

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій  
Кафедра комп'ютерних наук та технологій

*Дипломна магістерська робота*

на тему

Багатокористувацька економічна покрокова онлайн стратегія

Виконав: студент групи МгІТ-21  
спеціальності  
122 Ком'ютерні науки

Владислав ГОРІЛИЙ

Керівник доц. Оксана КОЛИСКО

Рецензент \_\_\_\_\_

Київ 2022

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

ФАКУЛЬТЕТ МЕХАТРОНИКИ ТА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Спеціальність 122 Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри КНТ**

\_\_\_\_\_ проф. Володимир ЩЕРБАНЬ

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 року

## **ЗАВДАННЯ**

**НА ДИПЛОМНУ МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

*Горілому Владиславу Олександровичу*

**1. Тема роботи** Багатокористувацька економічна покрокова онлайн стратегія

Науковий керівник роботи Колиско Оксана Зенонівна,

затверджені наказом вищого навчального закладу від “ \_\_\_\_\_ ” \_\_\_\_\_ 2022 року

№ \_\_\_\_\_

**2. Строк подання студентом роботи** 10.11.2021 р.

**3. Вихідні дані до роботи** Розробка кафедри комп'ютерних наук

**4. Зміст дипломної роботи:** 1) Стратегії та їх типи 2) Аналіз популярних стратегій 3) Створення прототипу стратегії

**5. Перелік графічного матеріалу:** презентація на \_\_\_\_\_ слайдах.

## 6. Консультанти розділів дипломної магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 1	Колиско О.З., доц.каф.КН		
Розділ 2	Колиско О.З., доц.каф.КН		
Розділ 3	Колиско О.З., доц.каф.КН		

## 7. Дата видачі завдання жовтень 2022 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної магістерської роботи	Терміни виконання етапів	Примітка про виконання	
			Студ	Керівн
1	Вступ	09.10.2022		
2	Розділ 1. Стратегії та їх типи	12.10.2022		
3	Розділ 2. Аналіз популярних стратегій	20.10.2022		
4	Розділ 3. Створення прототипу стратегії	25.10.2022		
5	Висновки	25.10.2022		
6	Оформлення дипломної магістерської роботи (чистовий варіант)	30.10.2022		
7	Здача дипломної магістерської роботи на кафедрі для рецензування (за 14 днів до захисту)	30.10.2022		
8	Перевірка дипломної магістерської роботи на наявність ознак плагіату (за 10 днів до захисту)	07.11.2022		
9	Подання дипломної магістерської роботи у відділ магістратури для перевірки виконання додатку до індивідуального навчального плану (за 10 днів до захисту)	07.11.2022		
10	Подання дипломної магістерської роботи на затвердження завідувача кафедри (з 7 днів до захисту)	07.11.2022		

Студент \_\_\_\_\_ Владислав ГОРІЛИЙ

Науковий керівник роботи \_\_\_\_\_ Оксана КОЛИСКО

Директор НМЦУПФ \_\_\_\_\_ Олена ГРИГОРЕВСЬКА

## АНОТАЦІЯ

**Горілий Владислав Олександрович. Тема Багатокористувацька економічна покрокова онлайн стратегія.**

Дипломна магістерська робота за спеціальністю 122 – Комп'ютерні науки. – Київський національний університет технологій та дизайну, Київ, 2022 рік.

Метою мого дослідження стала розробка нової програми-гри в жанрі стратегія, а завданням дослідження стало на основі теоретичної інформації та проведеного аналізу, розробити концепцію багатокористувацької покрокової стратегії.

Об'єктом дослідження виступила ігрова індустрія, а предметом – компанії, що займаються розробкою стратегій різних типів.

Діяльність були задіяні такі методи дослідження: аналіз, синтез, моделювання, аналогія, узагальнення, конкретизація, порівняння, опитування. Також були використані такі способи розробки: мова програмування C#, JavaScript, Python, PHP, стандартизована мова гіпертекстової розмітки HTML, ігровий движок Unity, мова запитів SQL.

Практична значимість полягає у розробці нової програми-стратегії, яка може бути запатентована в майбутньому та приносити дохід.

Наукова новизна полягає у поєднанні жанрів покрокової стратегії та стратегії у реальному часі у новий жанр.

У першому розділі був представлений опис стратегій, їх історія та типи.

У другому розділі було проведено аналіз основних ігор-стратегій, їх переваг, характерних рис.

У третьому розділі був описаний процес розробки розрахованої на багато кроків стратегії.

*ключові слова: покрокова стратегія, розробка гри, розрахована на багато користувачів додаток, стратегія в реальному часі*

## ANOTACIA

**Horilyi Vladyslav Oleksandrovich. Topic Multiplayer economic turn-based online strategy.**

The purpose of my research was to develop a new application game in the strategy genre, and the task of the research was to develop a step-by-step concept based on theoretical information and analysis strategies.

The object of the study was the gaming industry, and the subject was companies engaged in the development of strategies of various types.

The following research methods were used in the work: analysis, synthesis, modeling, analogy, generalization, concretization, comparison. The following development methods were also applied: programming languages such as C#, JavaScript, Python, PHP, standardized HTML hypertext markup language, Unity game engine, language of requests SQL.

The practical significance lies in the development of a new application-a strategy that can be patented in the future and generate revenue.

The scientific novelty consists in combining the genres of turn-based strategy and real-time strategy into a new genre.

The first section presented a description of the strategies, their history and types.

In the second section an analysis was made of the main strategy games, their advantages, and characteristic features.

The third section described the process of developing a multiplayer turn-based strategy.

*keywords: turn-based strategy, game development, multiplayer application, real-time strategy*

## ЗМІСТ

Вступ.....	7
Розділ 1. Стратегії та їх типи.....	9
1.1 Стратегія як жанр, перспективи.....	9
1.2 Історія становлення жанру стратегія.....	16
1.3 Типологія стратегій.....	19
1.4 Багатокористувацька покрокова стратегія в реальному часі.....	30
Висновки до розділу 1.....	33
Розділ 2. Аналіз популярних стратегій.....	35
2.1 Покрокові стратегії.....	35
2.1.1 Civilization V.....	35
2.1.2 Heroes of Might and Magic.....	39
2.2 Стратегії у реальному часі.....	44
2.2.1 StarCraft II.....	44
2.2.2 Age of Empires II.....	50
2.3 Покрокові стратегії реального часу.....	54
2.3.1 Total War: Shogun 2.....	54
2.3.2 Frozen Synapse 2.....	60
Висновки до розділу 2.....	64
Розділ 3. Створення прототипу стратегії.....	66
1.1 Сайт.....	66
1.2 База даних.....	81
1.3 Розробка гри.....	85
Висновки до розділу 3.....	103
Висновки.....	104
Список використаних джерел.....	107
Додатки.....	111

## ВСТУП

Дозвілля є важливою частиною людського життя як у дітей, так і у дорослих. Діти через гру пізнають світ, навчаються новому, а дорослі дають мозку та тілу відпочити від справ.

Дійсно важливо організувати своє дозвілля правильно, адже якщо є кілька годин на відпочинок, то можна витратити їх як на безглузде гортання стрічки новин, так і на корисну діяльність. Таким чином, питання з корисним проведенням часу досить актуальне в наш час.

Можна почитати книги, благо, їх досить багато існує на даний момент, можна вирішити sudoku, шахові завдання або провести шахову партію зі знайомою людиною. Також часто рятує звичайна прогулянка на свіжому повітрі.

Однак, це все може раптово стати недоступним (як прогулянки) через тривогу або погану погоду, так і може більше навіяти нудьгу своєю одноманітністю.

Як відомо, у шахах багато що можна передбачити лише за рахунок обмеженості вибору. Видатні шахісти можуть напам'ять відтворювати шахові партії із заплющеними очима.

Для такого випадку було вирішено розробити щось нове, більш непередбачуване. Був вибраний формат кросплатформової гри на персональні комп'ютери та телефони, адже в такому випадку розширюється цільова аудиторія, яка теоретично зацікавиться даним додатком.

Виходячи з мети, необхідно виконати такі завдання:

- виконати аналіз обраної предметної галузі, саме ігровий промисловості у жанрі стратегій різних підтипів
- проаналізувати найпопулярніші стратегії щодо їхнього інтересу для гравців
- вибрати тип жанру для гри
- створити робочий прототип гри, використовуючи ігровий движок Unity

- створити сайт та базу даних для реєстрації нових гравців та зберігання даних про вже існуючі облікові записи
- запуснути програму на сервері для мультиплеєра

Говорячи про підсумки виконаної роботи, можна сказати, що все пройшло успішно. У ході розробки гри був придуманий і реалізований на практиці прототип гри в жанрі «розрахована на багато користувачів покрокова онлайн стратегія».

Однак, на відміну від звичайних покрокових стратегій, цей жанр включає як покрокову дію гри (існує таймер, на результаті якого показуються всі дії, вчинені гравцями), так і участь у реальному часі всіх учасників поєдинку (для прикладу, в шахах кожен хід належить лише одній людині – що грає за білу сторону або за чорну; тут же обидва гравці ходять одночасно).

За допомогою мови запити SQL була розроблена база даних, що зберігає вигадані імена (логіни або нікнейми) та паролі від облікових записів гравців. За допомогою HTML, PHP та JavaScript був створений сайт для реєстрації нових користувачів. Мова Python була задіяна при створенні лаунчера гри.

Сама гра була розроблена мовою C# на ігровому движку Unity. Фінальний прототип має невелику вагу та мінімалістичну графіку, що чудово підходить для кросплатформної гри на ПК та мобільних пристроях (Android).

Програма має потенціал розвитку в майбутньому. За рахунок вибудовування успішної рекламної кампанії та вмілого маркетингу можна досягти досить великих результатів.



## РОЗДІЛ 1 СТРАТЕГІЇ ТА ЇХ ТИПИ

### 1.1 Стратегія як жанр, перспективи.

Стратегія — загальний, недеталізований план, що охоплює тривалий період часу, спосіб досягнення складної мети, у військовій справі, пізніше взагалі будь-якої діяльності людини.

Завданням стратегії є ефективне використання готівкових ресурсів для досягнення основної мети (стратегія як спосіб дій стає особливо необхідною у ситуації, коли для прямого досягнення основної мети недостатньо готівкових ресурсів).

Стратегічна гра — жанр комп'ютерних ігор, у яких гравцю перемоги необхідно застосовувати стратегічне мислення. У багатьох іграх такого жанру гравцеві пропонується грати не за конкретного персонажа, а за їх умовні маси, наприклад, керувати будівництвом міста або командувати цілими арміями у військових кампаніях.

Простіше кажучи, стратегія – це мистецтво ведення війни. У часи великих битв за участю тисяч солдатів розташування військ, послідовність атак і грамотне використання наявних одиниць могли принести перемогу навіть за чисельного програшу. У результаті такі битви стали улюбленою темою багатьох розробників.

Стратегічне мислення - це здатність аналізувати взаємодію з іншими людьми, тоді як вони, у свою чергу, роблять те саме.

Також для проектування будь-якого типу ігор необхідне знання такої дисципліни як теорії ігор – математичний метод вивчення оптимальних стратегій в іграх.

Під грою розуміється процес, у якому беруть участь дві і більше боку, які ведуть боротьбу реалізацію своїх інтересів. Кожна зі сторін має свою мету та використовує деяку стратегію, яка може вести до виграшу чи програшу – залежно від поведінки інших гравців.

Теорія ігор допомагає вибрати кращі стратегії з урахуванням уявлень про інших учасників, їх ресурси та їх можливі вчинки.

Теорія ігор – це розділ прикладної математики, точніше дослідження операцій. Найчастіше методи теорії ігор знаходять застосування в міжнародних відносинах, економіці, трохи рідше в інших суспільних науках - соціології, політології, психології, етики, юриспруденції та інших.

Хоча теорія ігор спочатку і розглядала економічні моделі, до 1950-х вона залишалася формальною теорією у межах математики. Починаючи з 1970-х років, її взяли на озброєння біологи для вивчення поведінки тварин і теорії еволюції. Дуже важливе значення вона має для штучного інтелекту та кібернетики, особливо з виявом інтересу до інтелектуальних агентів.

Історично першими у сферу інтересів математиків потрапили ігри з повною інформацією, у яких щодо просто аналізувати стратегію всіх учасників. Потім увагу дослідників привернули «ігри із неповною інформацією». Проаналізувавши покер та інші ігри цього класу, математики спробували застосувати математичний апарат до ігор «глобального масштабу» — війн, економіки та навіть до звичайних розлучень.

Ігри теоретично ігор є суворо певні математичні об'єкти. Гра утворюється гравцями, набором стратегій кожного гравця і вказівки вигравшів, чи платежів, гравців кожної комбінації стратегій. Більшість кооперативних ігор описуються характеристичною функцією, тоді як інших видів частіше використовують нормальну чи екстенсивну форму.

Характеризуючі ознаки гри як математичної моделі ситуації:

- наявність кількох учасників;
- невизначеність поведінки учасників, пов'язана з наявністю кожного з них кількох варіантів дій;
- відмінність (несупадання) інтересів учасників;
- взаємопов'язаність поведінки учасників, оскільки результат, який отримує кожен із них, залежить від поведінки всіх учасників;
- наявність правил поведінки, відомих усім учасникам.

Теоретично ігор стратегія гравця - це кожен із варіантів, що він вибирає за умов, коли результат залежить тільки від його власних дій, а й від дій інших. Дисципліна в основному стосується дій гравця у грі, що впливають на поведінку чи дії інших гравців. Деякі приклади "ігор" включають шахи, бридж, покер, монополію, дипломатію чи морський бій. Стратегія гравця визначає дії, які гравець почне на будь-якому етапі гри. При вивченні теорії ігор економісти використовують раціональніший підхід під час аналізу рішень, а чи не психологічні чи соціологічні підходи, прийняті під час аналізу взаємозв'язків між рішеннями двох чи більше сторін у різних дисциплінах.

Концепцію стратегії іноді (помилково) плутають із концепцією ходу. Хід - це дія, яку робить гравець у певний момент гри (наприклад, у шахах, переміщення білого слона a2 на b3). З іншого боку, стратегія - це повний алгоритм ведення гри, що вказує гравцю, що робити в кожній можливій ситуації протягом гри. Корисно думати про "стратегію" як про список напрямків, а про "рух" як про один поворот у самому списку напрямків. Ця стратегія полягає в виграші чи результаті кожної дії. Ціль кожного агента полягає в тому, щоб розрахувати свій виграш на основі дій конкурентів. Наприклад, конкурент А може припустити, що конкурент виходить ринку. Звідти учасник А порівнює виплати, які він отримує, входячи та не входячи. Наступний крок - припустити, що конкурент В не бере участі, а потім розглянути, який виграш краще, виходячи з того, чи вирішить конкурент А брати участь чи ні. Цей метод дозволяє виявити домінуючі стратегії, за яких гравець може визначити дію, яку він може зробити незалежно від того, що робить суперник, щоб спробувати максимізувати виграш. Це також допомагає гравцям визначити рівновагу Неша, яка більш детально обговорюється нижче.

Теорія ігор як із підходів у прикладної математиці застосовується вивчення поведінки людини і тварин у різних ситуаціях. Спочатку теорія ігор почала розвиватися в рамках економічної науки, дозволивши зрозуміти та пояснити поведінку економічних агентів у різних ситуаціях. Пізніше сферу застосування теорії ігор було розширено інші соціальні науки; в даний час теорія ігор

використовується для пояснення поведінки людей у політології, соціології та психології.

Таким чином, знання теорії ігор може допомогти у вибудовуванні збалансованої стратегії та створенні прототипу покрокової стратегічної гри.

Коли кожен учасник відіграє важливу роль у взаємодії (або тому, що з самого початку був великим гравцем, або тому, що зобов'язання або особиста інформація зводять відносини між гравцями до рівня, при якому кожен гравець стає ключовим у рамках цих відносин), така взаємодія слідує розглядати як стратегічну гру. Подібні ситуації — скоріше правило, ніж виняток, у бізнесі, політиці та навіть у соціальних відносинах. Отже, вивчення стратегічних ігор є важливим елементом усіх областей, які аналізують подібні питання.

Серед властивостей та визначень стратегії можна виділити такі основні ознаки:

- 1) план, орієнтир, напрямок розвитку, керівництво;
- 2) принцип поведінки чи дотримання деякої моделі поведінки;
- 3) позиція (зокрема, йдеться про позиціонування товару на ринку);
- 4) перспектива, погляд на майбутнє підприємства;
- 5) прийом, особливий маневр, що дозволяє обійти конкурентів.

Більшість стратегій має задовольняти наступним у достатній мері широким критеріям:

1. Послідовність. Стратегія не повинна містити взаємно несумісних цілей та політичних прийомів.

2. Гармонійність. Стратегія покликана забезпечувати адаптивну реакцію на зовнішнє середовище та зміни, що відбуваються в ній.

3. Переваги. Стратегія надає можливість придбання та (або) підтримки конкурентних переваг у обраній сфері діяльності.

4. Здійсненність. Стратегія не повинна пред'являти надмірні вимоги до ресурсів або породжувати нерозв'язні проблеми.

Стратегії, які не відповідають наведеним вище критеріям, є дуже підозрілими. Це означає, що така стратегія не здатна забезпечити виконання жодної з ключових функцій, необхідних для виживання фірми.

Стратегія в загальному вигляді відображає характер підприємства та його відмінні риси, вказує загальний курс розвитку підприємства в динамічному зовнішньому середовищі. Незважаючи на те, що поняття «стратегія» асоціюється зі стабільністю, в сучасних умовах акцент слід робити на змінах.

Брюс Хендерсон писав 1981 року, що: "Стратегія залежить від здатності передбачати майбутні наслідки поточних ініціатив". Він писав, що основні вимоги до розробки стратегії включають, серед інших факторів:

- 1) великі знання про навколишнє середовище, ринок та конкурентів
- 2) здатність розглядати ці знання як інтерактивну динамічну систему
- 3) уяву та логіку для вибору між конкретними альтернативами

Хендерсон писав, що стратегія цінна через: "обмеженості ресурсів, невизначеності щодо можливостей та намірів противника; незворотного витрачання ресурсів; необхідності координації дій у часі та на відстані; невизначеності щодо контролю ініціативи; та характеру взаємного сприйняття супротивників один одного". Таким чином, можна сказати, що ведення війни передбачає два абсолютно різні види діяльності:

- організація окремих бойових дій та їх проведення
- дотримання спільної мети війни

На думку Карла Клаузевіця, перший вид діяльності називається тактикою, а другий — стратегією. Тактика потрібна у тому, щоб виграти битву. Стратегія потрібна для того, щоб виграти війну.

У мирний час синонімом стратегій стали настільні ігри, знайомі людству з найдавніших часів — шашки, шахи, нарди, доміно та багато іншого. Усі вони вимагали від гравців чіткого плану дій, який потрібно ще й підлаштовувати під тактику противника. Розвитком цього напряму стала поява сучасних настільних ігор у найрізноманітніших варіаціях: із фігурками, кістками, картками тощо.

Найвідоміші представники – знайомі багатьом Монополія, Ерудит, Мафія та Каркассон.

З появою комп'ютерів та консолей розробники не втрачали можливості перенести мистецтво ведення війни на екрани у тому чи іншому вигляді. Так почав зароджуватися жанр комп'ютерних стратегій, що включає безліч піджанків, таких як стратегія в реальному часі, покрокова стратегія, МОВА, 4X та багато інших, про які буде розказано пізніше.

Більше того, зараз жанр стратегія добре почувається і виходячи на ринок мобільних ігор. За даними Statista, найпопулярніші жанри мобільних ігор у США в 2021 році за часткою доходів від iOS:

- 1) Puzzle (21.18%)
- 2) Casino (18.9%)
- 3) Strategy (17.17%)

Стратегія знаходиться на 3 місці, не сильно відстаючи від конкурентних жанрів. Отже, на ринку мобільних ігор стратегія має перспективи розвитку.

Також, враховуючи дослідження Newzoo, гравці різного віку найчастіше грають з мобільних пристроїв, потім йде ПК, а на останньому місці - консолі.

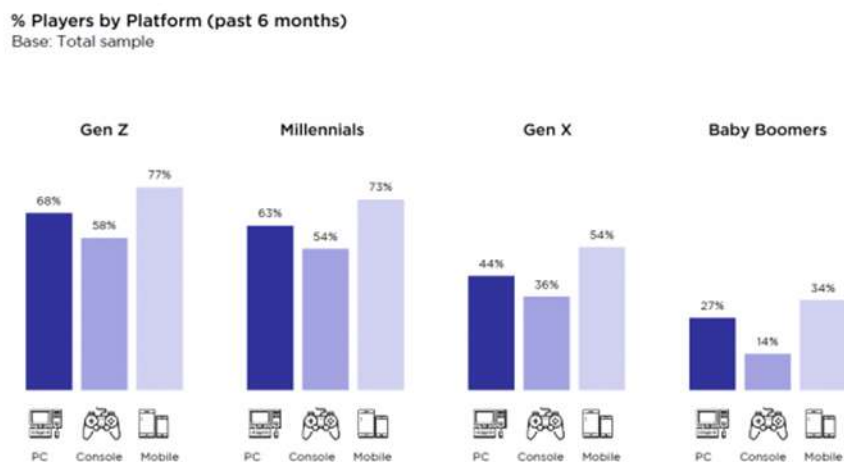


Рис. 1. Розподіл гравців по платформах.

Виходячи з цих даних, можна сказати, що жанр стратегій має особливо високий потенціал розвитку на мобільних пристроях. Для того, щоб розширити цільову аудиторію програми, можна додати кроссплатформеність.

Таким чином, гравці з різних пристроїв зможуть продовжувати грати в гру з гравцями на різних пристроях або, за бажання, тільки з гравцями на таких же пристроях, як у них, адже ймовірно, в деяких випадках кросплатформність може здатися нечесною.

Гравці з мобільних пристроїв можуть обуритися наявністю гарячих клавш і більшого екрану у гравців з комп'ютера, а ті, у свою чергу, можуть обуритися більшим інтерфейсом та більшою мобільністю по карті.

Однак у кросплатформності більше плюсів ніж мінусів. З таких можна виділити:

- Однак у кросплатформності більше плюсів ніж мінусів. З таких можна виділити

- Економічно вигідно замовляти розробку однієї програми, а не кількох
- Простий та швидкий запуск в AppStore и Google.Play
- Охоплення всієї можливої аудиторії
- Одинаковий інтерфейс для обох ос

Мала перспектива розвитку лише на консолях. Реалізація подібної стратегії може стати скрутною, а фінальний прототип інтуїтивно незрозумілим та заплутаним. Для такого випадку було вирішено створити програму, в яку можна грати тільки з ПК та мобільних пристроїв.

Далі ми детальніше розглянемо історію розвитку жанру стратегії.

## **1.2 Історія становлення жанру стратегія.**

Витоки стратегічних відеоігор сягають корінням у традиційні настільні стратегічні ігри, такі як шахи і го, а також багато інших настільних ігор. Перші ігри взагалі були сумішшю консольної та настільної гри, наприклад, Invasion, мова про яку піде далі.

