

## ДОДАТОК А

## Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

Дата звіту 5/26/2025

Дата редагування ---

Звіт не був оцінений

---

### Звіт подібності

---

#### метадані

---

Назва організації  
**Kharkiv National University of Radio Electronics**

Заголовок  
**2025\_Б\_ПІ\_ПЗПІ\_21\_5\_Шапошніков\_Л\_Л\_скорочений**

Автор                      Науковий керівник / Експерт  
**Шапошніков Л.І.Новіков Ю.С./Нечволод Вадим Юрійович**

підрозділ  
**каф. ПІ**

---

#### Обсяг знайдених подібностей

---

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

7.49%  
7.49%

КП 1

25

Довжина фрази для коефіцієнта подібності 2

1.15%  
1.15%

КЦ

7890

Кількість слів

1.15%  
1.15%

КЦ

56193

Кількість символів

---

#### Тривога

---

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		5
Інтервали		0
Мікропробіли		1
Білі знаки		0
Парафрази (SmartMarks)		34

---

#### Подібності за списком джерел

---

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копію тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз		Копію тексту
ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	<a href="https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download">https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download</a>	179 2.27 %
2	<a href="https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download">https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download</a>	134 1.70 %
3	<a href="https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download">https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download</a>	90 1.14 %
4	<a href="https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download">https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download</a>	89 1.13 %
5	<a href="https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download">https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download</a>	28 0.35 %

6	<a href="https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download">https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download</a>	27	0.34 %
7	<a href="https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download">https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download</a>	14	0.18 %
8	<a href="https://openarchive.nure.ua/server/api/core/bitstreams/38c96869-4c77-4ae1-842a-7400de5fc626/content">https://openarchive.nure.ua/server/api/core/bitstreams/38c96869-4c77-4ae1-842a-7400de5fc626/content</a>	12	0.15 %
9	<a href="https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download">https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download</a>	7	0.09 %
10	<a href="https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download">https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download</a>	6	0.08 %

### з бази даних RefBooks (0.00 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-----------	--

### з домашньої бази даних (0.00 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-----------	--

### з програми обміну базами даних (0.00 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-----------	--

### з Інтернету (7.49 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	<a href="https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download">https://openarchive.nure.ua/bitstreams/f6b97e44-0000-480c-a104-bdb41918048b/download</a>	579 (10) 7.34 %
2	<a href="https://openarchive.nure.ua/server/api/core/bitstreams/38c96869-4c77-4ae1-842a-7400de5fc626/content">https://openarchive.nure.ua/server/api/core/bitstreams/38c96869-4c77-4ae1-842a-7400de5fc626/content</a>	12 (1) 0.15 %

### Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-------	---------------------------------------

9

#### ВСТУП

Сучасна ігрова індустрія є однією з найдинамічніших галузей цифрового середовища. З кожним роком зростає кількість як великих студійних релізів, так і незалежних проєктів, які експериментують із механіками, жанрами та візуальними стилями. Особливої популярності набувають ігри, що пропонують гравцям змагальний досвід, глибoku кастомізацію та елементи колекціонування — саме тому все більше уваги привертають PvP-ігри з економічними та тактичними складовими.

У межах кваліфікаційної роботи було створено ігровий застосунок *Legions of the Vault* — колекційну покрову PvP-стратегію, яка поєднує елементи бойової гри, економічної системи та процедурної генерації юнітів. Основна ідея полягає у тому, що гравці збирають армії з унікальних юнітів, відкриваючи їх через систему лутбоксів, а потім змагаються між собою за допомогою стратегічно розрахованих дій у покровому бою.

У грі реалізовано систему вартості та рідкості юнітів, налаштування характеристик (здоров'я, атака), а також можливість їх вдосконалення. Економічна модель гри включає геймплейну валюту, механіку розплення непотрібних юнітів для отримання ресурсів, та системи прогресії гравця, що стимулюють регулярну взаємодію з грою.

Покровові PvP-стратегії вирізняються тим, що вимагають від гравця не лише реакції, а й глибокого тактичного мислення. Ретельно побудований баланс дозволяє зробити гру чесною для обох сторін, а гнучка економіка — підтримувати інтерес гравців до відкриття нових комбінацій і збирання унікальних армій. Метою цієї роботи є дослідження, проєктування та реалізація систем балансу

ДОДАТОК Б  
Слайди презентації



Рисунок Б.1 – Слайд 1

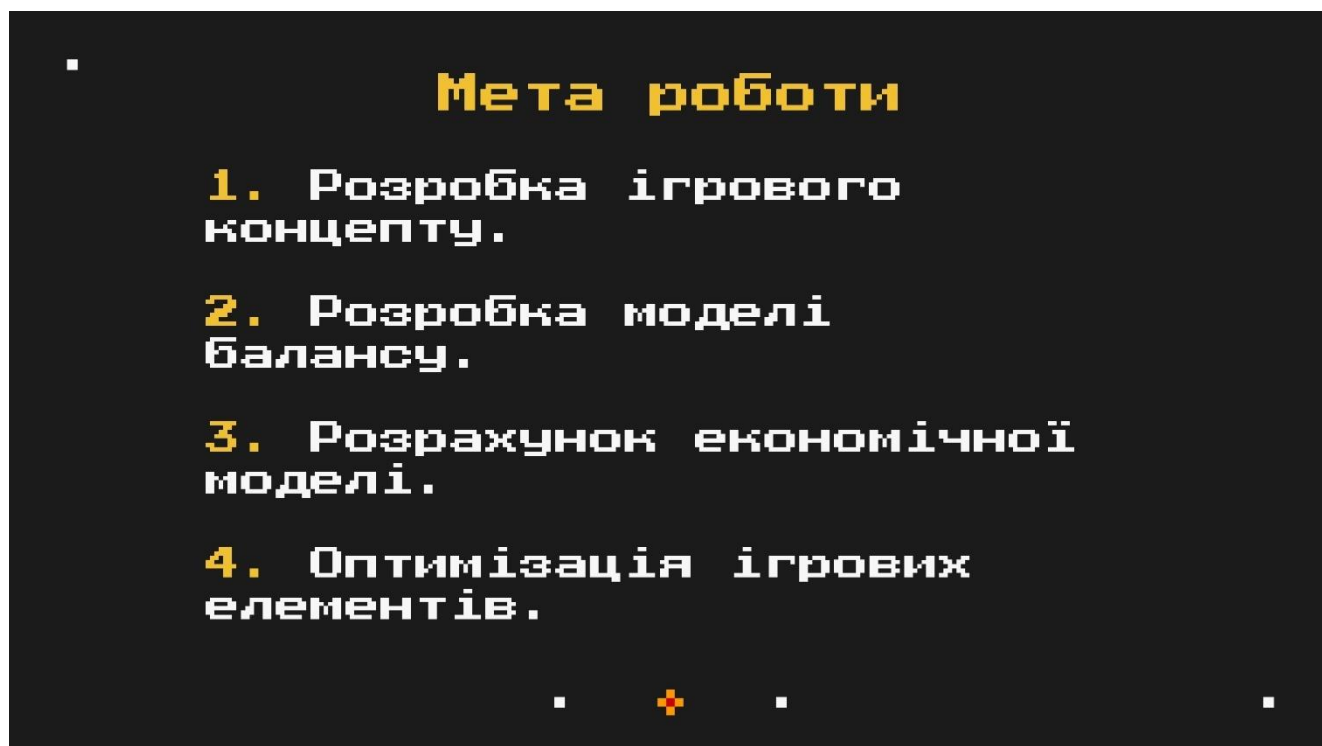


Рисунок Б.2 – Слайд 2

## Ідея проекту

**ККІ + Варгейми = Щось нове**  
 Елементи колекціонування. Елементи армії-будування та бойовки. Доступний та новий досвід



