоров'я та своє особливе уміння, що залежить від типу війська.

3. Війська ближнього бою отримують в два рази менше шкоди від стрільців.

4. Стрільці атакуючи ворога не отримують шкоди.

5. Кавалерія має змогу атакувати двічі за одну свою чергу ходити.

6. Анти кавалерійські війська наносять кавалерії ворога у два рази більше шкоди та якщо кавалерійський загін напав на анти кавалерійський загін на першому своєму нападі за хід, то він втрачає можливість атакувати другий раз на цьому ході.

7. Укріплення є одноразовими, тобто приймають участь у битві лише один раз і після цього зникають потрапляючи у загальний набір військ. Також укріплення

не можуть поєднуватись в укріплення наступної епохи так як їх характеристики завжди рівні теперішній епосі гравця.

Механіка бою

1. Під час битви кожен загін одного гравця по черзі б'ється з випадковим загоном іншого гравця з передачею ходу.
2. Удари починаються зліва направо для всього війська кожного гравця.
3. Право першого удару віддається за таким пріоритетом:
 - 1) Вища епоха;
 - 2) Більша кількість військових загонів з урахуванням укріплень;
 - 3) Більша кількість військових загонів без урахування укріплень;
 - 4) Поразка у минулому раунді;
 - 5) Кидок монетки, де перший удар віддається випадковому гравцю з вірогідністю 50%.
4. Під час нападу одного загону на інший кожен з цих двох загонів отримує шкоду у розмірі параметру шкоди опонента та втрачає стільки ж балів життя (не рахуючи бонуси або штрафи типів військ).
5. Стрільці не можуть наносити шкоду ворогу захищаючись від нападу.
6. Укріплення не можуть здійснювати напад, а лише захищатись.
7. Противники зобов'язані атакувати ворожі укріплення доки вони не будуть знищені.
8. Бій відбувається доки один з учасників битви не втратить все своє військо.
9. Гравець, що переміг у битві наносить шкоду ворожому замку у розмірі НОМЕР ЕПОХИ ПЕРЕМОЖЦЯ + $\frac{1}{2}$ * НОМЕР ЕПОХИ ПЕРЕМОЖЕНОГО + КІЛЬКІСТЬ ЗАГОНІВ ВІЙСЬК ПЕРЕМОЖЦЯ, ЩО ВИЖИЛА ПІСЛЯ БОЮ.
10. Якщо у гравця не залишилося балів життя у його замку, то він вибуває.

11. Після бою все втрачене військо крім укріплень відновлюється у кожного з гравців.
12. Після кожного бою гравці потрапляють у магазин з оновленими загонами військ.
13. Гра продовжується циклом магазин-битва-магазин, доки 7 гравців не вибудуть.
14. Перемагає гравець, що залишився останнім не вибулим.

Конкретна інформація, параметри, характеристики та налаштування балансу у ігровому додатку

Загальні параметри ігрових епох

Кожна епоха має відповідати не тільки своєму історичному аналогу, але й влаштовувати гравців у плані балансу гри. Саме тому велика частина роботи – це налаштування ігрових параметрів.

Таблиця Б.1 – Загальна інформація про епохи у грі

Номер епохи у грі	Назва епохи	Початкова ціна переходу у цю епоху	Кількість дій, що даються на один хід у епосі
0	Стародавній світ	Стартова епоха	3
1	Античність	4	3
2	Середньовіччя	5	3
3	Відродження	5	4
4	Промислова ера	5	4
5	Новий час	6	4
6	Епоха атому	6	5
7	Інформація ера	7	5
8	Майбутнє	8	6

Епоха «Стародавній світ»

Епоха, з якої стартує кожен гравець. В магазині у даній епосі зі 100% вірогідністю гравці будуть знаходити загоні військ лише епохи стародавнього світу або укріплення. Види військ даної епохи: воїни, лучники, колісниця та списоносець.

Воїн – загін війська ближнього бою епохи Стародавнього світу (рис. Б.1).