Перші «комп'ютерні» шахи з'явилися приблизно 1769 року. "Механічний турок" був зроблений спеціально для королеви Марії-Терезії. У ньому сидів професійний гравець, який відповідав за всі ходи «Турка».

Першою консольною стратегічною грою була ризикована гра Invasion, випущена в 1972 для Magnavox Odyssey. На думку деяких, це була більшою мірою настільна гра. До неї додалося досить велике паперове ігрове поле яким переміщалися гравці. Приставка в цій грі потрібна була для візуалізації атак замку. Накладка на екран була зроблена у вигляді замка в межах якого переміщалися квадрати гравців, що ілюстрували бій. Загалом це була повноцінна велика настільна гра.

Комп'ютерна гра Strategic Simulations (SSI) Bismarck, випущена у 1980 році, була першою історичною комп'ютерною військовою грою. Computer Bismarck це покрокова стратегічна комп'ютерна гра у жанрі варгейм. Була розроблена та видана компанією Strategic Simulations, Inc. 1980 року. Сюжет гри заснований на подіях 1941 Другої світової війни, в ході яких Збройні сили Великобританії переслідували німецький лінкор Бісмарк.

Гра була комерційно успішною та отримала хороші відгуки преси. Computer Bismarck відіграла важливу роль у популяризації комп'ютерних ігор загалом та стратегічних ігор зокрема.

Reach for the Stars 1983 року була однією з перших стратегічних ігор 4X, яка розширила взаємозв'язок між економічним зростанням, технологічним прогресом та завоюваннями. Reach for the Stars — покрокова стратегічна комп'ютерна гра, розроблена та випущена австралійською компанією Strategic Studies Group у 1983 році для платформ Apple II та Commodore 64. Дія гри відбувається у науково-фантастичному всесвіті. Метою гравця є будівництво власної галактичної імперії за допомогою колонізації, завоювань та вивчення технологій.

На окрему згадку заслуговує The Lords of Midnight. Ця гра об'єднала елементи пригод, стратегії та військових ігор та отримала премію журналу Crash Magazine за найкращу пригодницьку гру 1984 року, а також Найкращу стратегічну гру року на Golden Joystick Awards. The Lords of Midnight є сумішню покрокової стратегії та квесту у відкритому світі, з виглядом від першої особи. Гравець починає з чотирма персонажами (Luxor the Moonprince, Rorthron the Wise,



Corleth the Fey та Morkin) і може набрати до 28 майбутніх Лордів (Ithron, Utarg з Utarg, Blood, Hajorkith, Shadows тощо), які допомагатимуть (або навпаки) заважати) йому у справі знищення опівнічного короля-чарівника.

Herzog Zwei 1989 часто вважається першою стратегічною грою в реальному часі, хоча елементи стратегії в реальному часі можна знайти в декількох ранніх іграх, таких як Cytron Masters Дена Бантена і Утопія Дона Деглоу в 1982 році; Війни Бокосука Кодзі Сумії у 1983 році; "Стонкери" Д. Х. Лоусона та Джона Гібсона та "Епідемія" Стівена Фабера в 1983 році; і Evguware "Давнє мистецтво війни" в 1984 році. Гра містить набір рівнів-карт, дія на яких відбувається в різних умовах, що впливають на ігровий процес - у лісах, островах, вулкані. На карті борються два гравці (гравець та комп'ютер або гравець та другий гравець). Кожен гравець має головну базу, знищення якої призводить до поразки. Також на карті знаходяться малі бази, які можуть захоплювати гравці. Від кількості захоплених малих баз залежить швидкість додавання коштів на рахунок гравця.

Цей жанр був популяризований Dune II через три роки, в 1992 році. Бретт Сперрі, творець Dune II, вигадав назву "стратегія в реальному часі", щоб допомогти просувати на ринок новий ігровий жанр, який він допоміг популяризувати.

Стратегічні ігри в реальному часі змінили жанр стратегії, наголосивши на важливості тайм-менеджменту, коли на планування йде менше часу. Стратегічні ігри в реальному часі згодом стали перевершувати з продажу покрокові стратегічні ігри.

До лютого 2009 року в світі було продано більше 11 мільйонів копій StarCraft (1998), яка стала однією з найпопулярніших ігор для персональних комп'ютерів. Її хвалили за новаторське використання унікальних "фракцій" в ігровому процесі RTS та за захоплюючий сюжет. Однією з найважливіших нововведень гри є різницю між воюючими сторонами за дотримання балансу сил. StarCraft — одна з перших стратегій у реальному часі, в якій кожна сторона

відрізняється не лише зовнішнім виглядом, а й характеристиками своїх представників, а також загальною стратегією бою.

Warcraft III: Reign of Chaos, що вийшла в 2002 році, вплинула на стратегічні ігри в реальному часі, особливо на додавання рольових елементів і героїв як юніти. Більше, ніж сама гра, моди, створені за допомогою редактора світу, призвели до тривалих змін і надихнули багато майбутніх стратегічних ігор.

Defense of the Ancients (DotA), створене співтовариством доповнення (мод), засноване на Warcraft III, значною мірою вважається найбільш значним джерелом натхнення для розрахованої на багато користувачів онлайн-бойової арени (МОБА). Оскільки формат був прив'язаний до Warcraft'у, розробники почали працювати над власними іграми в стилі "DOTA", включаючи Heroes of Newerth (2009), League of Legends (2010) та окреме продовження мода, Dota 2 (2013). Blizzard Entertainment, власник власності Warcraft, розробила гру, натхненну DotA, під назвою Heroes of the Storm (2015), в якій представлено безліч героїв з франшизи Blizzard, у тому числі численні герої з Warcraft III.

Heroes of Newerth (скорочено HoN) — комп'ютерна розрахована на багато користувачів командна гра в жанрі МОБА, розроблялася Frostburn Studios. Процес гри схожий на DotA (частина героїв і предметів спочатку являли собою точну копію). На даний момент, після множини оновлень, з'явилися десятки абсолютно нових героїв і предметів, придуманих розробниками HoN, а багато старих сильно змінилися порівняно зі своїми DotA-аналогами.

Кожен гравець у HoN (максимум 10 гравців 5 на 5) управляє одним персонажем і основна увага приділяється управлінню та розвитку цього персонажа, а ігровий процес заснований на командній взаємодії.

Схожий процес гри мають і популярні на сьогоднішній день ігри як Dota 2, League of Legends, Heroes of the Storm. Відмінності у всіх цих іграх виражаються лише у різних персонажах, механіках відкриття персонажів, графікою, набором карт, а також механікою розвитку.

У Dota і League of Legends механіка розвитку гри залежить не тільки від здібностей, що прокачуються, але і від купівлі предметів, що впливають на ці здібності і основні показники гравця.

Однак, однією з найуспішніших серій стратегічних ігор усіх часів стала серія Total War, що починається від 2000 року, створена і розробляється британською компанією The Creative Assembly. Перша гра серії розійшлася тиражом понад 20 мільйонів копій.

В даний час серія включає десять самостійних ігор та сім додатків до них. Традиційно в іграх серії поєднуються жанри покрокової глобальної стратегії та тактичних битв у реальному часі. Ігри серії відрізняються масштабністю та опрацьованістю битв.

### **1.3 Типологія стратегій.**

Як можна зрозуміти з попереднього розділу, сам жанр стратегії поділяється на піджанри чи типи. Таких типологій можна навести безліч. Можна аналізувати графіку гри, тип ігрового світу, час у який це сталося чи міг би статися, проте тут були зібрані основні класифікації саме стратегічного жанру.

За кількістю гравців:

- 1) Одиночна гра. Бої гравця відбуваються проти П.
- 2) Спільна гра на одному пристрої. Для прикладу можна навести шахи на двох або додаток морський бій, коли люди грають на одному пристрої по черзі.
- 3) Спільна гра на одному пристрої. Для прикладу можна навести шахи на двох або додаток морський бій, коли люди грають на одному пристрої по черзі.

За наявністю кооперативної гри:

1) Кооперативна. Це означає, що є можливість об'єднатися з іншими гравцями для досягнення спільної мети – перемоги над сильнішим П або грати в режимах 2x2, 3x3 тощо. Також кооперативність може означати можливість вступати до тимчасових альянсів з іншими гравцями.

2) Не кооперативна. Поєднання з іншими гравцями неможливе.

З механіки вчинення дій:

1) Покрокова стратегія (Turn-Based Strategy, TBS). Це піджанр стратегічних ігор, в яких ігровий процес складається з послідовності фіксованих моментів часу, які називаються ходами (або кроками), під час яких гравці здійснюють свої дії.

Покрокові стратегії з'явилися раніше RTS (Real-Time Strategy) і відрізняються значною різноманітністю. Поділ ігрового процесу на ходи відриває його від реального життя і позбавляє гру динамізму, внаслідок чого ці ігри не такі популярні на сьогоднішній день, як стратегії в реальному часі. З іншого боку, у TBS у гравця набагато більше часу на роздуми, під час ходу його ніщо не підганяє, що дозволяє приділяти більше часу плануванню.

Таким чином, основною характеристикою покрокових стратегічних ігор є дискретність ігрового процесу. У більшості покрокових стратегій гравці здійснюють ходи по черзі, як у таких класичних настільних іграх, як, наприклад, шахи.

Також існує тип стратегічних ігор, які іноді відносять до покрокових, але гравці в них роблять ходи одночасно. Такі ігри стосуються непопулярного підтипу покрокової стратегії в реальному часі, про який буде написано в окремому пункті.

2) Стратегія у реальному часі (Real-Time Strategy, RTS). За часів виходу Dune II (одна з перших ігор у жанрі RTS) сформувалися загальні принципи класичних стратегій у реальному часі:

- Економіка в грі має допоміжний характер і націлена на збір ресурсів;
- В основі економіки лежать споруди, які можна будувати та руйнувати. Саме через них будують юніти (див. нижче) та проводять дослідження. Іноді споруди виробляють особливі юніти, іноді (як у Dune) інші конструкції. Деякі конструкції можуть атакувати супротивника відразу після закінчення створення, деякі після спеціальних удосконалень;
- Збір ресурсів здійснюється особливими юнітами в спеціально передбачених для цього місцях, після чого вони переносяться до спеціальної будівлі на базі і тільки після цього надходять у розпорядження гравця;

- Ресурси можуть бути різних видів (наприклад, золото, деревина, гроші, метал, вугілля) та витрачаються на будівництво юнітів та будівель (також на це йде час);

- Юніт (unit) – будь-яка бойова одиниця (піхотинець, танк, літак, корабель), яка зазвичай може атакувати інші юніти та будівлі та знищувати їх. Юніти мають параметри, головними серед яких є «окуляри здоров'я», пошкодження, що наносяться, броня (зменшує шкоду, що наноситься юніту), швидкість. Юніт може включати і кілька солдатів (танків, артилерійських знарядь і т. д.), але в цьому випадку командувати ними окремо не можна. Деякі юніти не можуть атакувати та призначені для збору ресурсів, будівництва будівель, транспортування інших юнітів;

- Усі будівлі та юніти мають радіус огляду, далі якого «бачити» вони не можуть;

- Гравець може бачити, що відбувається тільки на тих територіях, що потрапляють у радіус огляду його будівель та юнітів. Ті території, де він ще був, зафарбовані чорним. Ті, на яких його війська вже були, але в даний момент не можуть їх бачити, покриті так званим «туманом війни» (Fog of War), який зазвичай дозволяє бачити ландшафт цієї місцевості та будівлі (такими, якими вони були в останній). момент, коли їх бачили юніти гравця), але не дозволяє бачити пересування супротивника та пов'язані з цим можливі зміни ландшафту.

Таким чином, завдання, які гравець повинен виконати для успіху в RTS, можуть бути дуже вимогливими, тому, для того щоб допомогти впоратися із завданням, розвивалися складні інтерфейси користувача. Деякі функції були запозичені серед робочого столу; наприклад, метод «клацання та перетягування» для вибору всіх одиниць у цій галузі. Хоча деякі інші жанри мають концептуальні та геймплейні подібності із шаблоном RTS, визнані жанри, як правило, не вважаються RTS іграми.

За наповнення гри різними механіками:

1) Симулятори будівництва та управління - жанр комп'ютерних ігор, що стоїть на стику симуляторів та стратегій. Вони можуть бути присвячені будівництву міста, парку розваг та інших об'єктів, при цьому пропонуючи економічні випробування на протипагу військовим та тактичним. Умовно їх можна розділити на кілька піджанрів (містобудівний симулятор, Економічний симулятор, Політичний симулятор, Управління парком розваг, Управління колонією).

Економіка грає першорядне значення у симуляторах будівництва та управління, оскільки у цих іграх відбувається спорудження різних об'єктів за умов економічних обмежень. У деяких іграх користувачеві потрібно дослідити або прийти до розуміння ігрових принципів, проте більшості ігор перед гравцем ставиться економічна проблема. Іншими словами, ці ігри спрямовані на забезпечення зростання.

2) Варгейми – стратегії, у яких відсутня економічна складова. Зазвичай це ігри, в яких керуючий загonom або армією під час бою. Прикладом з популярних ігор можуть бути деякі доповнення для Sudden Strike і Бліцкрига.

«Варгейм» — скоріше характеристика конкретної гри, яка каже — вам не доведеться рахувати в ній гроші, більшу частину ігрового часу ви воюватимете всіма доступними засобами. А ось які кошти виявляться вам доступними, і що треба для цього зробити, залежить від фантазії розробників.

Прикладом такої гри є Z 1996 від англійської компанії The Bitmap Brothers. Незважаючи на повну відсутність економіки, гравці без проблем підвищують якість та кількість своїх загонів, вміло розпоряджаючись двома найважливішими ігровими ресурсами – територією та часом. Для стратегічного планування товарно-грошові відносини не обов'язкові, хоча, традиційні. Сам жанр залишається все тією ж стратегією чи тактикою.

3) Історичні стратегії. У глобальних стратегіях гравець бере участь у цілому періоді світової історії, керуючи державою; при цьому гра враховує різні чинники розвитку держав - економічний, військовий, культурний, соціальний та

політичний. Виділення глобальних стратегій у єдиний жанр притаманно для російськомовних розробників; на заході цей жанр зазвичай поділяють на два — 4X і grand strategy. Глобальні стратегії можуть бути як у реальному часі, так і покрокові.

Однак, строго кажучи, немає єдиного розуміння терміну «глобальна стратегія». На заході ігри, що розглядаються як глобальні стратегії, відносять до двох інших жанрів: 4X, до якого включаються ігри, подібні до Civilization або Master of Orion, та grand strategy, наприклад, історичні ігри, що випускаються шведською компанією Paradox Entertainment і розглядаються як підвид варгеймів. Проте існує думка, що такий поділ спірний, оскільки у перших і других ігор чимало спільного один з одним.

Глобальні стратегії можуть помітно відрізнитися одна від одної за ігровим процесом: бути покроковими, як Civilization, Master of Orion або Age of Wonders, йти в реальному часі, як Victoria: An Empire Under the Sun та Hearts of Iron, або змішувати елементи того й іншого, як Rome: Total War. На відміну від типових стратегій у реальному часі, у глобальних стратегіях результат партії це результат точного розрахунку, а не навичок швидкості; проте роль тактики сильно змінюється у рамках жанру.

Але основні характерні ознаки глобальної стратегії були виділені наступним чином:

- у глобальних стратегіях дія гравця завжди включені у певну історичну епоху;
- розвиток ігрових подій відбувається строго за історичною хронологією;
- в ігровий простір гри включають максимальну кількість історичних персонажів, подій, дій з епохи, що розігрується;
- ігровий ландшафт глобальних стратегій базується на реально існуючому або існуючому просторі, що історично склався.

Civilization вважається першою глобальною стратегією. Водночас елементи, що характеризують жанр, присутні у більш ранніх іграх, таких як Reach for the

Stars, Stellar Crusade, Imperium, деякі з яких вже згадувалися раніше. У контексті глобальних стратегій Civilization відрізнялася використанням історичної основи у сценарії гри, час та його вплив як окремий елемент ігрового процесу, можливість гравцеві створити свою власну «історію», ідентифікацію гравця з обраною цивілізацією-країною. Вплив Civilization був такий, що десятки ігор запозичували в неї ідеї або копіювали ігровий процес цілком.

У 1990-х Civilization не мав конкурентів у жанрі, і лише на початку 2000-х рр., на створеній основі сформується новий для індустрії комп'ютерних ігор жанр глобальних стратегій.

Серед космічних глобальних стратегій виділяється вплив на розвиток серія ігор Master of Orion, в іграх якої введено конструювання рас та їх співжиття, поділ у системах зірок і планет, концепція складання кораблів з модулів. Іншими значущими іграми в космічному сеттингу називають Galactic Civilizations і Distant Worlds.

Hearts of Iron у 2002 році стає першою глобальною стратегією, присвяченою Другій Світовій війні.

Sid Meier's Civilization IV, що вийшла в 2005 році, вводить поняття релігій, бойова система в Sid Meier's Civilization V змістилася у бік варгеймів.

Shogun: Total War і її сиквели змістили акцент на тактичну складову і ввели реальний час, що знизило стратегічну масштабність і поставило в основу битви. Значними іграми жанру вважають роботи від Paradox Entertainment (Europa Universalis, Hearts of Iron та інші), в яких отримала свій розвиток економіка, дипломатія, мікроменеджмент та його автоматизація, введена тривимірна графіка.

4) 4X-стратегії. Геймплей 4X стратегій включає 4 повноцінно реалізовані елементи: дослідження (нових земель), розширення, експлуатація, знищення. Назва походить від літери X у відповідних даним поняттям англійських словах "eXplore", "eXpand", "eXploit", "eXterminate". Класичним прикладом 4X стратегії є серія Civilization. У 4X стратегіях, як і в глобальних, гравець часто управляє



цілою державою. Однак це не є обов'язковим критерієм. 4X стратегії можуть бути як у реальному часі, так і покрокові.

Відмінність 4X стратегій від глобальних стратегій полягає в тому, що в глобальних стратегіях може бути відсутнім або бути слабо вираженим один із X-елементів. Наприклад, глобальна стратегія Hearts of Iron IV не містить елементу eXplore (дослідження нових земель). Це означає, що стратегія може бути одночасно 4X та глобальною, але не кожна глобальна стратегія є 4X.

4X є піджанром стратегічних комп'ютерних ігор, а представники жанру можуть належати як до покрокових стратегій, так і до стратегій реального часу. Ігровий процес зазвичай включає побудову імперії і дія може відбуватися на Землі, у світі фентезі або в космосі. Гравець при цьому контролює одну з різних рас або цивілізацій, кожна з яких має свої унікальні характеристики, які, як правило, полягають у наборі економічних і військових бонусів.

Термін 4X вперше з'явився в публікації Алана Емріха у вересні 1993 року під час анонсування Master of Orion у журналі Computer Gaming World, де грі було присвоєно рейтинг «XXXX» у контексті гри з рейтингом X rating. Згодом інші критики адаптували 4X у вигляді певних угод-правил для ігрового процесу:

Вивчення (англ. explore): навколишня територія невідома гравцям, і необхідно надсилати розвідників для вивчення карти території і визначення місцезнаходження конкурентів та їх активностей.

Розширення (англ. expand): гравці закріплюють нову територію, створюючи нові поселення, та іноді розширюючи з допомогою впливу інші, вже існуючі поселення; це дає можливість отримати в розпорядження великі ресурси з території, а в перспективі дозволить збільшити населення імперії, що в кінцевому підсумку сформує її економіку, науку та армію.

Експлуатація (англ. exploit): гравці збирають ресурси з підконтрольної їм області, формуючи інфраструктуру з видобутку; також вони покращують ефективність імперії у вигляді будівництва споруд чи розвитку науки.

Знищення (англ. exterminate): гравці атакують конкурентів та воюють з ними; з моменту, коли вся територія опиняється за кимось закріпленою, тільки шляхом знищення стає можливою подальша експансія; протистояння не обов'язково виражається у вигляді збройного конфлікту - це може бути космічна гонка, прагнення домінування, дипломатична перемога та ін.

4) МОБА-ігри. Multiplayer Online Battle Arena (МОБА) (буквально «розрахована на багато користувачів онлайнна бойова арена») — жанр комп'ютерних ігор, що поєднує в собі елементи стратегій в реальному часі і комп'ютерних рольових ігор.

Ігровий процес МОБА є матчем між двома командами, кожна з яких повинна домогтися знищення головної будівлі команди противника і одночасно захистити свою власну головну будівлю. Гравці керують особливими персонажами-«героями», які в ході матчу можуть отримувати бали досвіду, нові здібності та екіпірування, стаючи сильнішими. Дія гри відбувається на карті особливого вигляду: крім головних будівель, на ній знаходяться кілька ліній, по яких наступають керовані комп'ютером війська (крипи), і розташовані оборонні вежі, що стримують наступ противника. Хоча з точки зору геймплею та інтерфейсу МОБА схожі зі стратегіями в реальному часі, звичайні для стратегій у реальному часі елементи, як розвиток бази, видобуток ресурсів та наймання військ у МОБА відсутні.

Гравець керує одним персонажем – «героєм». Герої недостатньо сильні, щоб знищити всіх ворогів та виграти матч поодиночі; їхнє завдання — схилити підсумок битви на користь своєї команди. Перемагаючи супротивників, герой отримує бали досвіду. Накопичивши певну кількість досвіду, герой отримує підвищення рівня, що покращує його бойові характеристики та дозволяє вивчити нові вміння або покращити вже існуючі. Із втратою всіх очок здоров'я герой « гине » і на деякий час вибуває з гри, щоб потім знову з'явитися біля головної будівлі своєї команди із повним здоров'ям.

Як і говорилося раніше, цей жанр перетинається з жанром DotA, проте останній був зареєстрований та реалізований у вигляді гри Dota 2 компанією Valve. Тому для такого жанру було обрано нову назву – МОВА (фактично, щоб не допустити проблем з авторськими правами).

5) Карткові стратегії - це ігри, схожі з настільними колекційними картковими іграми. Прикладом такої гри може бути Magic The Gathering. Ігровий процес в іграх цього піджанру зосереджений навколо придбання величезної кількості карт та складання кількох колод для різного геймплею. Є певні набори правил, які можуть відрізнятися від гри до гри.

Існує багато таких стратегій від різних видавців. Кожна має свою структуру, свої правила, свої карти. Тим не менш, у них є спільні риси:

- У переважній більшості подібні ігри являють собою стратегічні ігри з елементом випадковості.
- Гравці самі складають колоди карт, яких, насправді, є величезна кількість - набагато більше, ніж може потрапити в колоду.
- Карти продаються як випадкові набори. Тому, щоб зібрати конкурентоспроможну колоду, доводиться не лише купувати ці набори в магазині, а й купувати, продавати та міняти карти «з рук у руки».

Крім загальних правил, описаних в інструкції, кожна карта має свої ефекти та особливості, описані прямо на карті. Крім того, карти прикрашені малюнками, які зазвичай відповідають сенсу карти.