Рисунок Б.3 – Слайд 3

## Ідеї геймплею

1. Головна ціль – вбити лідера суперника.
2. Великий різновид юнітів та можливих армій.
3. Унікальна генерація юнітів.
4. Потенціал для додаткових стратегічних елементів.

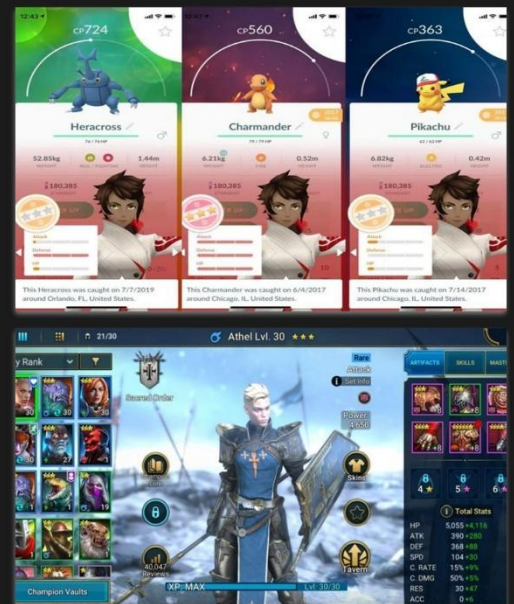


Рисунок Б.4 – Слайд 4

## Модель Балансу

### 1. Типи юнітів.

Архетип	Особливість
Файтер	Має середні характеристики.
Цілитель	Практично не завдає шкоди. Мультітул.
Танк	Мало завдає шкоди, але багато її витримує.
Асасин	Мало здоров'я – багато шкоди.

Рисунок Б.5 – Слайд 5

## Модель Балансу

### 2. Складові юніта.



Ім'я	Діапазон Здоров'я	Можлива атака (1)		Пересування
		Шкода	Опис	
Файтер	20-40	15	Ближній бій [1]	2
		20	Ближній бій [1], завдає 15 шкоди собі	
		10	Ближній бій [1], Руїниана (10)	
Цілитель	30-50	0	Ремонт [1] (30)	3
		0	Заплення [1] (10)	
		0	Руїниана [1] (30)	
Танк	40-60	10	Ближній бій [1]	1
		5	Блок (10)	
		15	Ближній бій [1], завдає 10 шкоди собі	
Асасин	10-20	20	Ближній бій [1]	2
		30	Ближній бій [1], завдає 10 шкоди собі	
		10	Дальній бій [3]	

Рисунок Б.6 – Слайд 6

## Модель Балансу

### 3. Вартість та транзитивні величини.

- З підвищенням вартості здоров'я збільшується на 30 одиниць, окрім асасина, у якого приріст в 15 одиниць
- Шкода юнітів балансується відносно архетипів.
- Зміна кількості можливих атак.
- Шкоду можна міняти на додаткові ефекти.

Легенда	
[X] та (X)	Число в [] скобках визначає дальність дії, а число в () скобках визначає числовий показник
Ближній бій [X]	атакує ворога або стіни на відстані X, зазвичай не далі однієї або двох клітин
Дальній бій [X]	атакує ворога або стіни на відстані X
Руйнівна (X)	додатково завдає X шкоди конструкціям
Ремонт [X] (X)	Відновлює міцність стіни на дальності [X] на значення (X)
Зцілення [X] (X)	Відновлює здоров'я юніта на дальності [X] на значення (X)
Блок (X)	Відіймає X шкоди від наступних входящих атак та ворожих здібностей

Рисунок Б.7 – Слайд 7

## Економічна Модель

### 1. Основи:

- Розширення колекції через лутбокси
- Генерація та прокачка юнітів
- Створення ігрового циклу побудованому на розширенні та покращенні колекції

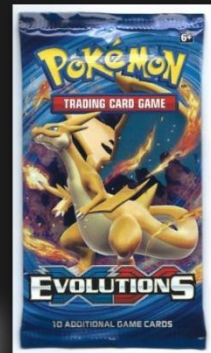


Рисунок Б.8 – Слайд 8

## Економічна Модель

### 2. Ймовірності та статистика

Вартість	1 слот	2 слот	3 слот	4 слот	5 слот	Лідер слот
1	100%	100%	0%	0%	0%	0%
2	0%	0%	78,5%	59,5%	0%	0%
3	0%	0%	20%	30%	50%	0%
4	0%	0%	1,5%	10%	30%	0%
5	0%	0%	0%	0,5%	20%	0%
Лідер	0%	0%	0%	0%	0%	5% (окремо від 5 основних)

В середньому за 20 лутбоксів (десь 40 днів гри) гравець отримує:

- 40 юнітів 1 вартості
- 27-28 юнітів 2 вартості
- 20 юнітів 3 вартості
- 8-9 юнітів 4 вартості
- 4-5 юнітів 5 вартості
- 1 лідер.

Рисунок Б.9 – Слайд 9

## Економічна Модель

### 3. Переведення юнітів в валюту для покращень

1 вартість	1 кристал
2 вартість	3 кристали
3 вартість	6 кристалів
4 вартість	15 кристалів
5 вартість	30 кристалів
Лідер	60 кристалів

Рисунок Б.10 – Слайд 10

## Економічна Модель

### 4. Система покращень

	+Здоров'я	Зміна Атаки
1 вартість	3	10
2 вартість	5	15
3 вартість	7	20
4 вартість	10	25
5 вартість	12	30

Повний апгрейд юніта може обійтись максимум в 4 покращення здоров'я та близько двох змін атак.