Рисунок Б.1 – Воїн епохи «Стародавнього світу»

Параметри:

- 1) Шкода: 1
- 2) Життя: 3
- 3) Покращується: два Воїни у одного Мечника.

Лучник – загін війська дальнього бою епохи Стародавнього світу (рис. Б.2).



Рисунок Б.2 – Лучник епохи «Стародавнього світу»

Параметри:

- 1) Шкода: 1
- 2) Життя: 1
- 3) Покращується: два Лучники у одного Довголучника.

Колісниця – загін війська кавалерії епохи Стародавнього світу (рис. Б.3).



Рисунок Б.3 – Колісниця епохи «Стародавнього світу»

Параметри:

- 1) Шкода: 1
- 2) Життя: 3
- 3) Покращується: дві Колісниці у одного Вершника.

Списоносець – загін анти кавалерійського війська епохи Стародавнього світу (рис. Б.4).



Рисунок Б.4– Списоносець епохи «Стародавнього світу»

Параметри:

- 1) Шкода: 1
- 2) Життя: 2
- 3) Покращується: два Списоносці у одного Гопліта.

Епоха «Античність»

Друга епоха для кожного гравця. В магазині у даній епосі з 50% вірогідністю гравці будуть знаходити загони військ епохи стародавнього світу та з 50% вірогідністю загони військ епохи Античності або укріплення. Види військ даної епохи: Мечник, Довголучник, Вершник та Гопліт.

Мечник – загін війська ближнього бою епохи Античності (рис. Б.5).



Рисунок Б.5 – Мечник епохи «Античності»

Параметри:

- 1) Шкода: 2
- 2) Життя: 4
- 3) Покращується: два Мечники у одного Довгомечника.

Довголучник – загін війська дальнього бою епохи Античності (рис. Б.6).



Рисунок Б.6 – Довголучник епохи «Античності»

Параметри:

- 1) Шкода: 2
- 2) Життя: 1
- 3) Покращується: два Довголучники у одного Арбалетника.

Вершник – загін війська *кавалерії* епохи Античності (рис. Б.7).



Рисунок Б.7 – Вершник епохи «Античності»

Параметри:

- 1) Шкода: 1
- 2) Життя: 5
- 3) Покращується: два Вершники у одного Лицаря.

Гопліт – загін анти кавалерійського війська епохи Античності (рис. Б.8).



Рисунок Б.8 – Гопліт епохи «Античності»

Параметри:

- 1) Шкода: 2
- 2) Життя: 3
- 3) Покращується: два Гопліти у одного Пікінера.

Епоха «Середньовіччя»

Третя епоха для кожного гравця. В магазині у даній епосі з 25% вірогідністю гравці будуть знаходити загони військ епохи стародавнього світу, з 25% вірогідністю загони військ епохи Античності та з 50% вірогідністю загони військ епохи Середньовіччя або укріплення. Види військ даної епохи: Довгомечник, Арбалетник, Лицар та Пікінери.

Довгомечник – загін війська ближнього бою епохи Середньовіччя (рис. Б.9).



Рисунок Б.9 – Мечник епохи «Середньовіччя»

Параметри:

- 1) Шкода: 2
- 2) Життя: 7
- 3) Покращується: два Довгомечники у одного Аркебузира.

Арбалетник – загін війська дальнього бою епохи Середньовіччя (рис. Б.10).



Рисунок Б.10 – Арбалетник епохи «Середньовіччя»

Параметри:

- 1) Шкода: 2
- 2) Життя: 3
- 3) Покращується: два Арбалетники у одну Гармату.

Лицар – загін війська *кавалерії* епохи Середньовіччя (рис. Б.11).



Рисунок Б.11 – Вершник епохи «Середньовіччя»

Параметри:

- 1) Шкода: 2
- 2) Життя: 6
- 3) Покращується: один Лицар та один Пікінер у одного Улана.

Пікінер – загін анти кавалерійського війська епохи Середньовіччя (рис. Б.12).