У ККГ (тобто Collectible Card Game, Колекційна карткова гра) гравець сам складає свою колоду (для початківців продаються сформовані набори - «стартери»). Формування колоди дозволяє урізноманітнити гру та дає можливість гравцеві використовувати свою стратегію та тактику. Оскільки кількість різноманітних карт в ККГ може на порядки перевищувати стандартний розмір колоди і самі вони часто відрізняються унікальними властивостями, то кількість можливих колод стає практично нескінченною. Однак дійсно

конкурентоспроможних колод зазвичай небагато, особливо в рамках одного видання.

Крім стартерів, також продаються «Бустер» (booster) - набори випадково підібраних карт. Наприклад, у Magic: The Gathering бустер складається з 15 карт різного ступеня рідкості. Враховуючи, що в ККІ зазвичай різновиди карт кожної рідкості обчислюються десятками, легко зрозуміти, що рідкісні карти цінуються значно дорожче і дістати їх дуже важко.

Хоча деякі ігри використовують кидання кубиків, основний елемент випадковості, звичайно, надається самими картами, які можуть розкласти тисячі варіантів. «Нерозклад» може дозволити виграти одну гру в будь-кого, навіть у найсильнішого професіонала. Справжньою майстерністю вважається вміння скласти колоди, мінімально схильні до «нерозкладу».

Колекційними ці ігри називаються тому, що кількість карток значно більша, ніж використовується одним гравцем, і придбати відразу всі існуючі карти неможливо. Крім самого процесу гри, для гравця важливим стає одержати карти, які вважаються рідкісними. Можна сміливо сказати, що процес обміну, купівлі, продажу карт є одним із складових частин задоволення, одержуваного від ККГ гравцями, і доходу, одержуваного виробниками. Багато гравців збирають тематичні колекції; Наприклад, в Magic: The Gathering прийнято збирати карти істот одного типу - ельфів, ангелів, звірів та багатьох інших.

У багатьох ККІ прийнято видавати обмежений тираж спеціальних «фойлових» (foil) карт. Вони зазвичай цінуються в півтора-два рази дорожче за звичайні і дуже часто є предметом колекціонування.

Найвідомішою на даний момент є Hearthstone від Blizzard. Взавши за основу Magic The Gathering, було створено унікальну гру, а найголовніше – зрозумілу для всіх. Гра має невисокий поріг входження і доступна на всіх платформах безкоштовно, що гарантувало їй популярність у всьому світі.

8) Тактичні стратегії Такі стратегії є симуляцією тактики як у реальному часі, так і в покроковому режимі, тим самим поділяючись на жанри RTT (Real-Time Tactics) і TBT (Turn-Based Tactics).

Тактика в реальному часі (англ. Real-time tactics - RTT) - жанр комп'ютерних ігор, ігровий процес якого є симуляцією військової тактики та оперативного мистецтва в режимі реального часу.

Покрокова тактика (англ. turn-based tactics - TBT) - це жанр стратегічної комп'ютерної гри, в якому ігровий процес розділений на окремі ходи. Ігровий процес найчастіше симулює бойові дії малого масштабу (як правило, на рівні не вище взводу або роти), використовуючи при цьому елементи військового оперативного мистецтва та військової тактики — на відміну від більш глобальних і покрокових стратегій, що тяжіють до більшого рівня абстрактності (TBS).

На відміну від класичних RTS, у цих стратегіях гравцеві не доводиться займатися менеджментом бази або видобутком ресурсів, тим самим дані гри сконцентровані переважно на тактиці ведення бойових дій в умовах заздалегідь заданої кількості військ та/або ресурсів. Можливість отримання підкріплень і додаткових ресурсів у таких іграх або зовсім немає, або на ній не робиться основний акцент. Приклади: World in Conflict, Achtung! Cthulhu Tactics, серія Jagged Alliance.

7) Окремим піджанром виносять також і браузерні/мобільні стратегії. У цьому типі гравець має свою базу, яку треба прокачувати, покращувати, будувати нові будівлі та захищати, а також створювати бійців, завдяки яким можна напасти на інших гравців. Присутні клани, таблиці рейтингів.

Це стратегії, що поєднують основні елементи класичної RTS з можливістю колективної гри з великою кількістю гравців через Інтернет. Прикладами браузерних онлайн-стратегій можуть бути ігри Травіан, My Lands, OGame, Сфера Долі, Війна племен та мобільна Clash of Clans – найпопулярніша гра цього жанру.

## **1.4 Розрахована на багато користувачів покрокова стратегія в реальному часі.**

Така стратегія є досить складним типом стратегії, який навіть не завжди виділяється в окремий піджанр. Два приклади таких стратегій будуть докладніше описані у Розділі 2.

У цьому типі, як то кажуть, «поєднується непоєднуване»: покрокові стратегії та стратегії в реальному часі. Однак, на практиці можна знайти дуже мало ігор подібного типу, але і знайдені ігри можна легко віднести до, наприклад, покрокових стратегій.

Принцип гри в таких додатках приблизно такий: гра розбита на кроки, що робить її покроковою, але кроки роблять усі учасники гри одночасно, що відносить її до жанру RTS.

Наприклад, серія ігор Total War поєднує у собі покрокову стратегію та управління ресурсами з тактичним контролем битв у реальному часі. Це досить унікально для стратегічних ігор у реальному часі, адже флангові маневри та побудови сильно впливають на ігровий процес.

Принцип планування ходів у другому прикладі (Frozen Synapse) дуже схожий на той, що використовується у грі Laser Squad Nemesis, розбирати який ми докладніше будемо у 2 Розділі.

Гравець керує невеликим збройним загоном солдатів, за допомогою яких необхідно виконати мету місії (найчастіше нею виявляється знищення всіх юнітів супротивника). Рухи плануються заздалегідь, необхідно роздати команди бійцям на тимчасовий ліміт у 5 секунд, цим займається опонент.

У процесі можна переглянути попередній результат, щоб можна було змінити якісь дії, якщо гравець знаходить їх невдалими.

Після натискання кнопки Prime відбувається калькуляція результатів планування обох опонентів. У процесі калькуляції враховуються такі фактори, як прицілювання, укриття, реакція (час для прицілювання) окремого юніту. Юніт,

який використовує укриття має перевагу над юнітом, який прицілюється, а він у свою чергу — перед юнітом, що не прицілюється..

Саме калькуляція дій, зроблених за однаковий час ходів є досить цікавим відмінним елементом даних ігор. Хоч серію ігор Total War офіційно і відносять відразу до двох жанрів «покрокова стратегія, тактика в реальному часі», а Frozen Synapse і зовсім лише до покрокової стратегії, але загалом і загалом, можна сказати, що вони утворюють невеликий, непопулярний підтип стратегій, який більш доцільно виділити та назвати «покроковою стратегією в реальному часі».

У рамках дипломної роботи фінальний додаток-прототип, ймовірно, матиме статус саме «покрокової стратегії», адже ніхто й ніколи раніше не вказував такий тип (або піджанр) стратегії, як «покрокова стратегія в реальному часі».

Однак, хотілося б висловити надію на популяризацію даного напряму та виділення цього типу стратегій в окремий піджанр.

Розрахований на багато користувачів фінальний прототип вважається за рахунок того, що гравці зможуть грати разом, об'єднуватися, змагатися. У додатку буде можливість грати і одному проти ботів, але основна мета – гра з іншими людьми.

Розрахована на багато користувачів гра на одному комп'ютері буває:

- Гравці беруть участь у грі одночасно. У спортивних симуляторах це природний режим. В інших іграх, наприклад, гонках, екран ділиться навпіл, і такий режим називається splitscreen.

- Гравці беруть участь у грі по черзі (англ. hotseat). Використовується в деяких покрокових стратегіях, наприклад Heroes of Might and Magic. У симуляторах Формули-1 Grand Prix 2, 3 і 4 також є режим почергової гри, коли один гравець веде свій болід, а іншими болідами управляє комп'ютер, при цьому кілька разів за гонку управління передається від одного гравця іншому.

- Один комп'ютер – кілька терміналів. На даний момент застосовується в основному в ігрових автоматах.

- Деякі ігри (в основному для ігрових приставок) підтримують мережеву гру, в якій на кожній машині можуть грати кілька гравців у режимі splitscreen.

Гра по мережі, кілька комп'ютерів з'єднані в обчислювальну мережу:

- Через послідовні чи паралельні порти.

- Через модем.

- Через локальну мережу та інтернет, за протоколами IPX або TCP/IP. (3D-шутери, стратегії в реальному часі, ...).

- Онлайн ігри

- Через власний клієнт: Ultima Online, Аллоди Онлайн.

- Браузерні ігри: Tale, Бійцівський клуб, Magic, Livegames.

- Через електронну пошту: PVEМ.

- Через спеціальний веб-сайт: Мафія.

- Через IRC (наприклад, вікторини).

- Через Bluetooth, IrDA, SMS (ігри для мобільних телефонів).

За правилами:

- За одним «життям». Цей режим застосовувався у деяких іграх перших ПК (наприклад, ZX Spectrum). У цьому режимі гравці зовсім не взаємодіють один з одним, кожен веде свою гру; коли гравець втрачає «життя», керування передається напарникові.

- Кооперативна гра. Мета гравців є спільною: разом перемогти комп'ютерних суперників. Цей тип гри з'явився набагато раніше за мережеві ігри.

- Особисті змагальні ігри: інтереси гравців перетинаються, і кожен гравець намагається перемогти інших.

- Командні змагальні ігри: гравці об'єднуються в команди, змагаються один з одним.

- Ігри з повною свободою, коли об'єднання у команди та війна один з одним повністю лежать на совісті гравців. Зрозуміло, зради не забороняються.



## Висновки по розділу 1

У цьому розділі було наведено множину різних визначень, що так чи інакше стосуються розробки покрокової стратегії, а також стратегії як жанру. Це необхідно для розуміння подальших дій у Розділах 2 та 3.

Для визначення перспектив направлення в 1 Розділ були додані графік (Рис. 1) та укорочена версія найприбутковіших додатків на мобільних пристроях, що дозволило зробити висновки, що створення кросплатформної гри для ПК та мобільних пристроїв має гарні перспективи.

Далі було приведено ознайомлювальна коротка історія жанру, у якій особливу увагу приділено появі нових типів стратегії, їх популярність у сучасному світі. Це є необхідним для правильного вибору напряму стратегії, який зможе виявитися популярним і в наш час.

Також було наведено коротку типологію, щоб вивчити цей жанр докладніше і знатися на нюансах сучасних стратегій. Окремим пунктом був винесений на даний момент неіснуючий, але такий, що добре описує ігровий процес мого прототипу, жанр.

Підсумовуючи і спираючись на досвід численних попередніх розробників та інді-розробників, можна сказати, що прототип гри має перспективи розвитку, а обраний напрямок має низьку конкуренцію і велику популярність.

## РОЗДІЛ 2. АНАЛІЗ ПОПУЛЯРНИХ СТРАТЕГІЙ

### 2.1 Покрокові стратегії

#### 2.1.1 *Civilization V*

Повна назва цієї стратегії - Sid Meier's Civilization V - це комп'ютерна гра компанії Firaxis, п'ята частина в серії глобальних покрокових стратегій Civilization.

Firaxis Games - американська компанія-розробник комп'ютерних ігор, найбільш відома за серією глобальних стратегій серії Civilization. Студія була заснована в 1996 році колишніми співробітниками MicroProse - Сідом Мейером, Джеффом Бріггсом і Брайном Рейнольдом і зараз знаходиться в Спарксі, штат Меріленд, США. З 2005 року входить до складу 2K Games, дочірньої компанії Take-Two Interactive.

Також компанія Firaxis Games додавала до всіх ігор, розроблених Мейером, приставку "Sid Meier's" - тенденцію, яку троє засновників перейняли у MicroProse, оскільки вони вважали, що ім'я Мейєра додає їх іграм більшої впізнаваності.

Перший реліз гри відбувся 21 вересня 2010 для Microsoft Windows у США; для Mac OS X гра була випущена 23 листопада 2010, для Linux — 10 червня 2014.

Як і раніше, місто залишається основним стрижнем, на яке нанизуються події гри, і навколо якого відбувається боротьба цивілізацій.

Міста тепер можуть обробляти територію на видаленні в три стільники замість двох клітин раніше. Розширення кордонів країни, щоправда, наразі здійснюється посотово. Гравець може в терміновому порядку освоїти

(представлено як купівлю) нову ділянку за гроші, але сусідні держави поставляться до цього негативно.

Міста можуть завдавати ударів по юнітам супротивника на відстані. Це виглядає як наявність вбудованого далекобійного юніту, що не відокремлюється. Оборона міста посилюється додатково за допомогою розміщення гарнізону із одного сухопутного бойового юніту. Тут можуть бути одночасно один бойовий юніт, один небойовий, один корабель і скільки завгодно повітряних юнітів.

При захопленні міст гравцеві дається вибір: анексія, створення сателіту, руйнування чи визволення. Останнє — коли це місто раніше було під владою держави-противника. Примітно, що якщо звільнити місто, яке належало знищеній раніше цивілізації, то вона повертається в гру з тим рівнем розвитку, який був перед припиненням її існування. Гравцеві не дозволяється безпосередньо керувати сателітами (містами-маріонетками), проте сателіти сплачують податки в скарбницю метрополії та дають доступ до територій та ресурсів, також не так негативно впливають на настрої населення, як при анексії.

Руйнування міста відбувається не миттєво, а займає кілька ходів, протягом яких руйнується місто стає частиною цивілізації, що захопила його, зі своїм населенням, вцілілими будинками, витратами та іншим. Хоча кінцевий результат необоротний, процес руйнування можна припинити будь-якої миті. При цьому не підлягають знищенню первісні столиці (найперше місто, засноване відповідною цивілізацією) переможених держав, а також усі міста-держави, що є столицями за визначенням. Також неможливо зруйнувати власноруч засновані міста, окрім випадку, коли вони відвойовані назад після захоплення чужою цивілізацією. Таким чином, до неруйнівних відносяться міста класу столиць, а також власні, жодного разу не окуповані противником. Міста, які неможливо зруйнувати, можуть створювати труднощі у плануванні забудови захопленої території.

Тепер розберемо те, що стосується мети. У грі Civilization V є вісім рівнів у міру посилення складності від «Поселенця» до «Божества».

Рівень "Князь" є усередненим рівнем складності. При грі на цьому рівні ні в кого немає переваг. Швидкість гри змінюється від «Марафонської» через «Епічну» та «Середню» до «Високої».

Відмінною особливістю деяких цивілізацій є залежність вибору стартової позиції від розташування океанів і річок, як і від цивілізаційного переваги/уникнення відповідних географічних поясів. Наприклад, стандартною установкою для Росії є перевага тундри, але якщо на карті немає пояса тундри, буде знаходитися пояс, що має вкраплення тундрового ландшафту, і в той же час

Сонгай уникатиме холодної тундри. Єгипет зазвичай не знайти поблизу лісів та джунглів, зате Ірокези завжди знаходяться серед лісів тощо.

Як і в попередніх версіях, гравець має наступні варіанти перемоги (як і раніше, у грі відсутній механізм «економічної перемоги», хоча підрахунок економічного рівня ведеться):

За сумою очок (вона ж перемога часом при настанні 2050 р. н. е.);

Наукова (вона ж технічна – запуск космічного корабля з колоністами до Альфи Центавра);

Територіальна (вона ж військова – утримання останнім у грі своєї столиці);

Культурна (отримання всіх застосовних громадських інститутів із п'яти гілок з побудовою «Проекту „Утопія“»);

Дипломатична (отримання титулу Світового лідера під час голосування в ООН).

Таким чином, є три шляхи до поразки: втратити всі свої міста, програти іншій цивілізації в одній ситуації з перерахованих вище або не набрати найбільшу кількість очок до 2050 року. Ця гра досі є найпопулярнішою із серії Civilization. Були проведені множинні аматорські (і професійні) порівняння найновіших 5 та 6 частин. На їх підставі можна виділити основні фактори, що впливали на вибір гравців:

- Якість гри більшою мірою не змінилося, залишилося на колишньому рівні. Загалом, ті, хто сказав, що це трохи чи набагато краще, перевершили тих, хто сказав, що це трохи чи набагато гірше – 43,2 відсотка проти 31,8 відсотка відповідно. Якщо ви складете всіх тих, хто сказав, що вона, принаймні так само гарна, як і старі Civ, ми побачимо, що близько 68 відсотків не розглядають Civ 6 як крок у неправильному напрямку, якщо врахувати всі обставини.

- При постійному якості, вартість 6 частина перевищувала ціну 5 частина. Таким чином виходило, що гравці начебто купують таку саму гру вдруге за великі гроші.

- Більшість опитаних не вірили в успіх 6 частин (див. рис. 2.1).

- П'ята частина менш вимоглива до «заліза». Не кожен користувач зможе повноцінно запускати і грати в 6 частину через вищу якість текстур і, відповідно, повільнішу відповідь гри на запити.

- Найменша наповненість контентом 6 частини. З цим пов'язано те, що багато гравців висловили ідею про те, що після глобальних оновлень для 6 частини вона легко обійде за якістю п'яту (див. рис. 2.1).

- Більш звичний інтерфейс та механіки 5 частини досі залучають гравців своєю відносною простотою.

- Користувачі, які відрізнялися великим досвідом подібних ігор, заявляли, що в новій частині їх не влаштовує ухил гри, вибраний розробниками. Якщо раніше ухил був у бік саме стратегії як жанру, то в новій частині наголос був не на тому, що так надихало гравців попередніх поколінь, а саме – на запровадження мікроменаджменту.

- Для звичайного користувача, що не є фанатом даної серії, цілком підійде 5 частина для рідкісного проведення часу.

- 6-а частина є новою, а отже, акції та розпродажі не розповсюджуються на неї. А ось 5-у частину цілком можна «виловити» за ціною в 5 разів меншою за заявлену.

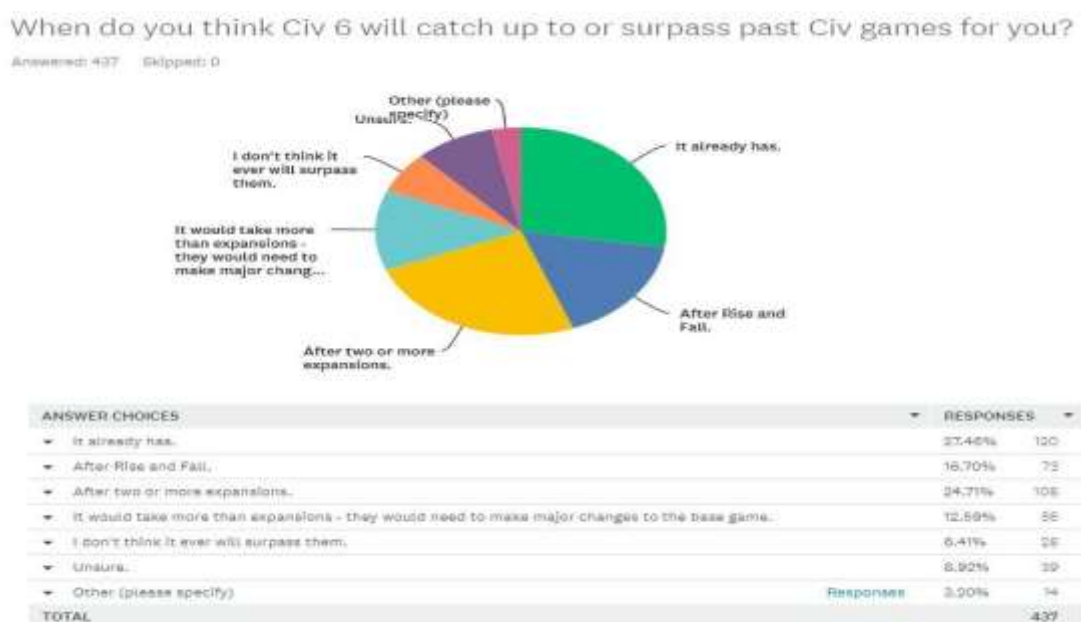


Рис.2.1. Думка гравців про 5 і 6 частин Civilization.

Підсумовуючи, можна зробити кілька висновків про те, що є привабливим в іграх для звичайного користувача. З метою упорядкування накопиченого матеріалу усі отримані висновки в короткому вигляді будуть описані у підсумках Розділу 2.

### *2.1.2 Heroes of Might and Magic III*

Heroes of Might and Magic (скорочено Heroes, HoMM; у російських виданнях «Герої меча і магії» або просто «Герої») — фентезійна серія комп'ютерних ігор у жанрі покрокової стратегії з елементами RPG, що на даний момент включає сім основних частин. Перша гра серії Heroes of Might and Magic: A Strategic Quest була запланована як спін-офф від основної серії Might and Magic, але пізніше перетворилася на самостійну франшизу. Назви шостої та сьомої частини серії Heroes, на відміну від попередніх, мають формат Might & Magic Heroes VI та Might & Magic Heroes VII відповідно до нової політики найменування всіх ігор, що належать бренду Might and Magic.

New World Computing, Inc., також відома як NWC – колишня американська компанія-розробник та видавець комп'ютерних ігор. Заснована в 1984 році Джоном Ван Канегемом (Jon Van Caneghem) та Марком Калдвелом (Mark Caldwell). Найбільш широко відома як розробник та видавець популярної серії РПГ Might and Magic та її відгалужень, особливо Heroes of Might and Magic. Компанія була придбана 3DO у 1996 році. У 2003 році, з банкрутством 3DO, New World Computing також зникла. У тому ж році компанія Ubisoft купила права на серію Might and Magic та серію Heroes of Might and Magic за 1,3 мільйона доларів.

Вигаданий всесвіт Heroes of Might and Magic заснований на всесвіті Might and Magic. Події всіх ігор серії до Heroes of Might and Magic IV розгортаються у вигаданому світі Енрот (Enroth), тоді як у четвертій частині дія перенеслася у новий світ під назвою Аксеот (Axeoth). Починаючи з Heroes of Might and Magic V, коли авторські права були продані компанії Ubisoft, дія відбувається в повністю новому вигаданому всесвіті — світі Асхан (Ashan).

Heroes III залишається найпопулярнішою грою в серії та однією з найпопулярніших покрокових стратегій. The Restoration of Erathia стала єдиною версією Heroes III, випущеною для трьох платформ Microsoft Windows, Apple Macintosh і Linux. Остання була розроблена компанією Loki Entertainment Software, яка займалася портуванням ігор на Linux. 29 січня 2015 року компанія Ubisoft, яка перейняла у The 3DO Company права на серію Heroes of Might and Magic, випустила HD-версію Heroes of Might and Magic III: The Restoration of Erathia з покращеною графікою та підтримкою платформ iOS та Android, але відсутністю обох доповнень до оригінальної гри.