Вартості всіх покращень було розраховано так, щоб десь за 40 днів гри можна було повністю покращити одного юніта 5 вартості

Рисунок Б.11 – Слайд 11

## Оптимізація ігрового поля

### 1. Оптимальна кількість клітин.

- Максимум юнітів на дошці – 22.
- Швидкість пересування від 1 до 5 клітин
- Потрібно мінімум 4 клітини на юніта
- Квадратне поле занадто скуче або габаритне

Результат: поле 13x16

Варіант мапи: Вузький прохід (28 стін)

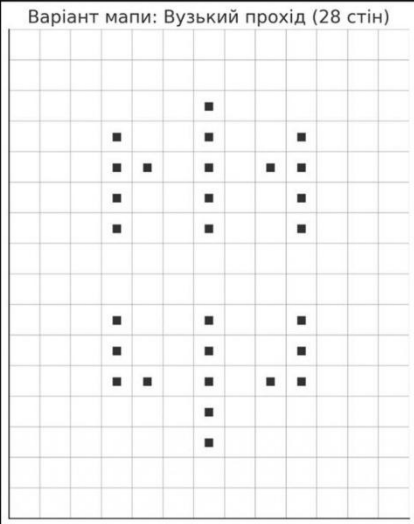


Рисунок Б.12 – Слайд 12

# Оптимізація ігрового поля

## 2. Перешкоди та різноманіття.

- Стандарт у 10%-15%.

**Результат: 19-28 перешкод.**

**Різні сценарії - різні переваги.**

Варіант мапи: Вузький прохід (28 стін)

Рисунок Б.13 – Слайд 13

# Впровадження

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
РАДІОЕЛЕКТРОНІКИ

**МАТЕРІАЛИ**  
**29-го МІЖНАРОДНОГО МОЛОДІЖНОГО ФОРУМУ**  
**«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У XXI СТОЛІТТІ»**

**КОНФЕРЕНЦІЯ**  
**«АКТУАЛЬНІ ПРОБЛЕМИ ЕКОНОМІЧНОЇ**  
**КІБЕРНЕТИКИ ТА ЕКОНОМІЧНОЇ БЕЗПЕКИ»**

**CONFERENCE**  
**«CURRENT PROBLEMS OF ECONOMIC CYBERNETICS**  
**AND ECONOMIC SECURITY»**

16-19 квітня 2025 р.  
Том 8

Харків 2025

УДК 330.4  
**МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ЕКОНОМІЧНОЇ СИСТЕМИ У**  
**FREE-TO-PLAY ГРИ: БАЛАНСУВАННЯ ВИТРАТ І ПРОГРЕСУ**  
**ГРИВЛЯ**

Шановников І.І.  
e-mail: lev.shanovnikov@pnu.edu.ua  
Харківський національний університет радіоелектроніки, каф. ПІ  
м. Харків, Україна

This paper explores the development and optimization of an economic system in a free-to-play game based on lootbox mechanics. The study presents the principles of in-game currency distribution, the probability-based unit drop system, and the balance of upgrade costs. Using mathematical modeling and probability theory, the optimal pricing for unit upgrades and modifications was determined, ensuring a balance between player progression and monetization. The results provide a framework for designing sustainable game economies that maintain player engagement while preventing economic stagnation.

Економічна система в free-to-play іграх повинна стимулювати як геймплейний прогрес, так і можливість монетизації. У такому ігровому процесі використовується система випадкового отримання юнітів і лутбоксів, а також система внутрішньої ігрової валюти для покращення юнітів. Основна мета дослідження – розробка математичної моделі балансування витрат гравця, яка б забезпечила комфортний темп прогресу та утримання користувачів.

Почнемо з лутбоксів, гравець зможе відкрити по одному лутбоксу кожні 2-3 дні та додатково ще один кожні 8-12 днів. Ця інформація є важливою для розрахунку того, скільки в середньому часу гравець має витратити для отримання або покращення базового юніта.

У кожному лутбоксі гравцю випадає 5 юнітів і, з малюю окремою ймовірністю, може випасти 6-й юніт-лідер. Ймовірність випадання юнітів ми можемо побачити у цій таблиці:

Вартість	1 слот	2 слот	3 слот	4 слот	5 слот	Лідер слот
1	100	100	0	0	0	0
2	0	0	78,5	59,5	0	0
3	0	0	20	30	50	0
4	0	0	1,5	10	30	0
5	0	0	0	0,5	20	0
Лідер	0	0	0	0	0	5

56

Так, у середньому виходить, що за один лутбокс гравець отримує: 2,0 юніта вартості 1; 1,38 юніта вартості 2; 1,0 юніта вартості 3; 0,415 юніта вартості 4; 0,205 юніта вартості 5; 0,05 лідера.

Тоді за формулою ми можемо розрахувати скільки в середньому треба відкрити лутбоксів, щоб отримати хоча б одного юніта певної вартості:

$$L = \frac{1}{P_x}$$

де P – середня кількість юнітів вартості X за один лутбокс, що виводить нас на такі результати: 1 лутбокс на юніта 1 вартості; 1,45 лутбоквів на юніта 2 вартості; 1 лутбокс на юніта 3 вартості; 2,41 лутбоквів на юніта 4 вартості; 4,88 лутбоквів на юніта 5 вартості; 20 лутбоквів на юніта-лідера.

Таким чином, у нас є економічна система для коротко-часової утримки гравця. Для ж довгострокової утримки я додав механіку покращення юнітів, яка також має свою економічну складову.

У такому процесі юнітам можна покращувати здоров'я та змінювати атаки. Вартість цих послуг можна розрахувати за такими формулами:

$$C_{зд}(v, u) = B_{зд} \times f(u)$$

де  $C_{зд}(v, u)$  – вартість покращення здоров'я юніта вартості (v) на покращенні номер (u);  
 $B_{зд}$  – базова вартість покращення здоров'я для юніта вартості (v);  
 $f(u)$  – множник залежно від разів покращення.  
 А вартість зміни атаки розраховується за формулою:

$$C_{ата}(v) = k_v \times D$$

де  $C_{ата}(v)$  – вартість зміни атаки юніта вартості (v);  
 $D$  – базова вартість зміни атаки;  
 $k_v$  – коефіцієнт вартості для кожного різно юніта.  
 Звичайно повне покращення юніта буде включати в себе чотири покращення здоров'я і приблизно одну зміну атаки. Для встановлення економічної системи на покращення необхідно ввести окрему валюту. В випадку такої проєкту я ввів валюту, яку можна отримувати лише за виділення юнітів зі своєї колекції. Назвемо для зручності цю валюту "кристалами". Так я вирішив встановити такий курс конверсії: за юніта 1 вартості – 1 кристал; за 2 вартості – 3 кристали; за 3 вартості – 6 кристали; за 4 вартості – 15 кристали; за 5 вартості – 30 кристали. Ця інформація

57

Рисунок Б.14 – Слайд 14

## Висновки

1. Розроблено ігровий концепт.
2. Розроблено модель балансу гри.
3. Розроблено економічну модель гри.
4. Оптимізовано ігрові елементи.