Рисунок Б.12 – Пікінер епохи «Середньовіччя»

Параметри:

- 1) Шкода: 2
- 2) Життя: 5
- 3) Покращується: один Лицар та один Пікінер у одного Улана.

Епоха «Відродження»

Четверта епоха для кожного гравця. В магазині у даній епосі з 25% вірогідністю гравці будуть знаходити загони військ епохи Античності, з 25% вірогідністю загони військ епохи Середньовіччя та з 50% вірогідністю загони військ епохи Відродження або укріплення. Види військ даної епохи: Аркебузир, Гармата та Улан.

Аркебузир – загін війська ближнього бою епохи Відродження (рис. Б.13).



Рисунок Б.13 – Аркебузир епохи «Відродження»

Параметри:

- 1) Шкода: 3
- 2) Життя: 8
- 3) Покращується: два Аркебузири у одного Стрільця.

Гармата – загін війська дальнього бою епохи Відродження (рис. Б.14).



Рисунок Б.14 – Гармата епохи «Відродження»

Параметри:

- 1) Шкода: 3
- 2) Життя: 4
- 3) Покращується: дві Гармати у один кулемет Гатлінга.

Улан – загін війська кавалерії, що спеціалізується на протидії іншій кавалерії епохи Відродження (рис. Б.15).



Рисунок Б.15 – Улан епохи «Відродження»

Параметри:

- 1) Шкода: 3
- 2) Життя: 9
- 3) Покращується: ні.

Епоха «Промислова ера»

П'ята епоха для кожного гравця. В магазині у даній епосі з 25% вірогідністю гравці будуть знаходити загони військ епохи Середньовіччя, з 25% вірогідністю загони військ епохи Відродження та з 50% вірогідністю загони військ епохи

Промислової ери або укріплення. Види військ даної епохи: Стрілець, кулемет Гатлінга та Кавалерист.

Стрілець – загін війська ближнього бою епохи Промислова ера (рис. Б.16).



Рисунок Б.16 – Стрілець епохи «Промислова ера»

Параметри:

- 1) Шкода: 3
- 2) Життя: 11
- 3) Покращується: два Стрільці у одного Піхотинця.

Кулемет Гатлінга – загін війська дальнього бою епохи Промислового ера (рис. Б.17).



Рисунок Б.17 – Кулемет Гатлінга епохи «Промислового ера»

Параметри:

- 1) Шкода: 4
- 2) Життя: 4
- 3) Покращується: два Арбалетники у одну Гармату.

Кавалерист – загін війська кавалерії епохи Промислового ера (рис. Б.18).



Рисунок Б.18 – Кавалерист епохи «Промислового ера»

Параметри:

- 1) Шкода: 4
- 2) Життя: 9
- 3) Покращується: два Кавалеристи у одну Танкетку.

Епоха «Новий час»

Шоста епоха для кожного гравця. В магазині у даній епосі з 25% вірогідністю гравці будуть знаходити загони військ епохи Відродження, з 25% вірогідністю загони військ епохи Промислового ера та з 50% вірогідністю загони військ епохи Нового часу або укріплення. Види військ даної епохи: Піхотинець, Кулеметна рота, Танкетка та Артилерійська батарея.

Піхотинець – загін війська ближнього бою епохи Новий час (рис. Б.19).



Рисунок Б.19 – Піхотинець епохи «Новий час»

Параметри:

- 1) Шкода: 4
- 2) Життя: 12

3) Покращується: два Піхотинці у одного Морського піхотинця.

Кулеметна рота – загін війська дальнього бою епохи Новий час (рис.Б.20).



Рисунок Б.20 – Кулеметна рота епохи «Новий час»

Параметри:

1) Шкода: 6

2) Життя: 5

3) Покращується: дві Кулеметні роти в одну Базуку.

Танкетка – загін війська кавалерії епохи Новий час (рис. Б.21).