Heroes of Might and Magic III розвиває ідеї, закладені у першій та другій частинах серії. Суть гри залишилася незмінною: гравець керує героями, які ведуть у бій армії істот. Основна дія гри відбувається на стратегічній карті пригод, якою переміщуються герої, захоплюючи при цьому міста і відвідуючи всілякі будівлі на карті. Міста є важливою складовою гри: гравець може розвивати їх, зводячи в них різні будівлі, які дають доступ до більш потужних істот, магічних заклинань та інших бойових переваг; Міста також є джерелами доходу. Будувати нові міста в Heroes III не можна, можна лише захоплювати ті, що вже є на карті. Гра є покроковою стратегією, що означає, що в ній дискретно: основний його одиницею є хід, іменований також днем. За один ігровий день гравець може вчинити будь-які доступні дії, обмежені лише його потребами та запасом ходу у його героїв (або обмеженням часу, якщо таке було виставлено до початку гри). У день можна зводити по одній будівлі в кожному з міст, що належать гравцю. Коли герой вступає в битву, гра переключається на екран бою, що віддалено нагадує шахи: дві армії розташовані по різні боки екрану, і групи істот, по черзі роблячи хід, повинні знищити ворожі війська.

Одним із основних аспектів серії Heroes of Might and Magic є розвиток героя. В основному, це робиться шляхом отримання досвіду та підняття героя на новий рівень, з новими навичками та параметрами. Основним джерелом досвіду є перемога у битві. Після битви герой, що переміг, отримує стільки досвіду, скільки

здоров'я було в сумі у всіх загиблих істот армії, що програла. У разі програшу битви герой нічого не отримує. Шкала досвіду прогресуюча, тобто для переходу на кожен наступний рівень необхідно набирати все більше досвіду. Крім героїв супротивника, на карті у великій кількості представлені нейтральні істоти — створення, які не належать нікому з гравців і беруть участь у грі «самі за себе». Зазвичай нейтральні істоти охороняють шахти ресурсів, артефакти тощо цінні об'єкти на карті, але також можуть охороняти нейтральні міста або гарнізони. Нейтральні істоти завжди нерухомі не можуть атакувати героя; битва з ними можлива, тільки якщо на них напасти. На початкових стадіях гри битви з нейтральними істотами дозволяють розвинути героя, підготувавши його до битв із противниками-гравцями.

Таким чином, в іграх серії Heroes of Might and Magic гравцю доведеться боротися з іншими людьми або комп'ютерними противниками за володіння містами, джерелами ресурсів, скарбами та артефактами. У процесі гри він керує героями — ігровими персонажами, які досліджують глобальну карту та очолюють армії істот під час бою, а також здатні застосовувати магію. Виграючи битви, герої набувають досвіду, і, набравши достатню його кількість, переходять на наступний рівень, збільшуючи свої параметри. Бої відбуваються на суші та на воді, у відкритому полі та при облозі замків, з ворожими героями та нейтральними істотами. Істоти наймаються в містах, також вони можуть бути найняті або приєднані на карті. Міста є центральним елементом серійних ігор. Вони будуються будівлі вивчення заклинань, доступу до нових бійцям, посилення обороноздатності тощо.

І битви, і подорожі відбуваються в покроковому режимі, а кінцева мета гри може полягати в різних речах — наприклад, знищення противників чи певного героя, захоплення міста чи артефакту.

Випущена в 1999 році гра Heroes of Might and Magic III визнається однією з кращих покрокових стратегій і однією з найпопулярніших ігор у серії Heroes of Might and Magic



За рейтингом, заснованим на думці читачів Absolute Games, гра неодноразово виборола високі місця на конкурсах кращих ігор 1999 і 2000 року:

«Найкраща стратегія 1999 року» - 1-е місце;

«Найкраща стратегічна гра ХХ століття» - 1-е місце;

«Найкраща гра ХХ століття» - 8-е місце.

І знову питання: є четверта частина, є п'ята частина, а люди продовжують грати в третю, чому? Із плюсів цієї гри зазначають, що:

1) Гарний баланс. Це виступає як плюсом, і мінусом гри. Були в опитуванні і нечесні гравці, які віддають перевагу 4-й або 2-ій частині в надії на легшу гру за рахунок «лазівок». Однак, таких меншість.

2) Гра складна, але цікава. Відмінно підходить для людей, які готові «день і ніч» розбиратися в численних механіках ігрового світу та шукати для себе найкращу тактику ведення бою.

Також гра добре підійде тим, хто любить довго вирішувати загадки та головоломки.

3) Ця гра має величезну кількість фанатів, які робили, роблять і робитимуть різноманітний контент для гри. При чому це може виражатися як у написанні статті-«гайда», знятті відеоролика, так і у створенні карток користувача, доповнень.

Це чудова можливість отримати нові емоції від гри, яка, ймовірно, за кілька років може набриднути. І це також полегшує завдання розробникам.

4) Ще один бонус від фанів – це постійний онлайн у грі. Відомі випадки, коли гравці не спали ночами заради гри. Це добре як для розробників, так і для простих користувачів, які можуть легко знайти з ким зіграти.

5) графіка. Дуже неоднозначна перевага. Як відомо, різні люди мають різні смаки. Однак із тих людей, кому подобалася 3 частина, більшість вписала у переваги саме графіку, яка, на відміну від інших частин, відрізнялася «яскравістю, барвистістю, привабливістю».

Більше того, є думка, що це перевага лише побічний ефект від фанатів чи «загартованих гравців», які звикли до гри і не хочуть нічого міняти роками.

6) Унікальність. Людині, не знайомій з цією грою, складно назвати її унікальною, враховуючи все різноманіття ігор-стратегій, у тому числі і 6 ігор серії магії. Проте, любителі та професіонали з цієї гри відзначають її виняткову неповторність та унікальність серед інших проектів.

7) Мультиплеєр. Багато гравців зазначали, що в цій частині мультиплеєр особливо гарний та зручний для гри з друзями.

8) Музичний супровід. Мелодії захоплюють гравця у світ фентезі, бо композитор протягом майже всієї серії Героїв один і той же, то ймовірно, що безліч гравців буквально зростали під ці мелодії. Це не може не впливати на їхнє сприйняття музики з гри.

9) Варіативність. Гра має величезний потенціал у цьому плані.

Таким чином, гра відзначилася насамперед часом, який вона протрималася на плаву. Далеко не кожна гра може зібрати так багато фанатів різного віку.

Але і це не дивно, адже для свого часу вона була чимось на кшталт прориву, завершення основної трилогії, яку самі собі написали фанати, адже після 3-ї частини всі права на володіння були продані більшій компанії, про яку говорилося раніше.

Heroes of Might and Magic 3 – це визначальна гра серії. Є Герої до і є після. Так чи інакше, всі наступні ігри серії робилися з огляду на третину.

Але не варто думати, що HoMM 3 хороша просто тому, що так склалися зірки. Розробниками оцінили реакцію спільноти на другу частину, обміркували її плюси та мінуси.

HoMM 3 - не в останню чергу відповідь розробників на проблеми минулої частини і в той же час спроба зберегти те, що подобалося гравцям. Окремо відзначимо, що гра все одно досить консервативна. Вона йде шляхом еволюції та розвитку, а не кардинального перегляду ідей, як та ж HoMM 4.

Якщо брати нові дані, статистику по онлайн, то можна помітити, що гравці і самі розуміють, що оригінальна НОММ 3 все ж таки застаріла.

Про більш конкретні висновки буде написано наприкінці розділу 2.

## **2.2 Стратегії у реальному часі**

### *2.2.1 StarCraft II*

StarCraft II - військова науково-фантастична відеогра, створена Blizzard Entertainment як продовження успішної відеоігри StarCraft, випущеної 1998 року. Дія гри розгортається у вигаданому майбутньому та зосереджена на галактичній боротьбі за панування серед трьох різних вигаданих рас всесвіту StarCraft.

Blizzard Entertainment (скор. Blizzard) - американський розробник та видавець комп'ютерних ігор. Компанія була заснована 8 лютого 1991 року трьома випускниками Каліфорнійського університету в Лос-Анджелесі: Алленом Адамом, Френком Пірсом та Майком Морхеймом. Раніше називалася Silicon & Synapse, потім Chaos Studios, а пізніше стала підрозділом видавництва Activision Blizzard. Штаб-квартира Blizzard Entertainment розташована у місті Ірвайн, штат Каліфорнія. Спочатку компанія була націлена на розробку портованих версій ігор інших студій. Після цього вона почала створювати своє власне програмне забезпечення і в 1993 році почала з розробки таких ігор, як Rock n' Roll Racing і The Lost Vikings. Після випуску Warcraft у 1995 році студія Blizzard Entertainment стала однією з найуспішніших у всьому світі. Потім було створення нових комп'ютерних ігор, у тому числі сиквели Warcraft, StarCraft, Diablo, Overwatch і World of Warcraft.

Однокористувацька кампанія StarCraft II розділена на три частини, кожна з яких присвячена одній з трьох рас: StarCraft II: Wings of Liberty (випущена в 2010 році), Heart of the Swarm (2013) та Legacy of the Void (2015). Фінальний пакет кампанії під назвою StarCraft II: Nova Covert Ops була випущена у 2016 році.

Розрахований на багато користувачів ігровий процес StarCraft II породив окреме кіберспортивне змагання, яке пізніше привернуло інтерес з боку інших

компаній, крім Blizzard, і привернуло увагу в Південній Кореї та інших країнах, подібно до оригінального кіберспортивного змагання StarCraft.

Режими розрахованої на багато користувачів гри:

#### 1) Протистояння.

Режим протистояння може бути 1 на 1, 2 на 2, 3 на 3, 4 на 4 з цифрами, що вказують кількість гравців у команді. Режим 1 на 1 є найбільш відомим та основою для кіберспортивних змагань. У ньому гравці набирають очки досвіду та досягнення, але це не впливає на майбутній ігровий процес як такий. Складність гри зростає в міру просування гравця сходами звань, заснованої на системі підбору гравців, що базується на кількості перемог та поразок. Так звана "сходи", порядок розташування всіх гравців, організований по географічному регіону і складається з рівнів, званих лігами, починаючи від нижчої (бронзової, срібної, золотої, платинової, діамантової, майстер-ліги) і, у свою чергу, вищої, гросмейстерської ліги.

У міру зростання кількості гравців у команді зазвичай змінюються і карти, на яких вони можуть грати. Також можливо брати участь у розрахованих на багато користувачів турнірах в командах будь-якого розміру, де команда або перемагає, або вибуває. Режим versus також включає опцію "Archon", де матчі проходять нібито 1 на 1, але більше одного гравця-людини контролюють те, що зазвичай контролюється однією людиною.

#### 2) Кооператив.

У кооперативному режимі два гравці-людини грають проти П, використовуючи спеціальних героїв, і де накопичення досвіду призводить до підвищення рівня, що, своєю чергою, впливає на майбутній ігровий процес. Герої створені на зразок персонажів одиночної кампанії з додаванням додаткових здібностей.

У цьому режимі також є так звані "мутації", коли параметри гри змінюються, щоб створити більш складне завдання. Усього на вибір пропонується 18 командирів кооперативу, які порівну розділені між трьома расами (по 6 від

кожної). У кооперативному режимі використовуються герої з основної сюжетної лінії гри, хоча не всі герої (46 героїв) є ігровими персонажами.

Wings of Liberty була випущена у 2010 році, через чотири роки після закінчення StarCraft: Brood War. Два доповнення Heart of the Swarm і Legacy of the Void (обидва в даний час є самостійними іграми) були заплановані з самого початку; перше було випущено у 2013 році, а останнє – у 2015 році.

Всі ігри основної серії є стратегічними іграми у реальному часі, в яких гравець розглядає події як військовий командир для кожного із трьох видів.

StarCraft II: Wings of Liberty – офіційне продовження StarCraft, випущене для Windows та Mac OS X компанією Blizzard Entertainment 27 липня 2010 року. Гра була анонсована на Worldwide Invitational у Південній Кореї 19 травня 2007 року з попередньо відрендереним трейлером кат-сцени та демонстрацією ігрового процесу протосів. Подальші демонстрації нових можливостей гри були продемонстровані на наступних BlizzCons та інших ігрових конвенціях. Гра включає новий двигун 3D-графіки і додає нові функції, такі як фізичний двигун Havok. StarCraft II також включає ефекти рівня DirectX 10 у Windows. Спочатку задумана як окрема гра, StarCraft II у процесі розробки була поділена на три частини, по одній для кожної раси. Основна гра, Wings of Liberty, розповідає про земляни, в той час як два доповнення, Heart of the Swarm та Legacy of the Void, були випущені, щоб доповнити Wings of Liberty і розвинути історію з погляду зергів та протосів відповідно. Історія Wings of Liberty триває чотири роки після завершення Brood War і обертається навколо боротьби Джима Рейнора проти Домініона Терран.

StarCraft II: Heart of the Swarm – додаток до StarCraft II: Wings of Liberty, випущений 12 березня 2013 року. Це друга частина трилогії StarCraft II. Додаток включає додаткових юнітів і розраховані на багато користувачів зміни з Wings of Liberty, а також кампанію, присвячену Керріган і расі зергів. Він охоплює 27 місій (20 основних місій та 7 побічних місій).

Сага про StarCraft зрештою завершена StarCraft II: Legacy of the Void, яка була випущена 10 листопада 2015 року. StarCraft II: Legacy of the Void - це автономна гра, в якій нові юніти додаються до всіх трьох рас, а також змінюються існуючі юніти, а також вносять кардинальні зміни в економічний аспект гри. Історія StarCraft завершується слідуванням за расою протосів у їхньому прагненні повернути свій рідний світ, а Керріган зрештою знищити найбільшу загрозу для всього всесвіту. Гра розділена на пролог з 3 місій, основну сюжетну кампанію з 19 місій та епілог з 3 місій, який завершує все.

На BlizzCon 2015, під час презентації "Майбутнього StarCraft II", було оголошено, що Blizzard випустить додаткові пакети місій, щоб гравці могли взаємодіяти зі StarCraft II. Набір місій Nova Covert Ops складається з трьох епізодів, загалом із дев'ятьма новими місіями. Для гри не потрібно купувати StarCraft II, і в неї можна було грати зі стартовим виданням. В той же час Blizzard оголосила, що планується додати нових командирів до кооперативного режиму Legacy of the Void як DLC, причому Karax стане першим безкоштовним доповненням.

З моменту свого виходу в StarCraft II професійно грали по всьому світу, хоча, як і у випадку з професійним змаганням StarCraft: Brood War, його попередником, найвищий рівень гри був зосереджений у Південній Кореї.

Гра вважалася найбільшою кіберспортивною грою у світі в перші роки свого існування, і вважалася, що вона принесла кіберспорт решті світу так само, як оригінал приніс його до Південної Кореї. З того часу вона пережила занепад і недавнє відродження після переходу до бізнес-моделі free-to-play.

У 2012 році Blizzard запустила серію чемпіонатів світу StarCraft II (WCS) як основний санкціонований турнір StarCraft II. З 2013 року як корейські індивідуальні ліги, такі як Глобальна ліга StarCraft II (GSL), так і некорейські змагання, такі як Intel Extreme Masters (IEM) та Dreamhack, були включені до системи WCS, розподіляючи очки та гарантуючи місця, які кваліфікують гравців у Глобальний фінал, який щорічно проводиться на BlizzCon.

З 2020 року Blizzard змінила формат WCS, уклавши трирічне партнерство з організаторами кіберспорту ESL та DreamHack. Реліз StarCraft II: Wings of Liberty пройшов дуже добре з комерційної та критичної точки зору, було продано 1,8 мільйона копій протягом перших сорока восьми годин після релізу, що побило рекорд стратегічної гри, що найбільше продається в історії ігрової індустрії. Вона отримала дуже позитивні відгуки із загальним показником GameRankings 93% і була номінована як "Найкраща комп'ютерна гра 2010 року" на GameSpot. До кінця 2012 року було продано понад 6 мільйонів копій Wings of Liberty. Цей успіх продовжився з випуском першого доповнення StarCraft II Heart of the Swarm, сукупний рейтинг гри якого склав 86%. Автономний пакет розширення був проданий тиражем 1,1 мільйона копій протягом перших двох днів після його випуску 12 березня 2013 року і став грою, що найбільше продається для ПК в цьому кварталі. StarCraft II: Legacy of the Void, третє доповнення, було також добре зустрінуте, отримавши сукупний рейтинг GameRankings у 88%, при цьому було продано понад 1 мільйон копій по всьому світу в перший день його випуску.

До кінця року було продано понад 17,6 мільйонів копій ігор та доповнень для StarCraft та StarCraft II. 2015. До кінця 2017 року Blizzard включила об'єднану франшизу StarCraft до списку своїх брендів, сукупний дохід яких за весь період становив понад 1 мільярд доларів.

Далі будуть розписані характерні риси Starcraft II (для одних деякі характеристики – це плюси, для інших – мінуси):

1) Це не зовсім командна гра. Більшість матчів, турнірів та ін. проходять у режимах 1 x 1. Звичайно, у грі присутній мультиплеєр, можна грати, як уже було сказано і 2 x 2, і 3 x 3, проте тільки як розвага.

Багато хто зазначає, що грати разом досить важко, навіть якщо не брати до уваги турніри. Коли людину одну – вона чітко бачить де помилився, розуміє, що це її вина. Коли людей кілька, буває таке, що починається хаос та взаємні звинувачення. Але насправді це суто справа смаку та особистої комунікабельності, хоча й більш-менш єдиної думки тут немає.

2) Другий пункт є актуальним для кіберспортсменів – у цій грі легко простежуються результати тренерівок.

Якщо в багатьох дисциплінах вирішує везіння чи неймовірна випадковість, то тут щоразу виходитиме все краще і краще саме за рахунок навчання та покращення наявних навичок.

3) Гарний баланс. Порівняно з багатьма іншими іграми, наприклад, жанром MOBA, Starcraft, навіть маючи якісь тимчасові труднощі з балансом, може не завадити гравцям продовжувати гру в нормальному темпі.

Більше того, ця гра найменше піддається впливу оновлень. Вони, безумовно, присутні, проте не «ламають» баланс настільки, як це відбувається в тих же MOBA-х.

4) Різноманітність. У грі є безліч різних карт, гравці та вороги з'являються скрізь по-різному. Щоразу по-новому використовувати ресурси, організовувати їх видобуток, атаку, відволікання.

Навіть якщо (з неймовірно маленьким шансом) і вийде повна копія стартової позиції, навіть противник буде той самий, гравець все одно вже діятиме інакше. Він знатиме, що очікується від противника і як цьому перешкоджати.

5) Висока динаміка гри. У порівнянні з багатогодинними (або тижневими) іграми у тих же Героїв з п. 2.1.2 тут ігри досить швидкі. Навіть коментатори не завжди встигають оглянути все, що відбувається.

Знов-таки, деякі люди віддають перевагу довгим іграм, інші – динамічні. Тому не можна сказати, що одна з цих характеристик є гідністю, а інша – недоліком.

### 2.2.2 *Age of Empires II*

Age of Empires II: The Age of Kings (тобто. Епоха імперій II. Вік королів) - відеогра в жанрі стратегія в реальному часі, розроблена Ensemble Studios і випущена Microsoft. Є другою грою із серії Age of Empires. Була випущена в 1999 році, а через рік побачило світ доповнення The Conquerors, що містить нові сюжетні лінії, ігрові цивілізації та вдосконалений ігровий процес.



Ensemble Studios – американська компанія, розробник комп'ютерних ігор. Була заснована у 1995 році як незалежна приватна компанія, проте у 2001 році була придбана Microsoft. Розпущена у 2009 році з економічних міркувань.

На рахунку студії безліч ігор, включаючи загальноновизнані хіти – серії Age of Empires та Halo Wars. Також Ensemble Studios розробила свій двигун Genie Engine, використаний в Age of Empires, Age of Empires II: The Age of Kings і Star Wars: Galactic Battlegrounds.

Події оригінальної The Age of Kings відбуваються в середньовіччі, а кількість доступних для гри цивілізацій дорівнює 13. Гравцю потрібно збирати ресурси, відбудовувати міста і створювати армії, за допомогою яких він повинен долати супротивників. Є п'ять кампаній, що ґрунтуються на історичних подіях, у яких гравцям ставляться певні умови, пов'язані з сюжетом та необхідні для перемоги. Крім них, є три додаткові режими одиночної гри, а також мультиплеер.

9 червня 2019 року було анонсовано, а 14 листопада 2019 року офіційно випущено друге перевидання гри під назвою Age of Empires II: Definitive Edition. Воно включає перемальовану графіку в роздільній здатності Ultra HD 4K, записаний заново саундтрек, а також доповнення The Last Khans з 3 новими кампаніями.

Дія гри відбувається в реальному світі, умовно охоплюючи період Середньовіччя, починаючи з падіння Римської імперії і закінчуючи Епохою Відродження. Гравець управляє однією з 16 (з усіма доповненнями - 37) представлених народностей, видобуваючи ресурси та відбудовуючи поселення за допомогою робітників, розвиває економіку, вивчає нові технології та створює армію. Основна мета гри – знищити своїх суперників, які паралельно розвивають свою цивілізацію на довколишній території.

У грі є досить розвинена економічна система. Є 4 види ресурсів - дерева, їжа, золото та камінь.

1) Дерева використовуються в основному для будівництва будівель, створення лучників, облогових знарядь, флоту, а також для будівництва ферм, які

у свою чергу виробляють їжу. Видобувається він шляхом рубки дерев і складування на найближчій тартаку. Більшість карток мають більш ніж достатню кількість цього ресурсу, причому розосереджений він по карті відносно рівномірно.

2) Їжа - мабуть, найбільш затребуваний ресурс у грі. Вона необхідна для будівництва робітників, для створення піхоти та кінноти, для більшості апгрейдів для військ. Також дуже велика кількість їжі потрібна для переходу у вищу епоху - ключових апгрейдів, необхідних для створення будівель і юнітів нового типу. Добуватися може багатьма способами. На початкових етапах гри переважно добувається шляхом полювання різних тварин (вівці, олені, кабани), шляхом збирання (ягідні куші), і навіть шляхом риболовлі за наявності на карті водойм з рибою. Рибалка може вестися селянами біля берега, або глибоководдя з допомогою рибальських судів. На пізніших етапах ці ресурси починають вичерпуватися, а їжі потрібно дедалі більше. Так гравець плавно переходить до будівництва ферм, які у достатній кількості забезпечують повністю автоматизований та постійний приріст цього ресурсу.

3) Золото необхідно в основному для створення найпотужніших бойових юнітів, а також високорівневих апгрейдів. На початкових стадіях гри грає значно скромнішу роль. Може видобуватися з копалень, точково розташованих по всій карті, проте це джерело швидко вичерпується, і тоді між гравцями виникають справжні сутички за родовища, що залишилися. Також може проводитись за рахунок сухопутної або морської торгівлі за наявності союзних гравців.

4) Камінь потрібний для будівництва замків, веж та кам'яних стін. Видобувається єдиним способом шляхом розробки робітниками родовищ, розташованих на карті аналогічно золотим родовищам, але в дещо меншій кількості.

У грі є можливість змінювати ресурси на ринку за певним курсом. На початку гри вартість ресурсів приблизно однакова, на пізніх етапах золото

скуповується гравцями в обмін на їжу, ліс та іноді камінь, що призводить до знецінення цих ресурсів (інфляція).