Рисунок Б.15 – Слайд 15

**ДОДАТОК В**

Тези за темою кваліфікаційної роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
РАДІОЕЛЕКТРОНІКИ

**МАТЕРІАЛИ**  
**29-го МІЖНАРОДНОГО МОЛОДІЖНОГО ФОРУМУ**  
**«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У ХХІ СТОЛІТТІ»**

**КОНФЕРЕНЦІЯ**  
**«АКТУАЛЬНІ ПРОБЛЕМИ ЕКОНОМІЧНОЇ**  
**КІБЕРНЕТИКИ ТА ЕКОНОМІЧНОЇ БЕЗПЕКИ»**

**CONFERENCE**  
**«CURRENT PROBLEMS OF ECONOMIC CYBERNETICS**  
**AND ECONOMIC SECURITY»**

16–19 квітня 2025 р.  
Том 8

Харків 2025

Рисунок В.1 - Титульна сторінка матеріалів конференції

## АЛФАВІТНИЙ ПОКАЖЧИК

Aliyev R.F.	<b>A</b>	6	Носарева А.Е.	<b>Н</b>	35
Rzazadeh T.I.	<b>R</b>	8	Панченко М.П.	<b>П</b>	40
Антонович В.Д.	<b>A</b>	87	Перепелюкова О.В.		75
Батіг В.В.	<b>Б</b>	10	Петренко Д.А.		42
Білаш Д.Р.		63	Пивоваров М.О.		44
Будяньський В.С.		12	Полозов М.О.		59
			Полозов О.Б.		78
			Полозова О.О.		40
			Помогалова Н.В.		81
Вешкін Є.П.	<b>B</b>	66	Соколова Л.В.	<b>C</b>	24
Герасимюк Д.Ю.	<b>Г</b>	81	Степаненко Р.Д.		84
Демченко Д.С.	<b>Д</b>	27	Сукнов О.М.		47
Дерюшев В.С.		15	Турчин О.А.	<b>T</b>	72
Жила О.В.	<b>Ж</b>	15, 21	Харченко В.В.	<b>X</b>	50
Іванов І. О.	<b>I</b>	18	Худяков Д.Л.		66
Зінов'єв А.П.	<b>З</b>	18	Чємов Є.Р.	<b>Ч</b>	53
			Чемеренко А.А.		90
Казанцева С.С.	<b>К</b>	69	Шапошніков Л.І.	<b>Ш</b>	56
Караєв Е.А.		21	Шеренок К.К.		87
Кодрул Р.Е.		24			
Кравець М.Ю.		53		<b>Щ</b>	
Лученко П.Ф.	<b>Л</b>	27	Щербак В.В.		59
Малигіна Т.Д.	<b>M</b>	63			
Мізін Д.С.		72			
Москальова М.С.		30, 33			
Мурзабулатова М.С.		37			

Рисунок В.2 - Алфавітний покажчик

УДК 330.4

**МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ЕКОНОМІЧНОЇ СИСТЕМИ У  
FREE-TO-PLAY ГРІ: БАЛАНСУВАННЯ ВИТРАТ І ПРОГРЕСУ  
ГРАВЦЯ**

Шапошніков Л.І.

e-mail: lev.shaposhnikov@nure.ua

Харківський національний університет радіоелектроніки, каф. ПІ  
м. Харків, Україна

This paper explores the development and optimization of an economic system in a free-to-play game based on lootbox mechanics. The study presents the principles of in-game currency distribution, the probability-based unit drop system, and the balance of upgrade costs. Using mathematical modeling and probability theory, the optimal pricing for unit upgrades and modifications was determined, ensuring a balance between player progression and monetization. The results provide a framework for designing sustainable game economies that maintain player engagement while preventing economic stagnation.

Економічна система в free-to-play іграх повинна стимулювати як геймплейний прогрес, так і можливість монетизації. У моєму ігровому проєкті використовується система випадкового отримання юнітів із лутбоксів, а також система внутрішньо ігрової валюти для покращення юнітів. Основна мета дослідження – розробка математичної моделі балансування витрат гравця, яка б забезпечила комфортний темп прогресу та утримання користувачів.

Почнемо з лутбоксів, гравець зможе відкривати по одному лутбоксу кожні 2-3 дні та додатково ще один кожні 8-12 днів. Ця інформація є важливою для розрахунків того, скільки в середньому часу гравець має витрати для отримання або покращення бажаного юніта.

У кожному лутбоксі гравцю випадає 5 юнітів і, з малою окремою ймовірністю, може випасти 6-й юніт-лідер. Ймовірність випадання юнітів ми можемо побачити у цій таблиці:

Таблиця 1 – Ймовірність випадання юнітів певної вартості в певному слоті лутбоксу, %

Вартість	1 слот	2 слот	3 слот	4 слот	5 слот	Лідер слот
<b>1</b>	<b>100</b>	<b>100</b>	0	0	0	0
<b>2</b>	0	0	<b>78,5</b>	<b>59,5</b>	0	0
<b>3</b>	0	0	<b>20</b>	<b>30</b>	<b>50</b>	0
<b>4</b>	0	0	<b>1,5</b>	<b>10</b>	<b>30</b>	0
<b>5</b>	0	0	0	<b>0,5</b>	<b>20</b>	0
<b>Лідер</b>	0	0	0	0	0	<b>5</b>

Так, у середньому виходить, що за один лутбокс гравець отримує: 2,0 юніта вартості 1; 1,38 юніта вартості 2; 1,0 юніта вартості 3; 0,415 юніта вартості 4; 0,205 юніта вартості 5; 0,05 лідера.

Тоді за формулою ми можемо розрахувати скільки в середньому треба відкрити лутбоксів, щоб отримати хоча б одного юніта певної вартості:

$$L = \frac{1}{P_x},$$

де  $P$  – середня кількість юнітів вартості  $X$  за один лутбокс, що виводить нас на такі результати: 1 лутбокс на юніта 1 вартості; 1,45 лутбоксів на юніта 2 вартості; 1 лутбокс на юніта 3 вартості; 2,41 лутбоксів на юніта 4 вартості; 4,88 лутбоксів на юніта 5 вартості; 20 лутбоксів на юніта-лідера.

Таким чином, у нас є економічна система для коротко-часової утримки гравця. Для ж довгострокової утримки я додав механіку покращення юнітів, яка також має свою економічну складову.

У моєму проєкті юнітам можна покращувати здоров'я та змінювати атаки. Вартість цих послуг можна розрахувати за такими формулами:

$$C_{hp}(v, u) = B_v \times f(u),$$

де  $C_{hp}(v, u)$  – вартість покращення здоров'я юніта вартості ( $v$ ) на покращенні номер ( $u$ );

$B_v$  – базова вартість покращення здоров'я для юніта вартості ( $v$ );

$f(u)$  – множник залежно від раза покращення.

$A$  вартість зміни атаки розраховується за формулою:

$$C_{atk}(v) = k_v \times D,$$

де  $C_{atk}(v)$  – вартість зміни атаки юніта вартості ( $v$ );

$D$  – базова вартість зміни атаки;

$k_v$  – коефіцієнт вартості для кожного рівня юніта.