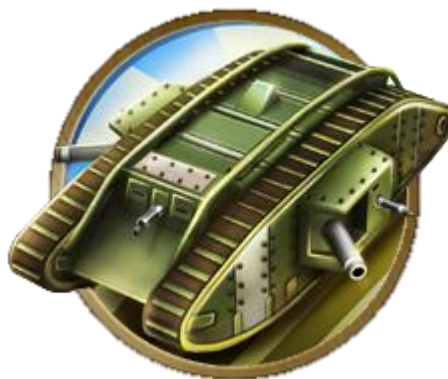


Рисунок Б.21 – Танкетка епохи «Новий час»

Параметри:

- 1) Шкода: 4
- 2) Життя: 14
- 3) Покращується: дві Танкетки у один Танк.

Артилерійська батарея – загін анти кавалерійського війська епохи Новий час (рис. Б.22).



Рисунок Б.22 – Артилерійська батарея епохи «Новий час»

Параметри:

- 1) Шкода: 6
- 2) Життя: 10
- 3) Покращується: дві Артилерійські батареї у одну Протитанкову гармату.

Епоха «Епоха атому»

Сьома епоха для кожного гравця. В магазині у даній епосі з 25% вірогідністю гравці будуть знаходити загони військ епохи Промислового ера, з 25% вірогідністю загони військ епохи Нового часу та з 50% вірогідністю загони військ епохи Епоха Атому або укріплення. Види військ даної епохи: Морський піхотинець, Базука, Танк та Протитанкова гармата.

Морський піхотинець – загін війська ближнього бою епохи Епоха Атому (рис. Б.23).



Рисунок Б.23 – Морський піхотинець епохи «Епоха Атому»

Параметри:

- 1) Шкода: 5

2) Життя: 14

3) Покращується: два Морських піхотинці у одну Мотопіхоту.

Базука – загін війська дальнього бою епохи Епоха Атому (рис. Б.24).



Рисунок Б.24 – Базука епохи «Епоха Атому»

Параметри:

1) Шкода: 8

2) Життя: 6

3) Покращується: дві Базуки в одну Керовану ракету.

Танк – загін війська кавалерії епохи Епоха Атому (рис. Б.25).



Рисунок Б.25 – Танк епохи «Епоха Атому»

Параметри:

- 1) Шкода: 4
- 2) Життя: 17
- 3) Покращується: два Танки у один Сучасний танк.

Протитанкова гармата – загін анти кавалерійського війська епохи Епоха Атому (рис. Б.26).



Рисунок Б.26 – Протитанкова гармата епохи «Епоха Атому»

Параметри:

- 1) Шкода: 8
- 2) Життя: 11
- 3) Покращується: дві Протитанкові гармати у одну Ракетку Артилерію.

Епоха «Інформаційна ера»

Восьма епоха для кожного гравця. В магазині у даній епосі з 25% вірогідністю гравці будуть знаходити загони військ епохи Нового часу, з 25% вірогідністю загони військ епохи Епоха Атому та з 50% вірогідністю загони військ епохи Інформаційна ера або укріплення. Види військ даної епохи: Мотопіхота, Керована ракета, Сучасний танк та Ракетна артилерія.

Мотопіхота – загін війська ближнього бою епохи Інформаційна ера (рис. Б.27).



Рисунок Б.27 – Мотопіхота епохи «Інформаційна ера»

Параметри:

- 1) Шкода: 7
- 2) Життя: 17

3) Покращується: дві Мотопіхоти в одну Екзоскелетну піхоту.

Керована ракета – загін війська дальнього бою епохи Інформаційна ера (рис. Б.28).



Рисунок Б.28 – Керована ракета епохи «Інформаційна ера»

Параметри:

- 1) Шкода: 11
- 2) Життя: 8
- 3) Покращується: ні.

Сучасний танк – загін війська кавалерії епохи Інформаційна ера (рис. Б.29).



Рисунок Б.29 – Сучасний танк епохи «Інформаційна ера»

Параметри:

- 1) Шкода: 6
- 2) Життя: 23
- 3) Покращується: ні.

Ракетна артилерія – загін анти кавалерійського війська епохи Інформаційна ера (рис. Б.30).