Як правило, гра починається з темних століть, коли доступними гравцеві є лише найпростіші будівлі та юніти. Кожен гравець має відбудований міський центр, не менше трьох селян та кінного розвідника. Початкові завдання - розвідка прилеглої території та видобуток значної кількості їжі, необхідної для безперервного створення нових робітників та подальшого переходу у феодальну епоху. Ліс видобувається в менших кількостях, золото та камінь – опціонально. З юнітів доступні лише робітники, ополченці (слабка піхота, створюється в казармах) та рибальські судна (у доках).

Феодальна епоха є ключовим моментом для вибору подальшої стратегії розвитку або будівництво армії для негайної атаки противника, або швидкий перехід в епоху замків. Останнє може на якийсь час поставити гравця у вразливе становище, зате дасть значні переваги в майбутньому. З військових юнітів стають доступні лучники, метачі списа, легка кавалерія та пікінери, також можливе покращення ополченців до мечників.

Основне нововведення замкової епохи — це можливість будувати додаткові міські центри, в яких можна буде одночасно створювати нових робітників. Це дає дуже сильний поштовх для розвитку економіки, проте вимагає спочатку чималих вкладень. Важливо правильно вибрати час для вливання інвестицій в економіку і не забувати при цьому про оборону та напад.

Зрештою, можливий перехід в останню імперську епоху. Цей перехід є надзвичайно дорогим і потребує багато часу, тому відбувається він при вже повністю налагодженій економіці. Імперська епоха дозволяє здійснювати найвищі рівні апгрейди на армію, будівлі та економіку. Також існує «постімперська епоха», яка умовно наступає після здійснення гравцем усіх можливих апгрейдів.

На додаток до оригінальної гри Age of Empires II: The Conquerors, а також у двох наступних є карти, що представляють території з реального світу (Скандинавія, Центральна Америка, Японське море).

У грі з кількома гравцями діє дипломатія – вони можуть бути союзниками, зберігати нейтралітет чи вести війну. Розрахований на багато користувачів режим підтримує до 8 гравців.

Важливою частиною буде думка гравців щодо цієї гри. А воно було наступним:

1) З одного боку, з'явилося багато нового, що виправдовує існування другого перевидання, але не всі експерименти розробників вдалися.

2) Думки щодо графіки розділилися. Одні хвалять її красу та атмосферність, а інші лають за мильність і непотрібні спецефекти, які збивають з пантелику.

3) З приводу мультиплеєра гравці також розділилися у думках.

Люди, у яких була своя компанія для цієї гри, проблем практично не помічали, хіба що були скарги на малу кількість функцій, неможливість вибору деяких налаштувань, кастомізацій.

Люди, які шукають компанію в мережі, були вкрай незадоволені тим фактом, що за наявності кількох тисяч людей у грі, не було серверів, на яких можна було б пограти з іншими, а рейтинговий підбір міг тривати годинами.

4) Але й довгий пошук себе не виправдовував: супротивник виявлявся або сильнішим, або сильнішим. Іншими словами, баланс сил у цій грі або був зламаний, або був відсутній зовсім.

Однак, розробники працюють над цією проблемою з самого релізу, тому цілком можливо, що саме ці дані застаріли.

5) Занадто мало статистики для перегляду. Гравці хочуть дізнатися щось про інших гравців, щоб, наприклад, провести з ними партію або прийняти їх до себе в команду, але, переглянувши профіль, у всіх завжди залишалися питання щодо рівня гри іншої людини.

6) Проблеми із локалізацією. Така проблема є у багатьох проектах на запуску, тому, ймовірно, що ці дані також застаріли.

## 2.3 Покрокові стратегії реального часу

Як говорилося раніше (у розділі 1), такого жанру практично немає. Однак тут були зібрані ігри, максимально наближені до такого складного і неоднозначного жанру, а головне – до фінального прототипу цієї дипломної роботи.

Основний жанр цих ігор – покрокова стратегія, але дуже незвичайна.

### 2.3.1 *Total War: Shogun 2*

Total War – серія стратегічних ігор, розроблених британським розробником Creative Assembly для персональних комп'ютерів. Вони поєднують покрокову стратегію та управління ресурсами з тактичним контролем битв у реальному часі.

Creative Assembly (офіційно The Creative Assembly, Ltd., літер. рус. Творчі Збори) - британська компанія-розробник комп'ютерних ігор, дочірня компанія корпорації Sega. Заснована 28 серпня 1987 року Тімом Анселлом (англ. Tim Ansell). Компанія найбільш відома за серією стратегічних комп'ютерних ігор Total War та survival horror грі Alien: Isolation. 23 березня 2005 року Creative Assembly була придбана компанією Sega.

Досить унікально для стратегічних ігор у реальному часі, флангові маневри та побудови сильно впливають на ігровий процес. Перша із серії, Shogun: Total War, була випущена у червні 2000 року. Останньою великою грою, випущеною 17 лютого 2022 року, була Total War: Warhammer III. Станом на жовтень 2022 року було продано понад 40,4 мільйона копій серії.

Найпопулярнішою частиною цієї серії виявилася стратегія Total War: Shogun 2. Shogun 2 отримав високу оцінку критиків від рецензентів, часто за спрощення та доопрацювання серії, повернувшись до її коріння. Окреме доповнення Total War: Shogun 2: Падіння самураю було випущено у 2012 році.

Total War: Shogun 2 - стратегічна відеогра, розроблена Creative Assembly та видана Sega у 2011 році. Є частиною серії Total War і повертається до японської обстановки 16 століття першої гри Total War, Shogun: Total War, після серії ігор, дія яких відбувається в основному в Європі і на Близькому Сході.

Дія "Сегуна 2" розгортається у феодальній Японії 16 століття, після війни Онін за часів сьогунату Асікага. Країна розколота на клани на чолі з місцевими воєначальниками, кожен з яких бореться за контроль. Гравець бере на себе управління одним із цих кланів з метою домінування над іншими фракціями та встановлення влади над Японією. У стандартному виданні гри представлено загалом вісім фракцій (плюс дев'ята фракція для навчального посібника), кожна з яких має унікальну стартову позицію та різну політичну та військову міць. Обмежене видання включає ексклюзивний клан ніндзя Хатторі, а доповнення відкриває десятий клан Ікко-Іккі. Гра відходить від європейського сеттингу попередніх ігор Total War і повертається до першого сеттингу серії Total War, але вносить суттєві зміни до основних елементів ігрового процесу Shogun 2. У порівнянні з Empire, яка охопила майже всю земну кулю, нова частина фокусується лише на островах Японії (включаючи Хоккайдо) та зменшену кількість типів юнітів.

Гра Shogun 2, що поєднує покрокову стратегію та тактику в реальному часі, є основним елементом серії Total War. Гравець грає роль як лідера клану, так і генерала, чергуючись між кампанією, де гравець керує землею та арміями по черзі, і битвами, де гравець бере під свій контроль армію на полі бою в режимі реального часу.

У кампанії гравець повинен стежити за розвитком поселень, військовим виробництвом, економічним зростанням та технологічним прогресом відповідно. Армії та підрозділи організуються та переміщуються гравцем по стилізованій карті кампанії для ведення битв з іншими фракціями. На додаток до битв, гравець може брати участь у дипломатії, політичному маневруванні та використанні спеціальних агентів, щоб перемогти. Ніндзя та гейші також присутні у грі як убивці та шпигуни. Хоча релігія не така актуальна, як це було в Medieval II: Total War, гравець не може нехтувати нею. Більш активна взаємодія з європейськими іноземцями (наньбанськими торговцями), наприклад, для розширення торгівлі та придбання вогнепальної зброї, піддає клан звернення до християнства, що

серйозно посилює релігійні хвилювання у провінціях. Релігійні агенти, такі як ченці та священики, можуть бути використані для навернення ворожого населення.

Існує дев'ять основних кланів, що населяють провінції Японії, з яких вибирає гравець. Є й інші, зокрема "клан слуг Акамасу". Всі клани мають особливі переваги у певних областях, щоб забезпечити різноманітність стилю гри з кожним. Клани:

- Клан Тесокабе населяє провінцію Тоса і може наймати найкращих лучників і отримувати більше прибутку від сільського господарства.

- Клан Дата контролює Івате і може наймати переважаючих самураїв но-дачі, які загони також отримують бонус при зарядці.

- Клан Ходзе - великі будівельники замків та фахівці з облоги. Вони населяють провінції Ідзу та Сагамі.

- Клан Морі населяє провінцію Акі та має довгу історію військово-морської майстерності.

- Клан Ода - ефективні командири асигару та влаштувалися в провінції Оварі.

- Клан Сімадзу населяє провінцію Сацума і може наймати чудових озброєних катаною самураїв, їх генерали також лояльніші до свого клану.

- Клан Такеда править провінцією Кай і набирає переважну кавалерію.

- Клан Токугава спочатку населяє провінцію Мікава як васал клана Імагава і покладається на дипломатичні відносини та вербування найкращих воїнів-ніндзя та мецукє.

- Клан Уесугі контролює провінцію Ечіго і спеціалізується на буддизмі, що дозволяє їм наймати кращих ченців та ченців-воїнів, а також отримувати більший прибуток від торгівлі.

Є також три фракції, доступні як завантажуваний контент:

- Хаторі є провідною родиною в провінції Іга і наймають спеціалізованих ніндзя та воїнів-ніндзя з великим досвідом.

- Клан Ікко-Іккі - це "родина" релігійних повстанців, які контролюють провінції Етидзен і Кага і вербують Ронін та вищих ченців-воїнів.

- Клан Отомо контролює провінції Бунго та Бузен, вони починають з католицької віри і можуть набирати чудові підрозділи вогнепальної зброї, а також португальську елітну піхоту, відому як терсос.

У Shogun 2 додано можливість розблокувати риси характеру та спеціальні здібності для генералів та агентів у міру отримання ними досвіду.

У Total War: Shogun 2 лідери та генерали наділені індивідуальністю та глибиною ігрового процесу, з великим акцентом на рольові ігри. Генерали та агенти зображуються як "видатні" герої з унікальними характеристиками та потужними здібностями.

Гравець може покращувати та відкривати риси характеру та спеціальні здібності персонажів у міру того, як вони набираються досвіду. Проте гравець також може схильний брати участь у сімейній політиці всередині клану, щоб зберегти лояльність його членів. Бій у Shogun 2 включають великомасштабні зіткнення між арміями, які зустрічаються на карті кампанії і можуть проходити на суші або воді. Розробники заявляють, що вони приділяють особливу увагу переробці морських та облогових битв, що відповідають новому сеттингу.

На відміну від європейських замків та фортів, замки у феодальній Японії мали кілька ярусів, і, таким чином, облогові битви у грі приділяли менше уваги захисту стін, а більше – бійкам у дворі та тактичному маневруванню. Крім того, гравці боротимуться у морських битвах з унікальними японськими кораблями, що нагадують "плавучі замки", і враховуватимуть ближній бій на кораблях, стрілянину з лука, прибережну місцевість та інші фактори.

Як і в останніх частинах, Empire та Napoleon: Total War, погодні та кліматичні умови впливають на битви. Наприклад, туман сильно знижує видимість, тоді як сильний дощ знижує ефективність ракетних військ, таких як лучники або артилеристи, що вимагає від гравців адаптації своїх стратегій. Також,



як і у випадку з Наполеоном, армії, що стоять у ворожих провінціях у зимовий сезон, або флоти, віддалені від узбережжя, страждають від виснаження.

Shogun 2 відомий унікальною подією для серії, відомою як Realm Divide, в якій один за одним усі клани, що керуються комп'ютером, оголошують війну гравцеві (або гравцям в режимі спільної кампанії) і вступають в союз один з одним, хоча союзники гравця, як правило, роблять це пізніше. Розділ королівства запускається, коли гравець захоплює достатню кількість територій чи захоплює Кіото. Популярність клану показує, наскільки гравець близький до запуску realm divide.

У Shogun 2 представлені розраховані на багато користувачів битви за участю до 8 гравців, а також розраховані на багато користувачів кампанії, що включають змагальну або спільну гру за участю 2 гравців. У розрахованій на багато користувачів кампанії гравці можуть бути згруповані в різні клани, так що для кожного клану один гравець бере на себе роль лідера клану, а інші приймають командування арміями.

Лідер клану має можливість спрямовувати інших гравців і призначати нагороди на основі лояльності та продуктивності, вводячи політику клану в розраховану на багато користувачів гру. Коли армія гравця вторгається на ворожу територію або піддається нападу ворожих армій, онлайн-сваха знаходить відповідного супротивника і ініціює битву. Коли гравець перемагає ворожі армії та завойовує території, гравець отримує очки та інші бонуси для клану. Крім того, система досягнень призначена для надання гравцям унікальних здібностей та косметичних покращень.

Таким чином, можна виділити основні переваги цієї частини:

1) «У цій частині все взаємозалежне, і гра ніде не здається переважаною». Така думка висловлювала абсолютну більшість гравців, а отже, баланс у цій частині настільки високий, наскільки це в принципі можливо.

2) Зручний і зрозумілий інтерфейс. Безперечно, це гідна перевага для стратегії. Зазвичай саме цей жанр навантажений незліченною кількістю кнопок,

ніж інші. А іноді такі нововведення взагалі заважають ігровому процесу. Таким чином, ця гра може підійти навіть новачкові в жанрі стратегії.

3) Гарна графіка. Більшості гравців графіка видається «актуальною, свіжою, молодою». Навіть незважаючи на те, що ця частина вийшла давно, вона досі чіпляє своєю зовнішньою складовою.

4) Гра не глобальна, проте це скоріше її перевага. У цій грі розробники приділили час не масштабним битвам, а опрацюванню деталей, що сприяло теплій реакції спільноти.

5) «Вузкий фокус за часом і по території робить Shogun 2 дуже багатим, з чудовою кампанією в один із періодів серії, що найбільш запам'ятовуються»

Так багато гравців відзначили цікавий сюжет та безліч цікавих механік взаємодій.

6) Такий формат усуває багато традиційних проблем Total War: II навчився використовувати кораблі, не гірше за гравця і агресивно поводить на більш високих рівнях складності і в той же час клани під керівництвом II, тепер не будуть дисбалансними.

7) Також люди знаходять хаос і переполох після оголошення війни гравцеві всіма кланами досить складною, але кумедною і по-справжньому цікавою подією.

З недоліків було знайдено подібні коментарі:

- «II іноді буває дурний»
- Є історичні неточності історії Японії.
- Мало яскравих і відомих «голлівудських» персонажів

Таким чином, гра отримала більше похвали, ніж критики. І якщо враховувати масштаб цієї частини, то можна сказати, що це найуспішніша з усієї серії Total War.

### 2.3.2 *Frozen Synapse 2*

Frozen Synapse 2 – покрокова тактична відеогра, розроблена Mode 7 Games. Це продовження відеоігри 2011 року Frozen Synapse.

Mode 7 Games is a United Kingdom-based video game developer and publisher. Mode 7 був заснований приблизно в 2005 році Полом Кілдаффом-Тейлором і Іаном Гардінгемом. Вони витратили два роки, щоб розробити Determinance, багатокористувацьку гру на мечях. За словами Кілдаффа-Тейлора, гра стала для них комерційним розчаруванням, але їхні можливості привернули контрактну роботу з такими клієнтами, як ITV, BBC і Novint Technologies. Їхня робота з Novint включала гру, яка викликала розбіжності – одні вважали її чудовою, а інші були збентежені. Це привело пару до мети створити гру, яка була б хорошою незалежно від того, скільки часу це займе.

Гра була анонсована у лютому 2016 року для Linux, MacOS та Windows. Спочатку розробники планували випустити гру в кінці 2016 року, але в результаті гра вийшла 13 вересня 2018 тільки для Windows.

Ігровий процес ділиться на одиночну гру та мультиплеер.

Режими одиночної гри

1) Міська гра

Це основний сюжетний режим. Карта міста і будівлі генеруються випадковим чином, і для сюжетних цілей деякі будівлі включають ті самі функції в кожному проходженні, з єдиною різницею в розміщенні юнітів і додаткових реквізитів, таких як дерева. Гравець може укладати спілки з іншими фракціями та укладати з ними контракти. Наприклад, деякі фракції можуть попросити вас допомогти їм у битві або доставити щось. Гравець може купувати додаткових юнітів на ринку найманців і об'єднувати їх в загони, що настроюються. Основна мета історії – зупинити терористичну організацію Sonata.

2) Розділ "Навчальні посібники" навчає основ гри і включає відеоролики по кожному аспекту ігрової механіки.

Цей ігровий режим включає тільки сюжетні місії і не фокусується на фракціях або управлінні ресурсами на стратегічному рівні.

3) Перестрілки – це матчі проти комп'ютера на випадково згенерованих картах із міською тематикою.

Багатокористувацькі режими.

1) Знищення: рандомізований матч, у якому гравці набирають очки, знищуючи загони іншої команди. Гра закінчується, коли одна команда повністю переможена або коли досягнуто ліміту раунду. Генерація включає однакову настройку юнітів для обох команд.

2) Знешкодження бомби: Новинка Frozen Synapse 2, в цьому ігровому режимі одна команда має підрозділ, що несе бомбу. Команда бомбардувальників повинна встановити бомбу в одному з двох місць до закінчення часу (за замовчуванням 16 ходів). Після того, як бомба закладена, команда знешкодження намагається знешкодити бомбу протягом 5 ходів, тоді як команда підривників захищає її. Типова генерація сценарію дає команді бомбардування більше одиниць підтримки, тоді як команді знешкодження надається більше наступальних одиниць.

3) Атака: Кожна з двох команд вибирає вертикально орієнтовану зону на карті, яку атакуючий повинен перетнути за 6 ходів і утримувати протягом 3 секунд. Гравець, який вибрав найдалшу зону, стає атакуючою командою, тоді як інший гравець захищає.

4) Суперечка: Дві команди змагаються, щоб зібрати посилки та доставити їх у зону висадки у кутку карти. Перемагає команда, що залишилася останньою, або команда, яка збрала найбільшу кількість пакетів, коли таймер ходу закінчився.

5) Заручник: У цьому режимі в одного гравця є цивільні особи, розташовані в середині карти, що стоять у зеленій та жовтій зонах. У зеленій зоні гравець може керувати своїми цивільними особами, тоді як у жовтій вони стають керованими комп'ютером. Щоб перемогти, гравець, який контролює цивільну особу, повинен привести хоча б одного громадянського в безпечну зону до кінця 12-го ходу, у той час як гравець, який захопив заручників, повинен убити всіх цивільних осіб, щоб перемогти, але не може атакувати цивільних осіб, що стоять у зеленій та жовтій зонах.

6) Безпека: Кожен гравець вибирає зону, яку, на його думку, може захистити. Гравець, який вибрав велику зону, стає командою, що захищається, і повинен захищати свою зону протягом 6 ходів. Захист може вибрати, де починаються його юніти у своїй зоні, в той час як атакуючий починає з далекого краю карти. Якщо захисник утримує свою зону протягом 6 ходів або перемагає всіх юнітів, захисник виграє. Якщо атакуючий або утримує зону захисника протягом 3 секунд, або перемагає всіх захисників, атакуючий виграє.

7) Завантаження: У кожного гравця є цивільний хакер і намагається усунути хакера противника. У кожній грі генеруються однакові юніти (хакер, 2 штурмові гвинтівки та 2 щити) для обох команд.

#### 8) Збіги в один хід

У цій категорії випробувань гравець має лише один хід (10 секунд у реальному часі), щоб виконати завдання. Команда противника завжди управляється комп'ютером за допомогою заздалегідь розробленого плану (створеного користувачем, який створив карту), та інші гравці можуть змагатися, щоб заробити найбільшу кількість очок за цю карту за один хід.

Об'єктивний вибір полягає в наступному:

8.1) Легкий видобуток: атакуюча команда має знищити мирних жителів, які не можуть рухатися. Бонусні очки нараховуються за перемогу над ворожими юнітами та захист власних юнітів гравця.

8.2) Безпека: У команди, що захищається, є один V.I.P. юніт, якого вони повинні захищати від команди, що атакує. Бонусні окуляри нараховуються за кожного переможеного ворожого юніта.

8.3) Знищити: дві команди просто змагаються, щоб перемогти якнайбільше противників.

#### 9) Додаткові параметри для збігів

9.1) Темний режим: У цьому режимі ворожі юніти видно лише тоді, коли власні юніти гравця знаходяться у прямій видимості.

9.2) Світлий режим: На відміну від темного режиму, ворожі юніти завжди видно у світлому режимі.

9.3) Тимчасові повороти: Гравцям може бути дано обмеження часу на те, як довго вони повинні завершити свій хід. Існує також можливість використовувати шаховий годинник.

9.4) Звичайне покоління: в налаштуваннях цього покоління гра створює карту з урахуванням розрахованої на багато користувачів гри, яка підходить для змагальної гри.

9.5) Генерація міста: цей тип генерації створює велику, докладну карту, подібну до карти в режимі міста.

9.1) Стара школа: Цей тип генерації створює карту так само, як і вихідний заморожений синапс.

Сильні сторони цієї гри:

1) «Frozen Synapse 2 справді стала грою більш амбітною, різноманітною та багатоплановою».

Так загалом говорять критики про нову частину гри. Здебільшого це стосується додавання нових карт, створення «більш масштабного» ігрового процесу, тобто. розширеному вибору тактики, стилю гри.

2) Поява повноцінного глобального стратегічного режиму і багато інших режимів, які вплинули на сприйняття гри користувачами.

3) Гравці відзначають цікавий, «захоплюючий» сюжет цієї частини.

4) Поява добре виконаного, зручного мультиплеєра дала гравцям можливість без перешкод грати зі своїми друзями.

5) Чудовий музичний супровід. За словами деяких критиків, воно не просто прекрасне, воно «геніальне».

З недоліків: гравці скаржилися на графіку, яка все ще відлякувала, а також на неправильний політ камери в деяких локаціях.

Реакція гравців на другу частину була більш позитивною, ніж негативною. І що дивно, то це те, що критики оцінили гру вище, ніж звичайні гравці.

## Висновки по розділу 2

Проаналізовано характеристики, які цінували гравці у тій чи іншій грі. Фінальний проект розроблятиметься саме в рамках тих характеристик, які будуть оптимальні для даного випадку. Якщо одна характеристика повторюється у двох або більше ігор – вона вказується лише в одній із них, без повторень.

У першому пункті Розділу 2 було проаналізовано дві гри жанру покрокова стратегія: Civilization V та Heroes of Might and Magic III.

Civilization V сподобалася гравцям через невибагливість, низьку ціну та звичний інтерфейс. Оскільки забезпечити «звичність» у новій, нікому незнайомій грі буде складно, треба дотримуватися перших двох характеристик.

Heroes of Might and Magic III сподобалася фанатам через гарний баланс, варіативність та барвисту графіку. Всі ці переваги було б непогано задіяти і при розробці додатка.

У другому пункті Розділу 2 було проаналізовано дві гри жанру покрокова стратегія: StarCraft II та Age of Empires II.