Зазвичай повне покращення юніта буде включати в себе чотири покращення здоров'я і приблизно одну зміну атаки. Для встановлення економічної системи на покращення необхідно ввести окрему валюту. В випадку мого проєкту я ввів валюту, яку можна отримувати лише за видалення юнітів зі своєї колекції. Назвемо для зручності цю валюту "кристалами". Так я вирішив встановити такий курс конвертації: за юніта 1 вартості – 1 кристал; за 2 вартості – 3 кристали; за 3 вартості – 6 кристалів; за 4 вартості – 15 кристалів; за 5 вартість – 30 кристалів. Ця інформація

необхідна, щоб вирахувати приблизну кількість часу, для повного покращення одного юніта.

Для прикладу припустимо, що сума всіх апгрейдів здоров'я для юніта 5 вартості = 48 кристалів за формулою:

$$C_{hptot}(5) = C_{hp}(5,1) + C_{hp}(5,2) + C_{hp}(5,3) + C_{hp}(5,4) = 48.$$

А нехай  $C_{atk}(5) = 30$  кристалів. Таким чином,  $C_{total}$  (сума  $C_{hptot}$  та  $C_{atk}$ ) = 78 кристалів.

Далі на основі того, скільки юнітів випадає та скільки кристалів отримуємо таку середню кількість кристалів за один лутбокс:

$$K_{серед} = N_1 \times K_1 + N_2 \times K_2 + N_3 \times K_3 + N_4 \times K_4 + N_5 \times K_5 = 24,515,$$

де  $N_1$  – середня кількість юнітів (1) вартості в одному лутбоксі;

$K_1$  – кількість кристалів, які отримуються за конвертацію юніта (1) вартості.

Що на період в 2-3 дні можна оцінити як проміжок 8,17–12,26 кристалів в день. І якщо поділити  $C_{atk}(5)$  на отримані значення, то отримаємо, що на одне повне покращення юніта 5 вартості піде приблизно 6-9,5 днів.

Таким чином, ми отримуємо довгострокову утримку гравця, бо з відкриттям кожного лутбоксу гравець отримує все більше і більше можливостей витратити свій час та ресурси. І все за рахунок грамотній економічній системі та конвертації одного прогресу в інший.

Список використаних джерел:

1. Schell J. The Art of Game Design: A Book of Lenses, Second Edition. A K Peters/CRC Press. 2019.

## ДОДАТОК Г

Концепт-документ до гри

### **“Legions of the Vault” Game Design Documentation**



- 1. Вступ**
- 2. Нарратив**
- 3. Геймплей**
- 4. Референси**
- 5. Аналітика**

Рисунок Г.1 – Сторінка 1 концепт-документу до гри

## 1. Вступ

"Legions of the Vault" — це гра, яка черпає натхнення з двох масштабних жанрів настільних стратегій: варгеймів та колекційних карткових ігор (ККІ).

Цей проект об'єднує найкращі елементи цих жанрів, створюючи новий досвід — новий жанр стратегічних ігор, що може бути настільки ж простим, як шахи, і настільки ж складним, як варгейм.

Це PvP-гра, де ви будете свою армію та намагаєтесь отримати кращі юніти через різні лутбоксы (бустер-паки). Різні юніти можуть належати до різних фракцій та володіти різноманітними здібностями, що допомагають знищити базу суперника.

Теоретично, ця гра також може мати PvE-режим з сюжетними лініями, битвами з босами тощо.

**У висновку – це гра про унікальний досвід та знаходження себе серед безлічі можливих способів гри.**

## 2. Нарратив

**Кількома словами:** це гра, що поєднує безліч всесвітів. Між цими світами існує вежа, у якій на найвищому рівні заховано великий скарб, що може бути здобутий лише тими, хто кине виклик багатьом вправним авантюристам з інших всесвітів.

### Довша версія:

Колись жив чаклун... Ні, вчений! А може, це було космічне створіння? Примара? Бог, можливо?

Зараз вже складно сказати, коли все це перемішалось. Лише небагатьом пощастило зустріти творця, і всі вони розповідають одну і ту саму історію. Вони зустріли істоту, одержиму своїм світом, ні, не лише своїм, а цілою меріадою інших вимірів.

Рисунок Г.2 – Сторінка 2 концепт-документу до гри

Творець міг говорити вічно про свої мандри світами, про людей і створінь, що там мешкали. Іноді він навіть дарував маленькі фігурки тих, про кого розповідав.

Але все змінилося, коли творець раптово зник. Натомість, по всім вимірам почали з'являтися входи до вежі, поєднуючи світи поза межами часу й простору.

Там, у нескінченних коридорах та лабіринтах, ожила спадщина творця. Фігурки, які він залишав під час своїх подорожей, почали оживати і вони діяли так само, як створіння з легенд. Варто було лише розмістити їх на таємничих дошках, що розкидані по поверхах вежі, і все навколо змінювалось, розпочиналась гра за право просування на наступний поверх. Переможці отримували багатства та підтримку вежі, а ті програвші опускались все нижче, поки не повртались до дому.

І так народилася нова легенда. Легенда про найбільший скарб між світами, захований десь у грі. Грі сміливості, удачі та стратегії! Грі, де історії з різних світів оживають, переплітаються та стикаються в запеклих битвах!

Чутки кажуть, що лише ті, хто здобуде ключ від верхнього поверху цього таємничого підземелля, будуть гідними отримати благословення творця і його найбільший скарб...

#### **Чому нарратив це важливо?**

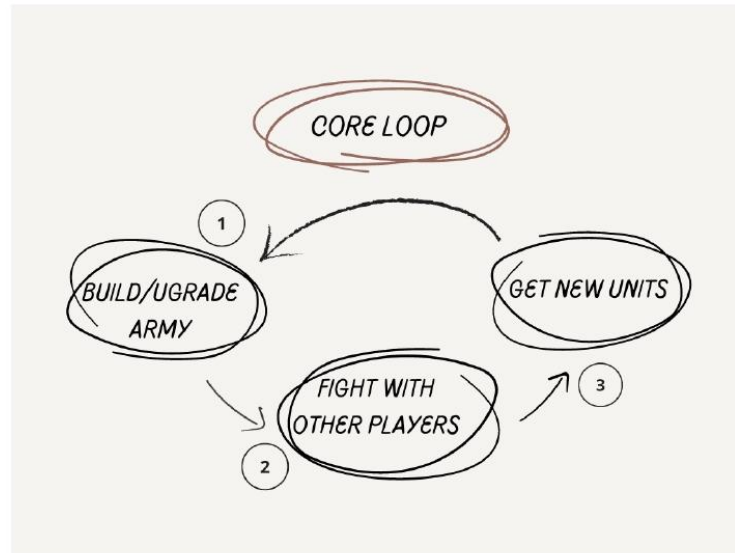
В ідеалі гра має постійно розвивати і додавати нові історії про своїх персонажів та їх світи – від оповідань до синематиків.