Рисунок Б.30 – Ракетна артилерія епохи «Інформаційна ера»

Параметри:

- 1) Шкода: 12
- 2) Життя: 17
- 3) Покращується: ні.

Епоха «Майбутнє»

Дев'ята і остання епоха для кожного гравця. В магазині у даній епосі з 25% вірогідністю гравці будуть знаходити загони військ епохи Інформаційна ера та з 75% вірогідністю загони військ епохи Інформаційна ера або укріплення. Види військ даної епохи: Екзоскелетна піхота та Людиноподібний робот.

Екзоскелетна піхота – загін війська ближнього бою епохи Майбутнє (рис. Б.31).



Рисунок Б.31 – Мотопіхота епохи «Майбутнє»

Параметри:

- 1) Шкода: 12
- 2) Життя: 26

3) Покращується: ні.

4) Ціна: 2 бали дій

Людиноподібний робот – загін війська *без спеціалізації* епохи Майбутнє (рис. Б.32).



Рисунок Б.32 – Людиноподібний робот епохи «Майбутнє»

Параметри:

1) Шкода: 9

2) Життя: 32

3) Покращується: ні.

4) Ціна: 4 бали дій.

Укріплення та їх різновиди у кожній епосі

Кожній епосі відповідає своє укріплення причому, якщо гравець купував укріплення ще у стародавньому світі, а скористався ними у Майбутньому, то характеристики укріплень будуть відповідати укріпленню Майбутнього.

Таблиця Б.2 – Параметри укріплення кожної з епох

Назва епохи	Кількість шкоди, що укріплення буде наносити нападнику	Кількість життя укріплення
Стародавній світ	0	5
Античність	1	6
Середньовіччя	1	7
Відродження	2	9
Промислова ера	2	11
Новий час	3	14
Епоха атому	3	17
Інформація ера	3	21
Майбутнє	4	26

ДОДАТОК В.

РЕАЛІЗАЦІЯ СТВОРЕННЯ ПОКАЖЧИКА ВІЙСЬКОВОГО ЗАГОНУ

Детальний опис конструктору створення ігрового загону на прикладі Воїну епохи Стародавнього світу.

```
Squad::Squad(enumSquad armyID) //Конструктор класу, що приймає лише енам
{
    this->armyTexture.loadFromFile("images/army.png"); //Додавання текстури як атрибуту
    this->armySprite.setTexture/armyTexture); //Додавання спрайту на основі текстури як
атрибуту
    switch (armyID) //порівняння з усіма можливими типами загонів
    {
        case enumSquad::SQUAD_WARRIOR: //якщо Воїн то..
            this->armySpriteCoordinateX = 0;
            this->armySpriteCoordinateY = 0; //беремо спрайт з (0;0) координат текстури
            this->armyAttackBasic = 1; //даємо базову 1 шкоди
            this->armyHealthBasic = 3; //та базові 3 життя
            this->armyName = "Warrior"; //називаємо Воїном
            this->eraName = enumEraName::ERA_ANCIENT; //Додаємо атрибут епохи, якій він
належить
            this->armyUpgrade = enumSquad::SQUAD_SWORDSMAN; //Додаємо атрибут у тип
війська, що буде покращений
            this->armyType = enumTypeSquad::TYPE_MEELE; //Додаємо атрибут типу війська з вже
визначеного енаму
            break;
            ... // тут був процес створення усіх інших загонів с точним налаштуванням параметрів
default:
```



```
        break;
    }
    this->armySprite.setTextureRect(sf::IntRect(static_cast<int>(this->armySpriteCoordinateX),
static_cast<int>(this->armySpriteCoordinateY), static_cast<int>(this->armySize), static_cast<int>(this-
>armySize))); //Вирізаємо спрайт
    this->armyAttackNow = this->armyAttackBasic;
    this->armyHealthNow = this->armyHealthBasic; //Додаємо як атрибути теперішні показники
шкоди та життя
}
```

На рисунку Б.1 зображений результат даного конструктору.



Рисунок Б.1 – Результат створення Воїну епохи Стародавнього світу