StarCraft II сподобався гравцям через високу динаміку ігрового процесу, різноманітності, сольнішої, некоманднішої гри. Сучасні гравці досить швидко перемикаються між різними іграми, хочуть різноманітності, тому можливості утримати гравців у грі на безперервних 3-4 години досить важко. Тому фінальний прототип обов'язково буде динамічним, різноманітним. А також наявність елемента «соло-ігри» не суперечитиме наміру створити мультиплеєр, адже в гру будуть додані Боти, які діятимуть приблизно, як і гравець, з якими так само можна боротися та отримувати нагороди.

Age of Empires II показав себе не дуже добре у відгуках, хоч і є однією з найпопулярніших ігор серії. Гравці висловлювали бажання грати з різними людьми (не лише з друзями), здобути хорошу локалізацію, побачити багато статистики про гравців. Перший пункт однозначно буде реалізовано, адже гра будуватиметься на принципі кімнат. Тобто кожен гравець при створенні «кімнати» вказує скільки гравців йому потрібно для початку матчу, скільки ботів,

час проведення тощо. Два інші пункти скрутні, адже в такому разі необхідно буде наймати перекладачів і створювати систему «друзів». Для прототипу це зайве. Але в теорії цілком реалізовано.

У третьому пункті Розділу 2 було проаналізовано дві гри жанру покрокова стратегія (який був інтерпретований як жанр покрокової стратегії у рельєфному часі): Total War: Shogun 2 та Frozen Synapse 2.

Shogun 2 запам'ятався гравцям неглобальністю (а тому й високою деталізацією) того, що відбувається, невеликою кількістю контенту, зручним інтерфейсом. Якщо перші два реалізувати відносно легко, то щодо зручного інтерфейсу буде важко визначитися – всі люди вважають зручними різні речі.

Frozen Synapse 2 відзначився відмінним музичним супроводом, захоплюючим сюжетом та зручним мультиплеєром. Музика та мультиплеєр обов'язково будуть додані до гри. Щодо сюжету – не думаю, що він має бути принаймні у прототипі. Це все-таки більша абстрактна гра для кількох людей, а не проходження «історичної лінії».

Таким чином, було вирішено, що сам ігровий процес складатиметься із захоплення чужих/порожніх територій гравцями та захист своїх. Території будуть являти собою шестикутники з прикрасами (які можна придбати в магазині за валюту). Валюта буде видана після закінчення матчу відповідно до підсумкового економічного рахунку гравця та його місця у рейтингу.

Сама структура сервера складатиметься з кімнат, на яких можуть грати як маленькі компанії людей, так і великі з Ботами (II) або без них. Буде можливість і соло-гри – необхідно створити кімнату на одну особу та додати від 1 до 10 ботів.



## РОЗДІЛ 3 СТВОРЕННЯ ПРОТОТИПУ СТРАТЕГІЇ

Після аналізу стратегій відбувається формування схеми роботи, алгоритм.

Остаточний порядок розробки програми був наступним:

- Створення сайту
- Створення бази даних та підключення до сайту
- Розробка логіки самої гри
- Тестування гри розробниками для налагодження помилок
- Створення лаучера для автоматичних оновлень програми

### 3.1 Сайт

Створення сайту розпочалося з планування. Послідовність створення сайту та чітке опрацювання етапів – невід'ємна частина проекту. Створення веб-сайту містить у собі не тільки розробку дизайну та програмування, а й детальний аналіз проекту, співпрацю із замовником та пошук рішень для досягнення поставлених цілей проекту. Основні етапи створення веб-сайту:

- Постановка цілей та завдань сайту;
- Створення, опрацювання технічного завдання (ТЗ) на розробку сайту;
- Прототипування;
- Створення макету дизайну сайту;
- Верстка;
- Програмування;
- Наповнення контентом;
- Тестування;

Було вирішено, що мета – це створення сайту з реєстрацією в грі.

Завданнями стало:

- виконання етапів, прописаних вище
- ухвалення ряду рішень з приводу дизайну
- визначення фінальної кількості сторінок
- визначення кількості передбачуваного контенту на сторінках

- організація переходу на сторінки

Наступним кроком стало складання ТЗ. Для цього було розроблено план дій для користувача: (див. рис. 3.1)



Рис. 3.1. Очікувані дії нового користувача під час заходу на сайт

Обов'язковими сторінками є: Головна, Реєстрація, Завантажити гру. Проте, можливо, гравець зацікавиться подробицями про гру, захоче дізнатися про зміни в оновленні, захоче обговорити новини з іншими. Це важливо для формування ком'юніті гри або, іншими словами, ігрової спільноти.

Таким чином, було вирішено, що сайт складатиметься з 4 сторінок: Головна, Новини, Реєстрація, Завантаження. Також було створено структуру сайту (див. рис. 3.2).



Рис. 3.2. Структура сайту.

Далі було вирішено які технічні функції виконуватиме сайт.

Різні сайти надають відвідувачам різну функціональність, а отже, різні технології повинні використовуватися в тому чи іншому випадку. Детально складений план, створений на основі даних, отриманих на цьому етапі, може запобігти витратам додаткових ресурсів на вирішення непередбачених труднощів, таких як зміна дизайну або додавання функціоналу, не передбаченого спочатку.

Для створення сайту було виділено три основні варіанти:

- 1) Створення сайту за допомогою спеціальних платформ (CMS)
- 2) Створення сайту за допомогою фреймворків
- 3) Створення сайту з нуля

CMS (Content Management System) — це система управління, двигун, платформа або конструктор, який дозволяє керувати вмістом сайту.

Згідно з аналітичними дослідженнями, у березні 2020 року лідируючі позиції в інтернеті серед CMS зайняли WordPress, 1С-Бітрікс, Joomla!, OpenCart та Drupal. Розберемося, які особливості у кожної системи, та розглянемо, для яких проектів вони підійдуть.

1) WordPress — безкоштовна та найпоширеніша система управління контентом. На цьому движку працюють понад 35% веб-сайтів у всьому світі. WordPress підходить для будь-якого типу сайтів: від блогів та односторінників до сторінок новин та інтернет-магазинів. Для цього на движку є безліч вбудованих функцій.

2) CMS 1С- Бітрікс – рішення переважно для великих сайтів. Двигун використовують для майданчиків, де потрібний широкий набір функцій: інтернет-магазинів, корпоративних порталів. Для невеликих сайтів, наприклад, лендингів або візиток 1С-Бітрікс використовувати не вигідно через високу вартість продукту. CMS недостатньо зручна для початківців розробників та користувачів.

3) Як і WordPress, Joomla! підходить для всіх типів веб-сайтів. При цьому немає значення масштаб: система однаково оптимальна для невеликих сторінок і великих майданчиків. Встановлення та налаштування системи не викликають

проблем навіть у новачків, тому двигун вважається гідною альтернативою ВордПрес.

4) OpenCart — безкоштовна CMS, призначена переважно для онлайн-магазинів, адже система спочатку розроблена для створення інтернет-магазинів. Для сайтів іншого напрямку функціоналу недостатньо. Для проектів у сфері e-commerce це одне з найкращих безкоштовних рішень.

5) Drupal — безкоштовна система керування контентом з відкритим вихідним кодом. Переважно двигун застосовують для великих проектів, які потребують високої продуктивності. Для невеликих блогів оптимальнішими будуть альтернативні безкоштовні системи. Drupal має не дуже «дружній» інтерфейс.

Однак було вирішено, що CMS не надає достатньо контролю за базою даних і, як з'ясувалося, це досить небезпечне рішення. Якщо буде вирішено додати плагіни, це ставить під загрозу дані гравців.

Веб-фреймворк – це каркас для написання веб-застосунків. Він визначає структуру, задає правила та надає необхідний набір інструментів для розробки. Бувають:

- 1) Фронтенд-фреймворки
- 2) Бекенд-фреймворки
- 3) Фуллстек-фреймворки

Фреймворки підходять для створення як великих проектів, так і простих сайтів та програм, які планується розвивати в майбутньому. Вони дозволяють правильно побудувати бізнес-логіку. В основному фреймворки використовуються для створення калькуляторів, інтернет-магазинів з нестандартним функціоналом, власних CRM, які не можна створити на CMS, десктопних та мобільних програмах. На відміну від CMS, frameworks — це низькорівневе рішення, яке має більшу продуктивність і гнучкість. Оскільки бажаний сайт досить простий як за структурою, так і за функціоналом, було вирішено не використовувати фреймворки, а робити сайт з нуля.

Етап прототипування був поєднаний з етапом створення макета дизайну сайту, адже по суті, враховуючи, що сайт невеликий, це те саме.

Таким чином, були створені макети кожної з чотирьох сторінок: Головна (див. рис. 3.3), Завантаження (див. рис. 3.4), Реєстрація (див. рис. 3.5), Новини (див. рис. 3.6).

Наступним етапом стала верстка сайту. У всіх 4 сторінок були спільні риси: вони мали однакове тло, стилі та заголовки (див. рис. 3.7) і меню вгорі сторінки (див. рис. 3.8).

Однак заповнення всіх сторінок було різним. Для прикладу в рамках цього розділу наведено приклад сторінки з реєстрацією (див. рис. 3.9).

З повною версією HTML-коду можна ознайомитись у Додатку А.

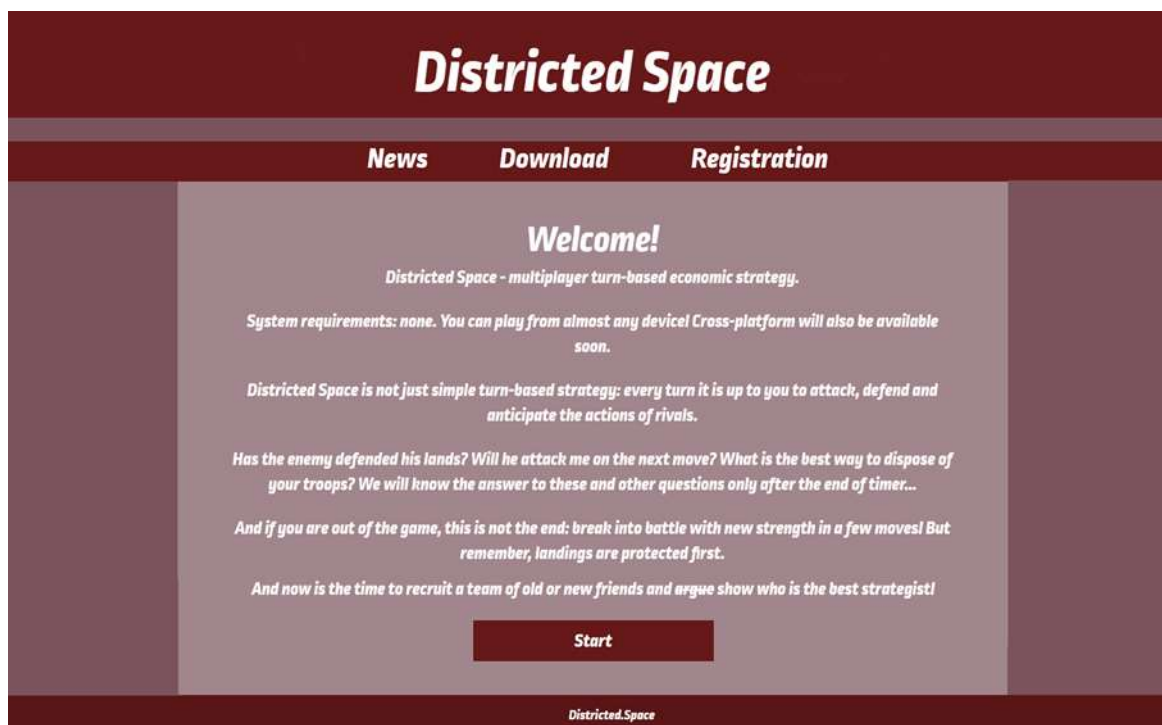


Рис. 3.3. Прототип Головна сторінка.



Рис. 3.4. Прототип Завантаження гри.



Рис. 3.5. Прототип сторінки Реєстрації.



Рис. 3.6. Прототип сторінки Новини.

```

<head><meta charset="utf-8">
  <title>Districted Space</title>
  <!-------
  <link href="favicon.ico" rel="icon" /><!-- 32x32 -->
  <link href="images/icon.svg" rel="icon" type="image/svg+xml" />
  <link href="images/apple.png" rel="apple-touch-icon" /><!-- 180x180 --><!-------
  <link href="style.css" rel="stylesheet" /><!-------
</head>
<!-------
<body><!-------
<div class="background"></div>
<div class="HeaderPanel">
  <h1 class="Header">
    
    
    
    
    <a class="ButtonMain" href="index">Districted Space </a>
    
    
    
    
  </h1>
</div>

```

Рис. 3.7. Підключення іконок, зображень та стилів.

```
<div class="PanelButtons">
    <a href = "news" class = "ButtonHeader" >News</a>
    <label class = "ButtonSeparator"> | </label>
    <a href = "download" class = "ButtonHeader" >Download</a>
    <label class = "ButtonSeparator"> | </label>
    <a href = "registration" class = "ButtonHeader" >Registration</a>
</div>
```

Рис. 3.8. Меню зверху сторінки

```
<div class="ContentBody">
<----->
<div class = "Registration">
    <div class = "Form"><br><br><br></div>
    <form class="js-form" action="..Php/functions.php" method="post">
    <label class="Label">Registration</label> <br> <br>
    <label class="RegistrationLabel">Enter your nickname:</label> <br>
    <input type="text" class="RegistrationInput js-input nickname" name="nickname" id="nickname" onclick="nicknameInputRestart();" <br>
    <label class="RegistrationLabel">Enter your password:</label><br>
    <input type="password" class="RegistrationInput js-input password" name="password" id="password" onclick="passwordInputRestart();" <br>
    <label class="RegistrationLabel">Repeat your password:</label><br>
    <input type="password" class="RegistrationInput js-input passwordconfirm" name="passwordconfirm" id="passwordconfirm" onclick="passwordconfirmInputRestart();" <br>
    <label class="RegistrationLabelformmessage" id="errormessage"></label><br><br> <br>
    <input type="submit" value="Register" class="RegistrationButton"></button>
    </form>
    <script src="..js/Validation.js"></script>
    <div class = "Form"><br><br><br></div>
</div>
```

Рис. 3.9. Форма реєстрації.

Наступним кроком була розробка функціоналу чи програмування. Для виконання цього завдання була створена проста локальна база даних із таблицею Users.

*Структура таблиці Users:*

- 1) PlayerID (integer + auto increment) (первинний ключ)
- 2) PlayerNickName (varchar)
- 3) PlayerPassword (varchar)

Реалізація етапу програмування складається з 2 частин: написання коду на JavaScript та PHP.

Загальна структура файлу із кодом JavaScript представлена на рис. 3.10.



Далі були детальніше представлені деякі реалізації валідації нового користувача:

- перевірка імені на валідність символів (див. рис. 3.11)
- перевірка пароля на валідність символів (див. рис. 3.12)
- перевірка імені та пароля на кількість символів (див. рис. 3.13)

Якщо користувач пройшов всі перевірки, запускається код РНР (див. рис. 3.14), який виконує реєстрацію користувача.

```
let form = document.querySelector('.js-form'),
    formInputs = document.querySelectorAll('.js-input'),

    inputPlayerNickName = document.querySelector('.nickname'),
    inputPlayerPassword = document.querySelector('.password'),
    inputPlayerPasswordConfirm = document.querySelector('.passwordconfirm');

var r = 0;
var arr;

function validationNickName(nickname) ...
}

function validationPassword(password) ...
}

function validationPasswordConfirm() ...
}

form.onsubmit = function() ...
}

function registration() ...
}

////////////////////////////////////

function errorMessege(textMessege, colorError) ...
}

////////////////////////////////////

function nicknameInputRestart() ...
}

function passwordInputRestart() ...
}

function passwordconfirmInputRestart() ...
}
```

Рис. 3.10. Загальний код JavaScript

```

function validationNickName(nickname)
{

    for (var i = 0; i < nickname.value.length; i++)
    {
        arr = nickname.value[i];

        if(/^[a-zA-Z0-9_-]/g.test(arr) === false)
        {
            r++;
        }
    }

    if (r === 0)
    {
        r = 0;
        return true;
    }

    else
    {
        r = 0;
        return false;
    }
}

```

Рис. 3.11. Перевірка імені на валідність символів.

```

function validationPassword(password)
{

    for (var i = 0; i < password.value.length; i++)
    {
        arr = password.value[i];

        if(/^[a-zA-Z0-9_@#%]/g.test(arr) === false)
        {
            r++;
        }
    }

    if (r === 0)
    {
        r = 0;
        return true;
    }

    else
    {
        r = 0;
        return false;
    }
}
}

```

Рис. 3.12. Перевірка пароля на валідність символів.

```

if(nickname.value.length < 5 || password.value.length > 20)
{
    var text = "Your nickname must be from 5 to 20 symbols";
    var color = "#8B0000";
    errorMessege(text, color);

    return false;
}
else
{
    var isValidateNickName = true;
}

if(password.value.length < 6 || password.value.length > 25)
{
    var text = "Your password must be from 6 to 25 symbols";
    var color = "#8B0000";
    errorMessege(text, color);

    return false;
}
else
{
    var isValidatePassword = true;
}

if(isValidateNickName && isValidatePassword)
{
    registration();
}

```

Рис. 3.13. Перевірка імені та пароля на необхідну кількість символів.

```

function registration()
{
    var text = "You have successfully registered!";
    var color = "green";
    errorMessege(text, color);
    window.open("../Php/Functions.php", "_self");
}

```

Рис. 3.14. Запуск PHP коду реєстрації (якщо дотримано всіх умов).

Далі, якщо все пройшло успішно, починається запис імені та пароля з полів у змінні (див. рис. 3.15). Щоб зі змінних дані записалися до бази, необхідно підключитися до бази даних (див. рис. 3.16). З метою безпеки паролі та логіни були розмиті. Після всіх цих дій ми додаємо дані про користувача зі змінних до бази даних (див. рис. 3.17).

```
$playerNickName = filter_var(trim($_POST['nickname']),FILTER_SANITIZE_STRING);  
$playerPassword = filter_var(trim($_POST['password']),FILTER_SANITIZE_STRING);
```

Рис. 3.15. Запис імені та пароля у змінні з полів.

```
$dbhost = 'localhost';  
$dbuser = 'root';  
$dbpassword = 'root';  
$dbbasename = 'users';  
  
$database = new mysqli($dbhost, $dbuser, $dbpassword, $dbbasename);  
$database->query("SET NAMES 'utf8'");  
////////////////////////////////////
```

Рис. 3.16. Підключення до бази даних (логін та пароль приховані).

```
////////////////////////////////////  
// Питання База данних  
////////////////////////////////////  
$sql = "SELECT * FROM Users WHERE PlayerNickName = '$playerNickName'";  
  
$result = $database -> query($sql);  
  
$row = $result->fetch_assoc();  
  
if($row)  
{  
    echo "User already registered";  
}  
else  
{  
    $database->query("INSERT INTO `Users` (`PlayerNickName`, `PlayerPassword`) VALUES ('$playerNickName', '$playerPassword') ");  
}  
  
////////////////////////////////////  
// Отключаемся от базы  
////////////////////////////////////  
$database->close();  
  
////////////////////////////////////  
// Редирект на страницу  
////////////////////////////////////  
header('Location: /download');  
  
?>
```

Рис. 3.17. Запис користувача до бази.

Таким чином, новий користувач був зареєстрований і був переадресований на сторінку із завантаженням гри. На даний момент сайт виглядає так:

- Головна (див. рис. 3.18)
- Завантаження гри (див. рис. 3.19)
- Реєстрація (див. рис. 3.20)

Сторінка з новинами поки неактивна, бо немає оновлень, немає гравців. Однак, по ходу розвитку гри і на ній з'явиться контент.



Рис. 3.18. Фінальний вигляд Головної сторінки.



Рис 3.19. Фінальний вигляд сторінки Завантаження гри.



Рис 3.19. Фінальний вигляд сторінки Реєстрації.

### 3.2 База даних та сервер

База даних - сукупність даних, що зберігаються відповідно до схеми даних, маніпулювання якими виконують відповідно до правил засобів моделювання даних.

У цьому проекті вона необхідна для запису, обробки, зберігання та подальшого використання даних про гравця у процесі гри (наприклад, виведення імені Гравця1 для інших користувачів ігрової сесії).

Так як гра задумана як розрахована на багато користувачів, то однієї невеликої локальної бази даних недостатньо. Більше того, вона не завжди буде доступною. Як тільки комп'ютер з базою даних переходить режим сну або завершує роботу, БД перестає бути доступною для користувачів, а отже, виникнуть проблеми з реєстрацією нових користувачів, з входом у додаток вже зареєстрованих користувачів та з записом результатів гри у тих, хто в момент відключення грав на сервері.

Так як звичайний ПК не може бути включений постійно, було вирішено організувати віддалену базу даних та віддалений сервер для гри.

На відміну від локальної бази даних, нова база даних складалася з 8 стовпців:

- PlayerID (унікальний ідентифікатор гравця)
- PlayerNickName (ігрове ім'я)
- PlayerPassword (пароль)
- PlayerMoney (кількість ігрової валюти, початкове значення - 200)
- PlayerSkinsMain (доступні основи)
- PlayerSkinsImages (доступні зображення)
- PlayerSkinsBorders (доступні рамки)
- PlayerRegistrationData (дата реєстрації гравця)

Про перші три було написано у пункті 3.1. Після них йде PlayerMoney - показник, який відповідає за кількість валюти гравця. За валюту можна купити будь-яку із трьох модифікацій своєї території:

1) Main, тобто "основа" території. Це основний колір, найчастіше принт, позначений (1) на рис. 3.20.

2) Image, тобто зображення усередині території. Це будь-яка картинка, не шестикутник. Наноситься поверх Main'a. Позначений (2) на рис. 3.20.

3) Border, тобто межа території. Найчастіше представлена як шестикутне обведення для Main'a. Позначений (3) на рис. 3.20.

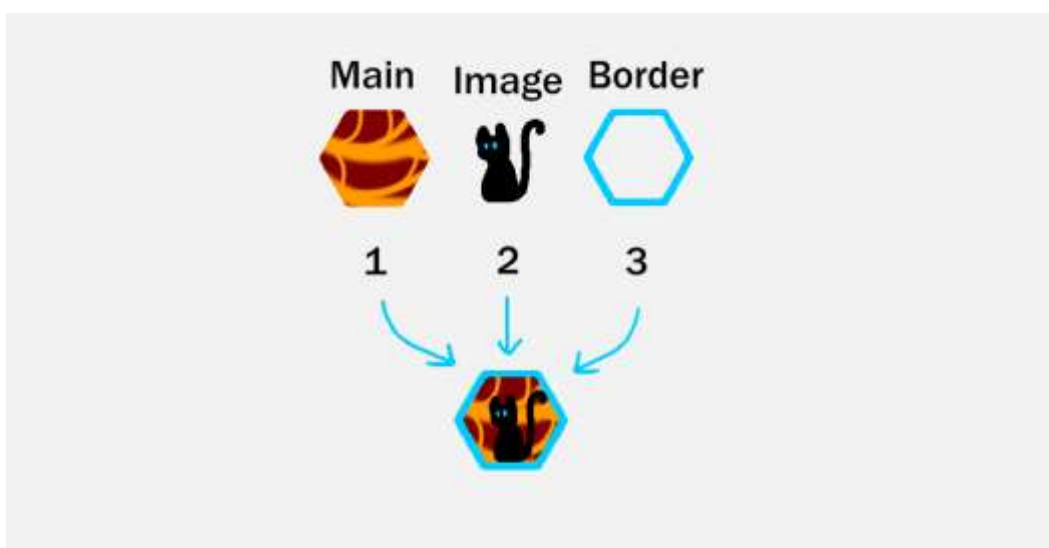


Рис. 3.20. Main, Image, Border.