Таким чином гра може привабити до себе багато потенційно нових гравців, які ніколи не цікавились жанром. Для постійних гравців історії про їх улюблених юнітів будуть додавати більше шарму та атмосфери.

Рисунок Г.3 – Сторінка 3 концепт-документу до гри

### 3. Геймплей

#### 3.1 Core loop



Ось як виглядає основний ігровий цикл гри. Цей цикл дуже схожий на цикл будь-якої карткової гри. У наступних абзацах я детальніше опишу, як працює кожен елемент циклу та які механіки є унікальними для цієї гри.

#### 3.2 Складання та покращення армії

Спершу, щоб розпочати гру, вам потрібно зібрати свою армію. Деякі юніти будуть доступні на початку гри, інші можна буде отримати в міру проходження гри.

У грі буде представлено різноманіття фракцій, з яких можна обирати юнітів, але деякі лідери будуть обмежувати ваш вибір. Тому першим що ви будете вирішувати – чи будете ви грати лідера, і якщо так, то якого?

Після того, як ви визначилися з лідером і зрозуміли свої обмеження, настав час обрати юнітів. Кожен юніт має **характеристики, здібності, прив'язану стіну** (детальніше про стіни у розділі "*Битва з іншими гравцями*"),

Рисунок Г.4 – Сторінка 4 концепт-документу до гри

**емблему фракції, модель (або арт) та вартість.**

**Вартість юніта** відіграє ключову роль у побудові армії. Для кожної армії передбачено фіксований пул у **30 очок**, які використовуються для найму юнітів. **Армія не може перевищувати цей пул очок, але частина очок може залишатися невикористаною** (аналогічно до побудови армій у [варгеймах](#)).

Коли очки розподілені, а ваша армія готова – ви можете розпочинати гру.

### 3.3 Сутички з іншими гравцями

На цьому етапі ви обираєте режим гри (рейтинговий, нерейтинговий, можливо, PvE), але для цієї документації ми припустимо, що обрано звичайну PvP-гру.

Коли супротивника знайдено, гра розгортається наступним чином:

#### 1. Тераформування.

Це одна з основних механік гри. Як вже згадувалося, кожен юніт має прив'язану до нього стіну, що додає ще один шар стратегічної цінності. Ви обираєте, де і яку стіну розмістити, після чого ваш супротивник робить те саме, поки у вас обох не закінчаться стіни. У кожного гравця є близько 30 секунд на хід, де не витрачений час може бути збережений на наступний хід. Головні умови розташування стін: вони не можуть налазити одна на одну, завжди має зберігатись прохід від однієї сторони дошки, до іншої.

#### 2. Ви та ваш супротивник по черзі використовуєте юнітів.

Одна з основних умов перемоги в грі – знищити базу супротивника або вбити його юніта-лідера, але можуть бути й інші способи виграти, залежно від того, які юніти ви та ваш супротивник обрали, або, можливо, існуватиме додаткова умова перемоги на підземеллі (мапі), на якій ви граєте (на кшталт місії: зібрати, знищити, активувати, контролювати тощо).

У вас буде кілька токенів активації (тестова кількість = 3), які ваші юніти можуть витратити на рух, атаку або активацію здібностей. Вибір найкращого способу витратити ці активації є ключем до перемоги. Тому ви та ваш супротивник по черзі намагатиметесь перехитрити один одного, що нагадує гру в шахи.

### 3. Кінець матчу та отримання нагород

Залежно від результатів вашої партії, ви отримуєте винагороди, які можна використати для розширення вашої колекції.

#### 3.4 Отримання нових юнітів

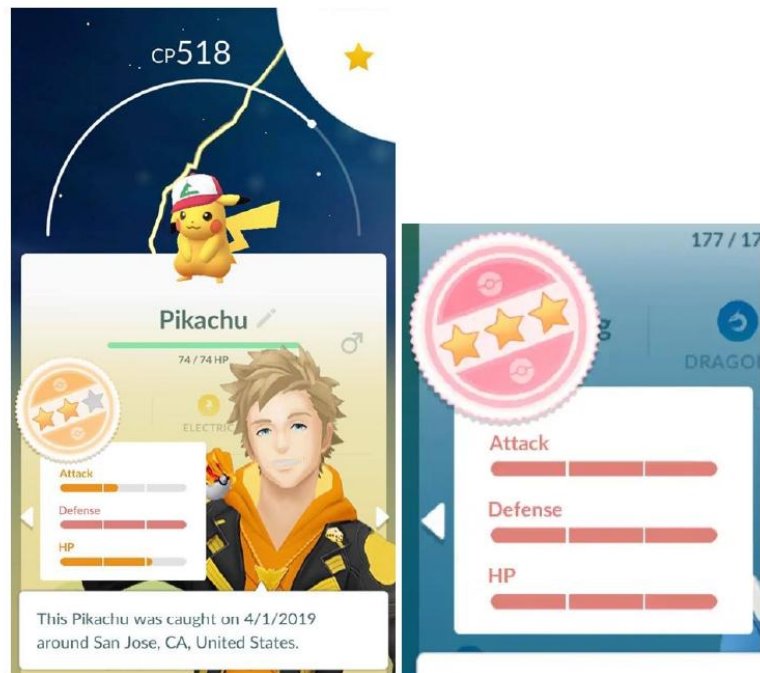
***!Дісклеймер!*** монетизація вказана нижче не є фінальною і може піддатись редагуванням через вимоги платформ.

У грі є магазин з лутбоксами, а також система крафту для отримання конкретних юнітів, які ви хочете (наприклад, як у Hearthstone).

Загалом у гравця буде доступно дві валюти:

1. Гроші – необхідні для покупки лутбоксів, отримуються з системи квестів та нагород.
2. Кристали – необхідні для крафту необхідних юнітів, можна отримати з лутбоксів, нагород та при “знищенні” не потрібних копій юніта.

Систему крафту буде сильно підтримувати факт того, що **у кожного юніта є багато його варіацій**, з різними атаками, стінами, вартістю та характеристиками. **Подібний підхід можна знайти в покемонах:**



У покемонах можна зловити більше 100 пікачу, але всі вони будуть різні, бо від початку будуть мати різні характеристики. На скрінах вище можна побачити шкалу за якою визначається якість характеристики.

### **В чому відмінність нашої системи і як вона працює?**

У кожного юніта є декілька варіацій того що йому може випасти:

- Різні атаки (1-2 штуки зі списку)
- Різні кількості хп
- Різні стіни (зазвичай кількість клітин займає завжди однаково, але форми можуть бути різні)

В результаті від комбінації яка вийде може змінюватись вартість юніта.

Гравцю вигідно збирати різні варіації юнітів, для того щоб використовувати максимум наданих поінтів для збору армії, але іноді він може шукати якусь дуже конкретну його варіацію – тоді гравець може

Рисунок Г.7 – Сторінка 7 концепт-документу до гри

використати кристали для отримання нових складових юніта (змінити атаку, отримати інші характеристики, змінити форму стіни)

Єдина зміна в юніті яку він не може нормально контролювати – отримання нових характеристик, бо кожного разу вони будуть випадкові, навіть після витрати кристалів.

### 3.5 Про юнітів та механіки.

#### 3.5.1 Вартість та характеристики

Вартість юнітів здебільшого через їх характеристики та здібності.

Ідеальним представником кривої вартості є такий юніт:

**Вартість:** 6 поінтів

**Кількість атак:** 2

**Особлива здібність або механіка:** наявна

**Шкода від основної атаки:** 50

**Кількість здоров'я:** 210

Розберемо крок за кроком звідки береться така вартість.

Перше за все, звернемо увагу на здоров'я та шкоду від атаки. При розгляді юніта на кожний 1 поінт у нас видається 20 очок. Як вони працюють? Ви розкидуєте їх між шкодою та здоров'ям, тобто зазвичай 20 очок будуть розподілені на 10 шкоди та 10 здооров'я. Але для підтримання певної динаміки гри здоров'я має бути стандартно вищим, тому воно помножується на 4. Тоді найслабший юніт має в середньому 10 шкоди та 40 здоров'я. (**ВАЖЛИВО:** розрахунки завжди ведуться в десятках, але для зручності ви можете стерти для себе по одному нулю)

Тоді на 6 поінтів має бути 120 очок які б розкинули на 60 шкоди та 240 здоров'я. Але у нас є і інші вигідні характеристики, які змушують зниження середніх показників. Так, якщо придивитись, це схоже на кращі

характеристики юніта на 5 поінтів (50 шкоди 200 здоров'я).

Тоді на що пішли ті 20 очок?

У кожного юніта має бути щонайменше одна дія, яка не витрачає тих очок, але за кожну додаткову дію ми маємо витратити 10 очок. Так як у нас 2 атаки ми знайшли куди пішли 10 очок.

Далі ми бачимо, що у юніта наявна особлива здібність, що також накидує 10 очок.

Тоді все ніби схожиться для 6 поінтів...але здоров'я цього юніта дорівнює 210, звідки взяли 10 додаткових здоров'я?

Це компенсація великої вартості юніта. Чим більше поінтів коштує юнітів, тим більше ймовірність, що у нього буде додаткові 10-20-30 здоров'я, які зможуть йому прожити довше в бою зі схожим юнітом по вартості і отримати перевагу над меншими. Все таки було б доволі нудно, як би будь-хто помирає від десь 4-х своїх же атак, іноді треба додати ще зовсім трохи. Це призводить до більш напружених партій і більшої корисності юнітів, які мають меншу вартість.

Це спрощення варіація моїх стандартних розрахунків, яка не враховує вплив здібностей, стін та механік на баланс гри.

#### 4. Референси

[\*Cardfight Vanguard\*](#): велика різноманітність [архетипів та естетики](#) на вибір, кожен може знайти щось, що відгукнеться саме йому.

[\*Legends of Runeterra\*](#): чудово попрацювали, поєднавши найкраще з Hearthstone та Magic: The Gathering. Але варто врахувати, що вони не були настільки орієнтовані на отримання грошей, як згадані раніше ігри, тому нові картки та інші ресурси здобувати було простіше. Також цікаве рішення обмежити колоду максимум двома регіонами.

[\*Hearthstone\*](#): вміли заробляти на колекційних елементах, але іноді були занадто

жадібними. Однак вони найкраще використовували механіки, які важко чи неможливо реалізувати в реальних карткових іграх.

*Warhammer*: одна з перших варгеймів, але правила можуть бути занадто складними для новачків. Мій проєкт використовує багато подібних механік, таких як побудова армії, але з набагато нижчим порогом входу для нових гравців.

*Шахи*: не потребує представлення.

*Yu-Gi-Oh! Dungeon Dice Monsters*: фентезійна PVP-гра з механікою побудови підземель та великою кількістю кидання кубиків. Мабуть, найтісніший і найвіддаленіший родич проєкту, описаного в цій документації.

## 5. Analytics

Цей абзац, описує загальну інформацію про гру

- **Цільова аудиторія гри:** чоловіки віком 16-35 років (так, віковий діапазон доволі широкий, але це типово для ігор на кшталт Magic: The Gathering).
- **Середній час гри** за одну партію: 20-30 хвилин (досить далеко від швидкого мобільного стилю гри або нескінченних партій у Warhammer).
- **Естетика:** ближче до чогось на зразок *AFK Journey* (не хочу йти в сторону занадто реалістичної чи занадто мультяшної графіки, має бути той самий золотий середняк між ними. Це дозволить гравцям насолоджуватися чимось епічним або милим в одній грі).
- **Основні платформи:** ПК перш за все, а потім мобільні.
- **Складність:** гра має вищий поріг входу, але після пари партій гравець має себе почувати доволі комфортно, але загалом гра має велику можливість заглиблення у її баланс для досвідчених гравців.

**ДОДАТОК Д**  
План тестування

<p><b>ТЕСТ-ПЛАН</b> ПЗП-21-5 Шапошніков Лев Ігорович</p>
<p><b>Назва проекту</b> <b>Ігровий програмний</b> <b>застосунок</b> <b>у жанрі покрокової дуельної</b> <b>стратегії «Legions Of The Vault»</b> <b>(Project Name)</b></p>
<p><b>План випробувань і тестування</b> <b>(Test Plan)</b></p>
<p><b>Версія 1.0</b> <b>(Version 1.0)</b></p>

Рисунок Д.1 – Сторінка 1 плану тестування

**Історія змін (Revision History)**

Дата (Date dd/mm/yyyy)	Версія (Version x.x)	Детальний опис (Description details)	ПІБ Автора (Author name)
09.12.25	Version 1.0	Створення тест-плану	Шапошніков Лев Ігорович

Рисунок Д.2 – Сторінка 2 плану тестування

### Зміст (Table of Contents)

Тестовий план (Test Plan).....	5
1. Вступ (Introduction).....	5
1.1 Мета (Purpose).....	5
1.2 Довідкова інформація (Background).....	5
1.3 Галузь застосування (Scope).....	6
1.4 Визначення проекту (Project Identification).....	7
2. Вимоги до тестування (Requirements for Test).....	8
3. Стратегія тестування (Test Strategy).....	11
3.1 Типи тестування (Testing Types).....	11
3.2 Інструменти (Tools).....	16
4 Ресурси (Resources).....	18
4.1 Ролі (Roles).....	18
4.2 Система (System).....	19
5 Етапи проекту (Project Milestones).....	19
6 Кінцевий продукт (Deliverables).....	20
6.1 Тестова модель (Test Model).....	20
6.2 Тестовий журнал (Test Logs).....	20
6.3 Звіти з дефектів (Defect Reports).....	21
7 Додаток А Задачі проекту (Appendix A Project Tasks).....	21
7.1 План тестування включає:.....	21
7.2 Дизайн випробувань включає:.....	21
7.3 Впровадження випробувань включає:.....	21
7.4 Виконання тесту:.....	22