Таким чином, при реєстрації гравця повинен вийде новий рядок у базі даних. Приклад такого запису на рис. 3.21.

PlayerID	PlayerNickName	PlayerPassword	PlayerMoney	PlayerSkinsMain	PlayerSkinsImages	PlayerSkinsBorders	PlayerRegistrationData
4	AnneLM	123455	215	0	-1	-1*5	2022-07-01

Рис. 3.21. Приклад запису гравця.

А загалом процес реєстрації виглядає так: рис. 3.22. Гравець, заходячи на сайт, відправляє свої дані (Логін та Пароль двічі) сайту, який, своєю чергою, відправляє їх у базу даних. У БД отримані дані аналізуються на валідність і користувачеві надходить або сповіщення про успішну реєстрацію, або помилка із зазначенням проблеми (або неправильна кількість символів, або небажані символи).

Сервер являє собою віддалений ПК, на якому запущено один із двох типів збірок гри – Server. Друге складання, Client, запускається на пристрої користувача.

На сервері відбувається відлік таймерів, калькуляція всіх дій вчинених за хід та все, що відбувається в ігрових кімнатах. При правильному підключенні (рис. 3.23) Гравець заходить у гру, вводить свої дані (Логін та Пароль) для входу на сервер.

Якщо у користувача є Інтернет-з'єднання та всі компоненти (Сервер, Сайт, БД) працюють коректно, то дані, введені гравцем, відправляються на Сайт.

Далі дані потрапляють у БД і зіставляються з усіма рядками таблиці Users. Безпосередньо до БД підключитися не можна (оскільки вона знаходиться на одному ПК з Сайтом), тому, якщо запитуються Логін та Пароль існують, то всі наявні дані про гравця (а саме PlayerID, PlayerMoney, PlayerSkinsMain, PlayerSkinsImages, PlayerSkinsBorders, PlayerRegistrationData) відправляються назад у Клієнт гри через Сайт.



І останнім кроком стає підключення гравця (з усіма даними, отриманими з БД) до Сервера. Якщо Логін або Пароль невірні, то БД повертає до Клієнта помилку (див. рис. 3.24), а подальшого підключення до Сервера немає.

### 3.3 Розробка гри

Для серверної частини була використана версія гри, яка виконує логіку ігрового сервера:

- Створення ігрових кімнат
- Створення картки
- Запуск гри
- Таймер
- Розрахунок ходу
- Битва загонів
- Переділ територій
- Закінчення гри

З метою економії оперативної пам'яті було вирішено запускати не окремий сервер на кожну гру, а створити «кімнати» всередині одного сервера (див. рис. 3.25). Таким чином, вийшло менш витратно і по бюджету, і по пам'яті, що виділяється.

```
public RoomGenerator CreateRoom(int playerId, string playerName, string roomName, int playerCount, int timerCycle, int moneyStart, int moveCountLimit, int botCount, string m

if (RoomManager.Singleton.IsServer)

    GameObject spawnedRoomObject = Instantiate(roomPrefabObject, transform.position, transform.rotation);
    spawnedRoomObject.GetComponent<RoomObject>().Spawn();

    _uniqueRoomId += 1;

    spawnedRoomObject.GetComponent<Room>()._m_RoomId.Value = _uniqueRoomId;
    spawnedRoomObject.GetComponent<Room>()._m_RoomName = roomName;
    spawnedRoomObject.GetComponent<Room>()._m_PlayerCountToStart = playerCount;
    spawnedRoomObject.GetComponent<Room>()._timerObject.GetComponent<Timer>()._m_TimerCycleTime = timerCycle;
    spawnedRoomObject.GetComponent<Room>()._m_PlayerStartMoney = moneyStart;
    spawnedRoomObject.GetComponent<Room>()._m_MoveCountLimit = moveCountLimit;
    spawnedRoomObject.GetComponent<Room>()._botListObject.GetComponent<List>()._m_BotCount = botCount;
    spawnedRoomObject.GetComponent<Room>()._m_MapFile = mapName;
    spawnedRoomObject.GetComponent<Room>()._m_IsActiveRoom = false;

    m_RoomList.Add(spawnedRoomObject);

    spawnedRoomObject.name = "Room " + _uniqueRoomId + "/" + roomName + "/";

    Debug.Log("Створена кімната з назвою " + roomName + ", кількістю гравців " + playerCount + ", розміром карти " + mapName + ", початковою грою " + moneyStart + ",

yield return new WaitForSeconds(1);
StartCoroutine(_serverObject.GetComponent<Server>().ServerJoinPlayerToRoom(playerID, playerName, _uniqueRoomId));

yield return null;
}
```

Рис. 3.25. Код створення ігрової кімнати.

При створенні кожної такої кімнати їй надається порядковий номер та певні параметри, що задаються гравцем:

- Ім'я кімнати
- Кількість гравців для початку гри
- Час циклу ходу
- Кількість стартових грошей
- Ліміт ходів
- Кількість ботів у грі
- Розмір карти

Також сервер під час створення нової гри генерує карту (ігрове поле) (див. рис. 3.26). Приклад однієї ітерації такої генерації подано на рис. 3.27. У карти також є параметри:

- Порядковий номер кімнати якої належить карті
- Розмір карти X
- Розмір карти по Y

```
Ссылка: 1
public void TerritorySpawnOnServer(int territoryRoomID, int sizeX, int sizeY)
{
    if (NetworkManager.Singleton.IsServer)
    {
        x = sizeX;
        y = sizeY;

        StartCoroutine("AsyncSpawnPentagonFirstIteration", territoryRoomID);
        StartCoroutine("AsyncSpawnPentagonSecondIteration", territoryRoomID);
        StartCoroutine("AsyncSpawnPentagonThirdIteration", territoryRoomID);
        StartCoroutine("AsyncSpawnPentagonFourthIteration", territoryRoomID);
    }
}
```

Рис. 3.26. Генерація карти.

```

class Iterator KeyspacePentagonFirstIteration(int territoryRoomID)
{
    for (float x1 = 0; x1 < ((x * 70) / 4); x1 += 70)
    {
        for (float y1 = 0; y1 < y * 2; y1 += 2)
        {
            _randomInt = Random.Range(1, 101);
            if (_randomInt < _randomity)
            {
                _spawnedFirstIterationTerritory = Instantiate(m_TerritoryPrefabObject, new Vector3(x1, y1, 0), Quaternion.identity);
                _spawnedFirstIterationTerritory.GetComponent<Transform>().SetParent(_worldSpaceObject);
                _spawnedFirstIterationTerritory.name = "Territory " + x1 + ", " + y1 + "1";
                _spawnedFirstIterationTerritory.GetComponent<Territory>().m_TerritoryRoomID = territoryRoomID;

                int randomIncomeValue = Random.Range(0, 35 + 1);

                if (randomIncomeValue == 0)
                else if (randomIncomeValue >= 1 && randomIncomeValue <= 3)
                else if (randomIncomeValue >= 4 && randomIncomeValue <= 7)
                else if (randomIncomeValue >= 8 && randomIncomeValue <= 12)
                else if (randomIncomeValue >= 13 && randomIncomeValue <= 18)
                else if (randomIncomeValue >= 19 && randomIncomeValue <= 25)
                else if (randomIncomeValue >= 26 && randomIncomeValue <= 30)

                _spawnedFirstIterationTerritory.GetComponent<Territory>().m_Text_TerritoryComm.text = _spawnedFirstIterationTerritory.GetComponent<Territory>().m_TerritoryRoomID;
                _spawnedFirstIterationTerritory.GetComponent<Territory>().m_Text_TerritoryScore.text = _spawnedFirstIterationTerritory.GetComponent<Territory>().m_TerritoryRoomID;

                int randomDestantValue = Random.Range(1, 10 + 1);

                if (randomDestantValue > 0)
                {
                    _spawnedFirstIterationTerritory.GetComponent<Territory>().m_IsTerritoryDesant = true;
                    _spawnedFirstIterationTerritory.GetComponent<Territory>().m_IsTerritoryDesant.SetActive(true);
                }
                else
                {
                    _spawnedFirstIterationTerritory.GetComponent<Territory>().m_IsTerritoryDesant = false;
                    _spawnedFirstIterationTerritory.GetComponent<Territory>().m_IsTerritoryDesant.SetActive(false);
                }

                _worldListObject.GetComponent<TerritoryList>().m_Territories.Add(_spawnedFirstIterationTerritory);
                _spawnedFirstIterationTerritory.GetComponent<Territory>().m_TerritoryID = _worldListObject.GetComponent<TerritoryList>().m_Territories.Count;

                yield return new WaitForSeconds(5.0f);
            }
        }
    }

    yield return isTerritorySpawnedFirstIteration == true;
}

```

Рис. 3.27. Варіант ітерації генерації карти.

Далі після того, як карта буде створена і до кімнати підключиться потрібна кількість гравців, сервер запускає гру цим кодом (див. рис. 3.26).

```

if (GetComponentInChildren<PlayerList>().m_PlayerList.Count >= m_PlayerCountToStart && isGameCanStart == 0)
{
    isGameCanStart = 1;

    if (isGameCanStart == 1)
    {
        timerStartedBeforGame = true;

        //Тільки для тих хто в кімнаті
        List<ulong> clientsIDList = new();
        foreach (GameObject player in GetComponentInChildren<PlayerList>().m_PlayerList)
        {
            clientsIDList.Add((ulong)player.GetComponent<Player>().m_ClientID);
        }

        ClientRpcParams clientRpcParams = new ClientRpcParams
        { Send = new ClientRpcSendParams { TargetClientIds = clientsIDList.ToArray() } };
        //////////////////////////////////////

        PlayerTimerWaitingGameClientRpc(clientRpcParams);
    }
}

```

Рис. 3.26. Запуск гри.

Як тільки гра розпочалася, на сервері запускається таймер та його значення передається всім гравцям у реальному часі.

Для реалізації таймера було використано наступний код (див. рис. 3.27).

```
Ссылка 1
private void TimerMove()
{
    if (NetworkManager.Singleton.IsServer && m_IsTimerActive && GetComponent<Room>().isGameStarted)
    {
        _timerTime.Value -= 1 * Time.deltaTime;
    }
}
```

Рис. 3.27. Реалізація таймера.

Його функція – кожну секунду віднімати 1 від поточного значення таймера до 0, після чого таймер вимикався (див. рис. 3.28), а сервер сигналізував, що хід завершено і починав розрахунок ходу (див. рис. 3.29).

```
Ссылка 1
private void TimerMoveEnd()
{
    if (NetworkManager.Singleton.IsServer && GetComponent<Room>().isGameStarted)
    {
        if (_timerTime.Value <= 0)
        {
            _timerTime.Value = 0;

            if (_timerTime.Value == 0 && !m_IsMoveEnd)
            {
                StartCoroutine(TimerMoveEndMessege());
            }

            m_IsMoveEnd = true;
            m_IsTimerActive = false;

            if (!m_IsTimerActive && m_IsMoveEnd && !GetComponent<Room>().m_IsGameOver)
            {
                StartCoroutine(CountingMove());
            }
        }
    }
}
```

Рис. 3.28. Завершення таймера

```

Ссылка: 1
private IEnumerator CountingMove ()
{
    if (NetworkManager.Singleton.IsServer)
    {
        if (m_IsMoveEnd)
        {
            CountingMoveMessege ();

            StartCoroutine (GetComponent<Room>().ProccesingMoveServer ());
        }

        m_IsMoveEnd = false;

        if (!m_IsMoveEnd && !m_IsTimerActive && !GetComponent<Room>().m_IsGameOver)
        {
            StartCoroutine (TimerStartingNextMoveMessege ());
            StartCoroutine (TimerStartNextMove ());
        }
    }

    yield return null;
}

```

Рис. 3.29. Розрахунок ходу

Розрахунок ходу досить складний процес як з точки зору обчислювальної потужності (навантаження на процесор), так і передачі даних (навантаження на інтернет). Так як треба обчислювати і передавати по мережі великі обсяги даних, то серверу потрібно пройтися всіма територіями і виконати наступні дії:

- Запустити битву загонів на території де більше 1 загону
- Зібрати імена власників кожної території
- Зібрати Main, Image, Border, власника кожної території
- Здійснити зміну власників територій
- Провести фарбування територій у колір власника
- Підрахунок доходу за хід з усіх територій для кожного гравця

Коли сервер закінчує розрахунки, він передає підсумкові дані всім гравцям. Як тільки кількість ігрових ходів добігає кінця, сервер оголосить закінчення гри і визначить переможця за наступним планом.:

- Відбувається підрахунок усіх набраних очок за гру (див. рис. 3.30)
- Якщо у кількох гравців однакова максимальна кількість очок, то переможців декілька
- Визначення підсумкового (-их) переможця (переможців) гри (див. рис. 3.31)

```

foreach (GameObject player in GetComponentInChildren<PlayerList>().m_PlayerList)
{
    if (player.GetComponent<Player>().m_MyPlayerResources.GetComponent<PlayerResources>().m_PlayerScore >= scoreToWin)
    {
        scoreToWin = player.GetComponent<Player>().m_MyPlayerResources.GetComponent<PlayerResources>().m_PlayerScore;
        isMultipleWinners = false;
    }
    else if (player.GetComponent<Player>().m_MyPlayerResources.GetComponent<PlayerResources>().m_PlayerScore == scoreToWin)
    {
        scoreToWin = player.GetComponent<Player>().m_MyPlayerResources.GetComponent<PlayerResources>().m_PlayerScore;
        isMultipleWinners = true;
    }
}

```

Рис. 3.30. Підрахунок очок та визначення кількості переможців

```

foreach (GameObject player in GetComponentInChildren<PlayerList>().m_PlayerList)
{
    if (!isMultipleWinners)
    {
        if (player.GetComponent<Player>().m_MyPlayerResources.GetComponent<PlayerResources>().m_PlayerScore == scoreToWin)
        {
            playerWinnerNickName = player.GetComponent<Player>().m_PlayerNickName;
        }
    }
    else
    {
        if (player.GetComponent<Player>().m_MyPlayerResources.GetComponent<PlayerResources>().m_PlayerScore == scoreToWin)
        {
            multipleWinnerNickName += player.GetComponent<Player>().m_PlayerNickName.ToString() + ", ";
        }
    }
    playerWinnerNickName += multipleWinnerNickName;
}
}

```

Рис. 3.31. Визначення переможця

Як тільки переможець стане відомим, сервер відправить його ігровий нікнейм усім гравцям матчу (див. рис. 3.32), а потім відключить їх усіх від кімнати, відправивши їх тим самим у меню (див. рис. 3.33 та рис. 3.35). Результати гри будуть збережені в базі даних (див. рис. 3.34), а сама кімната буде видалена (див. рис. 3.36).



```

Ссылка 2
public struct SendNickNameWinnerList : INetworkSerializable
{
    public string[] _playerWinnerNickNameArray;

    Ссылка 0
    public void NetworkSerialize<T>(BufferSerializer<T> serializer) where T : IReaderWriter
    {
        int arrayLength = 0;

        if (serializer.IsWriter)
        {
            arrayLength = _playerWinnerNickNameArray.Length;
        }

        serializer.SerializeValue(ref arrayLength);

        if (serializer.IsReader)
        {
            _playerWinnerNickNameArray = new string[arrayLength];
        }

        for (int i = 0; i < arrayLength; i++)
        {
            serializer.SerializeValue(ref _playerWinnerNickNameArray[i]);
        }
    }
}

```

Рис. 3.32. Відправка нікнейму переможця всім гравцям

```

////////////////////////////////////
Ссылка 1
public IEnumerator BackAllPlayerToMenuServer(string winnerPlayerWickname)
{
    List<int> playerSkinMoneyIncomeList = new();
    List<int> playerIDList = new();
    foreach (GameObject player in _roomResourcesDataObject.GetComponent<RoomPlayerResourcesData>().m_RoomPlayerResourcesList)
    {
        string playerNickName = player.GetComponent<PlayerResources>().m_ResourcesOwnerPlayerNickName;
        int playerScore = player.GetComponent<PlayerResources>().m_PlayerScore;

        int playerSkinMoneyIncome = playerScore / 3;

        if(winnerPlayerWickname == playerNickName)
        {
            playerSkinMoneyIncome = _playerSkinMoneyIncome * 2;
        }

        playerSkinMoneyIncomeList.Add(playerSkinMoneyIncome);
        playerIDList.Add(player.GetComponent<PlayerResources>().m_ResourcesOwnerPlayerID);

        StartCoroutine(ServerSkinMoneyUpdate(playerNickName, playerSkinMoneyIncome));
    }

    //только для тех кто в комнате
    List<ulong> clientsIDList = new();
    foreach (GameObject player in GetComponentInChildren<PlayerList>().m_PlayerList)
    {
        player.transform.SetParent(null);

        clientsIDList.Add((ulong)player.GetComponent<Player>().m_ClientID);
    }
    ClientRpcParams clientRpcParams = new ClientRpcParams
    { Send = new ClientRpcSendParams { TargetClientIds = clientsIDList.ToArray() } };
    //////////////////////////////////////
    BackToMenuAllClientRpc(playerIDList.ToArray(), playerSkinMoneyIncomeList.ToArray(), clientRpcParams);

    yield return new WaitForSeconds(1.0f);
    DestroyRoom();

    yield return null;
}

```

Рис. 3.33. Обробка даних після завершення гри

```

Ссылка: 1
private IEnumerator ServerSkinMoneyUpdate(int[] playerSkinMoneyIncome)
{
    WWWForm form = new();
    form.AddField("PlayerNickName", playerNickName);
    form.AddField("PlayerSkinMoneyIncome", playerSkinMoneyIncome);

    using (UnityWebRequest www = UnityWebRequest.Post("https://distributed-space.com/DistributedSpaceSkinMoneyIncome.php", form))
    {
        yield return www.SendWebRequest();

        if (www.result != UnityWebRequest.Result.Success)
        {
            Debug.Log(www.error);
            Debug.Log("На цьому етапі виникла помилка при надсиланні даних до серверної бази");
        }
    }

    yield return null;
}

```

Рис. 3.34. Надсилання результатів гри до бази даних

```

[ClientRpc]
Ссылка: 1
private void BackToMenuAllClientRpc(int[] playerIDArray, int[] playerSkinMoneyIncome, ClientRpcParams clientRpcParams = default)
{
    GameObject music = GameObject.FindGameObjectWithTag("Music");
    music.GetComponent().Stop();

    GameObject spawnedResultsPanel = Instantiate(_playerResultsPrefabObject, Vector3.zero, Quaternion.identity);
    for (int i = 0; i < playerSkinMoneyIncome.Length; i++)
    {
        if (GetComponentInChildren<PlayerList>().m_PlayerList[0].GetComponent<Player>().m_PlayerID == playerIDArray[i])
        {
            spawnedResultsPanel.GetComponent<MatchResults>().MatchResultsCalculate(playerSkinMoneyIncome[i]);
        }
    }

    SceneManager.LoadScene("Menu");
}

```

Рис. 3.35. Повернення до меню гравців

```

Ссылка: 2
private void DestroyRoom()
{
    foreach (GameObject room in _roomListObject.GetComponent<RoomList>().m_RoomList.ToArray())
    {
        if (room.GetComponent<Room>().m_RoomId == m_RoomId)
        {
            _roomListObject.GetComponent<RoomList>().m_RoomList.Remove(room);
        }
    }

    NetworkManager.Singleton.OnClientDisconnectCallback -= Singleton_OnClientDisconnectCallback;

    if (m_IsAutoRoom)
    {
        if (m_MapSize == "S")
        {
            _roomListObject.GetComponent<RoomList>().CreateAutoRoomSizeS();
        }
        else if (m_MapSize == "M")
        {
            _roomListObject.GetComponent<RoomList>().CreateAutoRoomSizeM();
        }
        else if (m_MapSize == "L")
        {
            _roomListObject.GetComponent<RoomList>().CreateAutoRoomSizeL();
        }
    }

    Destroy(GetComponentInChildren<PlayerList>().gameObject);
    Destroy(gameObject);
}

```

Рис. 3.36. Видалення кімнати.



Для клієнтської частини було використано іншу версію гри, виконує логіку клієнта. Основні функції:

- Відображення ігрових кімнат
- Підключення до ігрових кімнат
- Завантаження картки
- Завантаження даних про території
- Можливість купівлі загонів, елементи кастомізації
- Можливість зміни зовнішнього вигляду території

Для підключення до ігрової кімнати гравцеві потрібно надіслати запит серверу на підключення (див. мал. 3.37).

```
Ссылка: 1
public void PlayerJoinToRoomRequest(int roomID)
{
    Debug.Log("Отправлен запрос на подключение к комнате " + roomID);
    roomUiBlock = true;

    PlayerJoinToRoomRequestServerRpc(m_PlayerID, m_ClientID, roomID);
}
```

Рис. 3.37. Запит на підключення до сервера.

Після підключення до кімнати гравець відправляє до кімнати:

- ігровий нікнейм
- вибрані кольори
- вибрані елементи кастомізації

Користувачеві призначаються потрібні змінні, камера гравця переміщується в потрібну позицію (див. рис. 3.38 та рис. 3.39). Також гравцеві підвантажуються карта (див. рис. 3.40 та рис. 3.41).

```

Служок 1
public void PlayerInRoomInitialization()
{
    if (NetworkManager.Singleton.IsClient && IsOwner)
    {
        _roomObject = GetComponentInParent<Room>().gameObject;
        _worldObject = _roomObject.GetComponent<Room>()._worldObject;

        _cameraObject = GameObject.FindGameObjectWithTag("MainCamera");

        _gameUIObject = GameObject.FindGameObjectWithTag("GameUI");

        _gameUIObject.GetComponent<GameUI>().m_WorldObject = _worldObject;

        _roomObject.name = "Room" + "[" + _roomObject.GetComponent<Room>().m_RoomId.Value + "]";

        _roomObject.GetComponentInChildren<PlayerList>().gameObject.name = "RoomPlayerListData";

        _gameUIObject.GetComponent<GameUI>().m_Text_PlayerNickName.text = m_PlayerWickName;

        _worldObject.GetComponent<WorldInteract>().m_MyPlayer = gameObject;
        _worldObject.GetComponent<WorldInteract>()._roomObject = _roomObject;
        _worldObject.GetComponent<WorldInteract>()._GameUI = _gameUIObject;

        _roomObject.GetComponent<Room>().isRoomInGame = true;

        PlayerRoomInitializationServerRpc(m_PlayerID, m_ClientID, m_PlayerNickName);

        PlayerButtonListeners();
    }
}

////////////////////////////////////
[ServerRpc]
Служок 1
private void PlayerRoomInitializationServerRpc(int playerID, ulong playerClientID, string playerNickName)
{
    Debug.Log("В Комнату зашел игрок с ID " + m_PlayerID + " и ником " + m_PlayerNickName);

    _roomObject = GetComponentInParent<Room>().gameObject;

    StartCoroutine(_roomObject.GetComponent<Room>().TerritorySpawnOnClient(playerID, playerClientID, playerNickName));
}

```

Рис. 3.38. Надсилання даних серверу та налаштування гравця в кімнаті

```

[ServerRpc]
Служок 1
public void PlayerSendColorServerRpc(Color32 playerSkinColorMain, Color32 playerSkinColorImage, Color32 playerSkinColorBorder, int skinMainID, int skinImageID, int skinBorderID)
{
    m_PlayerColorSkinMain = playerSkinColorMain;
    m_PlayerColorSkinImage = playerSkinColorImage;
    m_PlayerColorSkinBorder = playerSkinColorBorder;

    m_MyPlayerSquads.GetComponent<PlayerSquads>().m_SquadPlayerOwnerSkinColorMain = playerSkinColorMain;
    m_MyPlayerSquads.GetComponent<PlayerSquads>().m_SquadPlayerOwnerSkinColorImage = playerSkinColorImage;
    m_MyPlayerSquads.GetComponent<PlayerSquads>().m_SquadPlayerOwnerSkinColorBorder = playerSkinColorBorder;

    m_MyPlayerSquads.GetComponent<PlayerSquads>().m_SquadPlayerTerritorySkinMainID = skinMainID;
    m_MyPlayerSquads.GetComponent<PlayerSquads>().m_SquadPlayerTerritorySkinImageID = skinImageID;
    m_MyPlayerSquads.GetComponent<PlayerSquads>().m_SquadPlayerTerritorySkinBorderID = skinBorderID;

    m_MyPlayerResources.GetComponent<PlayerResources>().m_ResourcesPlayerOwnerSkinColorMain = playerSkinColorMain;
    m_MyPlayerResources.GetComponent<PlayerResources>().m_ResourcesPlayerOwnerSkinColorImage = playerSkinColorImage;
    m_MyPlayerResources.GetComponent<PlayerResources>().m_ResourcesPlayerOwnerSkinColorBorder = playerSkinColorBorder;

    m_MyPlayerResources.GetComponent<PlayerResources>().m_ResourcesPlayerTerritorySkinMainID = skinMainID;
    m_MyPlayerResources.GetComponent<PlayerResources>().m_ResourcesPlayerTerritorySkinImageID = skinImageID;
    m_MyPlayerResources.GetComponent<PlayerResources>().m_ResourcesPlayerTerritorySkinBorderID = skinBorderID;

    StartCoroutine(_roomObject.GetComponent<Room>().SinglePlayerListUIUpdate(m_PlayerID, m_ClientID, m_PlayerNickName));
    // Debug.Log("Данные user " + "E:" + m_PlayerColor.r + " G:" + m_PlayerColor.g + " B:" + m_PlayerColor.b);
}

```

Рис. 3.39. Надсилання кольорів та шкінів серверу

```

class:1
public IEnumerator TerritorySpawnOnClient(int playerID, string playerClientID, string playerNickName)

    Debug.Log("Class Name: Hypocry " + playerNickName);

    List<bool> territoryDesant = new();

    List<int> territoryIncome = new();

    List<int> territoryScore = new();

    List<float> x = new();
    List<float> y = new();

    int territoryCount = _worldObject.GetComponent<TerritoryList>().m_Territories.Count;

    for (int i = 0; i < _worldObject.GetComponent<TerritoryList>().m_Territories.Count; i++)

        x.Add(_worldObject.GetComponent<TerritoryList>().m_Territories[i].transform.position.x);
        y.Add(_worldObject.GetComponent<TerritoryList>().m_Territories[i].transform.position.y);

        territoryDesant.Add(_worldObject.GetComponent<TerritoryList>().m_Territories[i].GetComponent<Territory>().m_IsTerritoryDesant);
        territoryIncome.Add(_worldObject.GetComponent<TerritoryList>().m_Territories[i].GetComponent<Territory>().m_TerritoryIncome);
        territoryScore.Add(_worldObject.GetComponent<TerritoryList>().m_Territories[i].GetComponent<Territory>().m_TerritoryScore);

    ClientRpcParams clientRpcParams = new ClientRpcParams
    { Send = new ClientRpcSendParams { TargetClientIds = new ArrayList { playerClientID } } };

    TerritorySpawnClientRpc(x.ToArray(), y.ToArray(), territoryCount, playerNickName, territoryDesant.ToArray(), territoryIncome.ToArray(), territoryScore
    yield return null;

```

Рис. 3.40. Надсилання даних про картку сервером гравцю

```

IEnumerator (BaseObject player in GetComponentInChildren<PlayerList>()).m_PlayerList)

    if (player.GetComponent<Player>().m_PlayerNickName == playerNickName)

        Debug.Log("Class Name:");

        for (int i = 0; i < territoryCount; i++)

            BaseObject spawnedTerritoryClient = Instantiate(_worldObject.GetComponent<TerritorySpawn>().m_TerritoryPrefabObject, new Vector3(x[i], y[i], 0), Quaternion.identity);

            spawnedTerritoryClient.transform.SetParent(_worldObject.transform);
            spawnedTerritoryClient.name = "Territory " + "(" + x[i] + ", " + y[i] + ")";

            _worldObject.GetComponent<TerritoryList>().m_Territories.Add(spawnedTerritoryClient);

            spawnedTerritoryClient.GetComponent<Territory>().m_TerritoryID = _worldObject.GetComponent<TerritoryList>().m_Territories.Count;

            spawnedTerritoryClient.GetComponent<Territory>().m_IsTerritoryDesant = territoryDesant[i];

            if (spawnedTerritoryClient.GetComponent<Territory>().m_IsTerritoryDesant == true)

                spawnedTerritoryClient.GetComponent<Territory>().m_IsTerritoryDesant.SetActive(true);

            spawnedTerritoryClient.GetComponent<Territory>().m_TerritoryIncome = territoryIncome[i];
            spawnedTerritoryClient.GetComponent<Territory>().m_Text_TerritoryIncome.text = spawnedTerritoryClient.GetComponent<Territory>().m_TerritoryIncome.ToString();

            spawnedTerritoryClient.GetComponent<Territory>().m_TerritoryScore = territoryScore[i];
            spawnedTerritoryClient.GetComponent<Territory>().m_Text_TerritoryScore.text = spawnedTerritoryClient.GetComponent<Territory>().m_TerritoryScore.ToString();

        player.GetComponent<Player>().PlayerSpawnPrefabRequest();
        player.GetComponent<Player>().PlayerSetWorldData();

```

Рис. 3.41. Створення світу на клієнті

Тепер, з додаванням лаунчера, у користувачів немає необхідності щоразу завантажувати та розпаковувати самим нову версію гри. Тепер цим займається лаунчер.

Лаунчер звіряє версію програми на сервері з версією програми у клієнта (див. рис. 3.42) і, якщо версії різні, то відбувається процес видалення старого додатка, завантаження нового архіву (3.43) в ту ж папку, його розпакування,

встановлення (див. рис. 3.44), і запуск програми (див. рис. 3.45). Також, з метою зменшення ваги програми, завантажений архів видаляється.

З боку користувача видно лише мінімалістичний інтерфейс завантаження, що робить програму легкою (у плані установки) для будь-якої людини (див. рис. 3.46).

З більш детальним кодом лаунчера можна ознайомитись у Додатку Г.

```
def VersionCheck(self):
    self.label.setText("Проверка обновлений")
    self.progressBar.setProperty("value", 0)

    versionLocal = 0.2
    versionRemote = 0

    http = urllib3.PoolManager()
    receivedData = http.request("GET", "https://distriected.space/Build/Version.txt")

    versionRemote = receivedData.data.decode('utf-8')
    versionActualString = "Текущая версия игры " + receivedData.data.decode('utf-8')

    self.label.setText(versionActualString)

    print(versionActualString)

    versionLocal = str(float(versionLocal))

    if (versionLocal == versionRemote):
        print("Версии одинаковые го играют")
        self.label.setText("Версия игры актуальная")
        #self.PlayGame()

    else:
        #self.UpdateGame()
        print("Версии разные игра пошла обновляца")
        self.label.setText("Обновление игры...")

#-----#
```

Рис. 3.42. Перевірка версій гри

```

def initDownload(self):
    self.label.setText("Обновление игры...")
    # Disable the button while the file is downloading.

    # Run the download in a new thread.
    self.downloader = Downloader(
        "https://distriected.space/Build/DistrictedSpaceUpdate.zip",
        "./DistrictedSpace/DistrictedSpaceUpdate.zip"
    )
    # Connect the signals which send information about the download
    # progress with the proper methods of the progress bar.
    self.downloader.setTotalProgress.connect(self.progressBar.setMaximum)
    self.downloader.setCurrentProgress.connect(self.progressBar.setValue)
    # Qt will invoke the `succeeded()` method when the file has been
    # downloaded successfully and `downloadFinished()` when the
    # child thread finishes.
    self.downloader.succeeded.connect(self.downloadSucceeded)
    self.downloader.finished.connect(self.downloadFinished)
    self.downloader.start()

def downloadSucceeded(self):
    # Set the progress at 100%.
    self.progressBar.setValue(self.progressBar.maximum())
    self.label.setText("Обновление загружено!")
    self.InstallGame()

```

Рис. 3.43. Завантаження оновлення

```

def InstallGame(self):
    dirname = os.path.dirname(__file__)
    archive = "./DistrictedSpace/DistrictedSpaceUpdate.zip"
    with zipfile.ZipFile(archive, 'r') as zip_file:
        zip_file.extractall(dirname)

    UpdateGameZip = "./DistrictedSpace/DistrictedSpaceUpdate.zip"
    os.unlink(UpdateGameZip)
    self.PlayGame()

```

Рис. 3.44. Встановлення оновлення

```

#-----#
def PlayGame(self):
    print("Запуск игры")
    self.label.setText("Запуск")

    dirname = os.path.dirname(__file__)
    filename = os.path.join(dirname, "DistrictedSpace.exe")
    os.startfile(filename)

    #thT = threading.current_thread()

    #thT.join()
    #os.abort()
    os._exit(0)

```

Рис. 3.45. Запуск гри

DistrictedSpace



Рис. 3.46. Готовий лаунчер

Потім гравець запускає гру.

Після підключення до сервера гравцю надається можливість розпочати гру або відвідати магазин для купівлі кастомізації своєї території за ігрову валюту. У цілому нині головне меню виглядає так (див. рис. 3.47), а магазин (див. рис. 3.48).

При відкритті списку ігрових кімнат, гравцеві представляється на вибір щонайменше три автоматично згенеровані кімнати з різною кількістю П, а також, можливо, й інші ігрові кімнати гравців (див. рис. 3.49).



Рис. 3.47. Головне меню.





Рис. 3.48. Магазин.



Рис. 3.49. Список ігор.

На початку гри учасникам (П та користувачам) відкривається можливість купити загопи та висадитися на землю, позначену «зеленою стрілочкою». Але вже на початку гри учасники роблять вибір: вони хочуть купити два слабкі загопи, один сильний або накопичити грошей на наступний хід (див. рис. 3.50).

Дохід за хід видається відповідно до суми всіх доходів із територій, які вдалося утримати. Якщо гравець взагалі не має територій, то йому видається кожен хід по 300 монет.

Сам ігровий процес складається із захоплення чужих/порожніх територій та захисту своїх (див. рис. 3.51). У грі також є таке поняття як «окуляри» - це накопичувана, вогнетривка сума, що зберігає всі дані про будь-коли утримані

території, відзначається значком «зірочка».

Пропорція зірочок/доходів з територій розподіляється так: чим вищий дохід з території, тим менше «зірочок» гравець отримує за хід із цієї території.

У результаті перемагає той гравець, у якого найбільше «зірочок», а відслідковувати процес змагання можна, іноді переглядаючи список гравців (див. рис. 3.52).



Рис. 3.50. Ігровий процес на початку гри.



Рис. 3.51. Ігровий процес.





Рис. 3.52. Список гравців.

### Висновки по розділу 3

Таким чином, було створено працюючий прототип гри-стратегії, що складається з самого додатка, бази даних, сервера та сайту.

У першому пункті було розроблено план створення сайту, а також було описано саме створення сайту з прикладами. Докладніший код можна побачити в Додатку А, Додатку Б, Додатку В.

У другому пункті було розказано про роботу бази даних та сервера, було виділено та описано взаємозв'язок усіх елементів системи.

У третьому пункті було описано основні моменти коду гри та лаунчера до неї. На жаль, кількість файлів з кодами гри досить велика, щоб їх не можна було додати до додатків через обмеження.

Вся карта розбита на безліч територій, а метою гри є захоплення контролю над якомога більшою кількістю територій. Тому назвою в результаті стало словосполучення «Districted Space». Буквально, це перекладається як «розділене простір» або «районований простір», що цілком підходить для гри про безліч «районів», тобто, територій

## ВИСНОВКИ

Було досягнуто мету розробки – робочий прототип багатокористувацької покрокової стратегії було створено.

Розрахованим на багато користувачів його можна назвати завдяки тому, що для гравців було організовано підключення до сервера через мережу Інтернет, а в кожному матчі гри може брати участь кілька учасників.

Покрокова стратегія завдяки тому, що гра ділиться на кроки, тобто. ходи та на кожний хід у всіх учасників матчу є однакова кількість часу, після чого відбувається калькуляція всіх дій та гравці бачать уже результат.

Новизна програми полягає в тому, що хід гравці роблять не по черзі, а одночасно.

Щодо завдань, вони теж були виконані відповідно до плану:

- аналіз обраної предметної області (зокрема ігровий промисловості у жанрі стратегій різних підтипів) було виконано та її результати було представлено у Розділі 1.

Було наведено ознайомлювальну коротку історію жанру, в якій особлива увага приділялася появі нових типів стратегії, їх популярність, роль у сучасному світі, коротка типологія, щоб розумітися на нюансах сучасних стратегій.

Більш того, для визначення перспектив направлення в 1 Розділ були додані графік та укорочена версія найприбутковіших додатків на мобільних пристроях, що дозволило сказати, що створення кросплатформної гри для ПК та мобільних пристроїв має гарні перспективи.

- були проаналізовані найпопулярніші стратегії щодо їхнього інтересу для гравців (6 різних популярних ігор жанру стратегія представлені в Розділі 2).

- тип жанру для гри був обраний і офіційно звучить як «розрахована на багатокрокова покрокова стратегія», неофіційно, є досить новим – «покрокова стратегія в реальному часі».

Дійсно, гравці мають карту з різної кількості територій, що генеруються. Завдання користувача – захопити якнайбільше територій. Гра покрокова, а

значить, є певні «кроки» або «ходи», які відмірюються таймером.

Приставку «в реальному часі» було додано у зв'язку з тим, що хід відбувається у всіх гравців одночасно, що нетипово для покрокових стратегій.

- був створений робочий прототип гри, використовуючи ігровий движок Unity і для автоматичного оновлення був створений лаунчер мовою Python.

Додавання в план лаунчера полегшило і ускладнило завдання: з одного боку довелося розбиратися в мові Python і створювати з нуля лаунчер, а з іншого - тепер гравцям не потрібно щоразу скачувати і розпаковувати самим нову версію гри. Тепер цим займається лаунчер.

Лаунчер звіряє версію програми на сервері з версією програми у клієнта і, якщо версії різні, то відбувається процес видалення старого додатка, скачування нового архіву в ту ж папку, його розпакування та запуск програми.

З боку користувача видно лише мінімалістичний інтерфейс завантаження, що робить додаток легким (у плані встановлення) для будь-якої людини.

Також додаток був неодноразово протестований з групою незалежних тестувальників для виявлення багів та технічних проблем. Таким чином, за допомогою результатів опитування тестувальників було з'ясовано, що версія 1.0 (чинна на момент написання диплома) є повноцінним робочим додатком.

- сайт та база даних для реєстрації нових гравців та зберігання даних про вже існуючі облікові записи були створені та інтегровані в систему.

Сайт та сервер, БД були орендовані на віддалених пристроях для зручності та функціональності. Як було сказано раніше, ПК не може бути включений постійно: трапляються і збої, і планові відключення світла, також він може бути випадково вимкнений кимось із співмешканців. Не виключені також і поломки ПК, внаслідок яких усі дані можуть бути втрачені.

Ризик таких же проблем є і на віддалених пристроях, проте він досить малий, а у разі несправностей буде відповідна компенсація.

- програма була запущена на сервері, а значить, став доступний мультиплеєр.

Оскільки було прийнято рішення, що гра ділиться на 2 частини: Сервер і Клієнт, необхідно було робити дві збірки. Після успішного запуску серверної версії розпочався перший етап тестування мультиплеєра.

Ця робота наводить деякі (актуальні на момент написання) рішення до певного спектру завдань, пов'язаних з:

- розробкою розрахованого на багато користувачів програми на мові C# в середовищі розробки Unity
- створенням сайту з нуля на HTML
- написанням коду валідації для сайту такими мовами, як PHP та JavaScript
- інтеграцією бази даних у розрахований на багато користувачів додаток

Основною метою роботи було створення робочого прототипу стратегії для подальшого розвитку та отримання матеріальних вигод від реалізації цього додатку.

Таким чином, вибудувавши маркетинговий план (ґрунтуючись на дослідженні з'ясувати, яка стратегія реалізації більше підійде для додатку), тепер можна отримувати заробіток або від продажів екземплярів, або від запровадження рекламної системи (і відповідного пакета, який за певну плату не відображатиме рекламу зовсім), або від запровадження внутрішньо ігрових покупок

## СПИСОК ЛІТЕРАТУРИ

1. Unity и C#. Геймдев від ідеї до реалізації / Джереми Гибсон Бонд – СПб: Piter, 2019. – 928 с.
2. Unity в дії. Мультиплатформна розробка на C# / Джозеф Хокинг, Бонд – СПб: Piter, 2018. – 640 с.
3. Геймдизайн / Джесси Шелл – М: Alpina Pabliher, 2019. – 640 с.
4. Кров, піт і пікселі / Джейсон Шреер – Київ: Book Chef, 2020. – 368 с.
5. Game Development Essentials: An Introduction / Джинни Новак – Boston: Cengage Learning, 2011.-510с.
6. Mathematics for 3D Game Programming and Computer Graphics / Eric Lengyel – Boston: Cengage Learning, 2011. – 563 с.
7. Artificial Intelligence for Games / Ian Millington, John Funge – Boca Raton: CRC Press, 2009. – 896 с.
8. Unity Game Development Essentials / Will Goldstone – Birmingham: Packt Publishing, 2009. – 316 с.
9. Game Programming Patterns / Robert Nystrom – Washington: Genever Benning, 2014.-354 с.
10. GPU Pro 7: Advanced Rendering Techniques / Wolfgang Engel – Boca Raton: CRC Press, 2016. – 306 с.
11. . Game Programming Gems 8 / Mark DeLoura – Newton Centre: Charles River Media, 2010. – 640 с.
12. . Game Engine Architecture, Second Edition / Jason Gregory – Boca Raton: CRC Press, 2014. – 1052 с.
13. . Game Design Workshop: A Playcentric Approach to Creating Innovative Games, Fourth Edition / Tracy Fullerton – Boca Raton: CRC Press, 2018. – 522 с.
14. . Rules of Play: Game Design Fundamentals / Katie Salen Tekinbas, Eric Zimmerman – Cambridge: The MIT Press, 2003. – 688 с.

15. . The Art of Game Design: A Book of Lenses, Third Edition / Jesse Schell – Boca Raton: CRC Press, 2019. – 610 с.
16. . What Video Games Have to Teach Us About Learning and Literacy. Second Edition / James Paul Gee – New York: St. Martin's Griffin, 2007. – 256 с.
17. . Android Game Recipes: A Problem-Solution Approach / Jerome DiMarzio – New York: Apress, 2014. – 275 с.
18. . Гени генія / Хидео Кодзима – СПб: Piter, 2021. – 320 с.
19. . Game Programming Patterns / Robert Nystrom – Washington: Genever Benning, 2014. – 354 с.
20. . Design Patterns / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides – Boston: Addison-Wesley Professional, 1994. – 416 с.
21. . Hey! Listen!: A journey through the golden era of video games / Steve McNeil – London: Headline, 2019. – 288 с.
22. . Game Engine Architecture, Third Edition / Jason Gregory – Boca Raton: CRC Press, 2018. – 1240 с.
23. . Location-Based Mobile Games: Design Perspectives (SpringerBriefs in Applied Sciences and Technology) / Davide Spallazzo, Ilaria Mariani – Berlin: Springer, 2018. – 108 с.
24. . Artificial Intelligence and Games / Georgios N. Yannakakis, Julian Togelius – Berlin: Springer, 2018. – 360 с.
25. . Learn to implement games with code / John M. Quick – Boca Raton: CRC Press, 2017. – 355 с.
26. . Games, Design, and Play / Colleen Macklin, John Sharp – Boston: Addison-Wesley Professional, 2016. – 288 с.
27. . Game Audio Programming: Principles and Practices / Guy Somberg – Boca Raton: CRC Press, 2016. – 312 с.
28. . Комп'ютерні стратегії німецької школи / Jesse Russell, Ronald Cohn – М: Книга на вимогу, 2013. – 104 с.

29. . Games of Strategy, Fourth Edition / Avinash K. Dixit, Susan Skeath, David H. Reiley Jr – New York: W. W. Norton & Company, 2014. – 732 с.
30. . Ігри і стратегії с точки зору математики / А. Шень – МЦНМО, 2008. – 40 с.
31. . Introduction to Database Systems / C. J. Date – London: Pearson, 2003. – 1040 с.
32. . Database Systems: A Practical Approach to Design, Implementation, and Management / Thomas Connolly, Carolyn Begg – London: Pearson, 2014. – 1440 с.
33. . Helping you thrive in the games market [Електронний ресурс] – Режим доступу до ресурсу: <https://newzoo.com/>
34. . Empowering people with data [Електронний ресурс] – Режим доступу до ресурсу: <https://www.statista.com/>
35. . Обираємо кращу частину Civilization [Електронний ресурс] – Режим доступу до ресурсу: <https://www.cybersport.ru/tags/games/pyataya-koronnaya-shestaya-pokhoronnaya-vybiraem-luchshuyu-chast-civilization>
36. . Why is Civilization 5 still more popular than Civilization 6? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.pcgamer.com/why-is-civilization-5-still-more-popular-than-civilization-6/>
37. . Frozen Synapse 2: Обзор [Електронний ресурс] – Режим доступу до ресурсу: [https://stopgame.ru/show/101400/frozen\\_synapse\\_2\\_review](https://stopgame.ru/show/101400/frozen_synapse_2_review)