Рисунок Д.3 – Сторінка 3 плану тестування

7.5 Оцінка випробувань:.....	22
------------------------------	----

Рисунок Д.4 – Сторінка 4 плану тестування

## Тестовий план (Test Plan)

### 1. Вступ (Introduction)

#### 1.1 Мета (Purpose)

Наведений тест-план для ігрового програмного застосунку «Legions of the Vault» описує:

- перелік вимог для проведення випробування покрокової дуельної стратегії;
- рекомендації щодо опису стратегії тестування унікальних механік гри;
- визначення необхідних ресурсів і забезпечення оцінки випробувань системи збірки армії та бойових механік;
- перелік тестових елементів проєкту з урахуванням специфіки PvP-геймплею.

Тестовий план буде використаний для проведення тестування, що базуватиметься на визначених в тест-плані критеріях якості і типах тестування, з особливим фокусом на баланс юнітів та унікальні механіки тераформування.

#### 1.2 Довідкова інформація (Background)

Згідно до сюжету гравці є авантюристами з різних всесвітів, які потрапили до таємничої вежі, створеної невідомим творцем. У цій вежі зберігається найбільший скарб між світами, який можна здобути лише перемагаючи інших авантюристів у стратегічних поєдинках. Гравці використовують оживлі фігурки різних створінь та героїв, розміщуючи їх на таємничих дошках для участі в битвах за право просування на наступний поверх вежі.

Гра «Legions of the Vault» є покроковою дуельною стратегією top-down, що поєднує елементи варгеймів та колекційних карткових ігор. Розроблена для операційної системи Android з підтримкою ПК-версії. Наявні локалізації: англійською та українською мовами.

Рисунок Д.5 – Сторінка 5 плану тестування

Унікальними особливостями гри є система тераформування з розміщенням стін, обмежений пул в 15 очок для збірки армії, система токенів активації та варіативність юнітів з різними атаками, характеристиками та стінами.

### 1.3 Галузь застосування (Scope)

Проведення тестування необхідне для досягнення певних критеріїв якості, визначених тестовим планом для покрокової стратегії. Це допоможе знизити ризики втрати користувача через складність унікальних механік та забезпечити збалансованість ігрового процесу. Зокрема, основними цілями тестування є:

- перевірка виконання вимог до програмного продукту з урахуванням специфіки стратегічного геймплею;
  - підтвердження збалансованості системи вартості юнітів та коректності формули розрахунку характеристик;
  - забезпечення стабільності системи "походил-передав" та правильності роботи токенів активації;
  - підвищення характеристик юзабіліті інтерфейсу збірки армії та бойової системи;
  - перевірка функціонування системи лутбоксів та крафту юнітів.
- Для досягнення зазначених цілей, було прийнято рішення обрати наступні типи тестування:
- функціональне тестування збірки армії та бойових механік;
  - тестування балансу юнітів та системи вартості;
  - тестування інтерфейсу користувача (GUI) з фокусом на стратегічний геймплей;
  - тестування продуктивності з великою кількістю юнітів на полі;
  - тестування системи лутбоксів та монетизації;
  - тестування локалізації;
  - тестування конфігурації та інсталяції.

Рисунок Д.6 – Сторінка 6 плану тестування

## 1.4 Визначення проекту (Project Identification)

Документ і версія / дата (Document and version / date)	Створено або Доступно (Created or Available)	Поступило або Відгук (Received or Reviewed)	Автор або ресурс (Author or Resource)	Примітка (Notes)
Концепція гри (Game Conception)	Так	Так	Game Designer	
Дизайн-документ гри (Game Design Document)	Так	Так	Game Designer	Детальний опис механік
Документ балансу юнітів (Unit Balance Document)	Так	Так	Game Designer	Формули розрахунку
План проекту (Project Plan)	Так	Так	Project Manager	

Рисунок Д.7 – Сторінка 7 плану тестування

## 2. Вимоги до тестування (Requirements for Test)

### Функціональне тестування збірки армії:

- можливість обрати лідера та перевірити його обмеження на фракції;
- можливість додати юніта до армії в межах 15 очок вартості;
- можливість видалити юніта з армії та перерахувати загальну вартість;
- можливість переглянути характеристики юніта (здоров'я, атаку, здібності);
- можливість переглянути форму стіни юніта перед додаванням до армії;
- можливість зберегти готову армію та завантажити її для використання;
- валідація армії на відповідність обмеженням лідера та максимальній вартості;
- можливість переглянути різні варіації одного юніта з різними атаками;
- можливість використовувати невикористані очки армії або залишити їх порожніми;
- перевірка правильності розрахунку вартості юніта за формулою (20 очок на поінт, здоров'я x4).

### Функціональне тестування бойової системи:

- можливість розпочати фазу тераформування та розмістити стіни по черзі;
- валідація розміщення стін (відсутність накладання, збереження проходу);
- можливість розмістити юнітів на дошці після завершення тераформування;
- функціонування системи токенів активації (тестова кількість = 3);
- можливість використати токен на рух, атаку або активацію здібності;
- перевірка умов перемоги (знищення бази противника або лідера);
- правильність розрахунку шкоди та здоров'я під час бою;
- функціонування системи "походив-передав" між гравцями;
- можливість завершити партію та отримати винагороди;

Рисунок Д.8 – Сторінка 8 плану тестування

- збереження стану гри при перериванні з можливістю продовження.

Тестування системи лутбоксів та прогресії:

- можливість відкрити лутбокс за ігрову валюту (гроші);
- можливість отримати кристали з лутбоксів або за знищення копій;
- функціонування системи крафту для отримання конкретних юнітів;
- можливість змінити атаку юніта за кристали;
- можливість змінити форму стіни юніта за кристали;
- генерація випадкових характеристик при крафті;
- правильність нарахування винагород за перемоги та поразки;
- функціонування системи квестів для отримання валюти;
- збереження прогресу колекції юнітів;
- валідація покупок та витрат валют.

Тестування інтерфейсу користувача (GUI):

- перевірити навігацію по екранах збірки армії та колекції;
- перевірити читабельність тексту характеристик юнітів та описів здібностей;
- перевірити відображення прогресу збірки армії (використані/доступні очки);
- перевірити візуалізацію стін юнітів та їх форм;
- перевірити відображення стану бою та токенів активації;
- перевірити анімації руху юнітів та атак на бойовій дошці;
- перевірити інтерфейс магазину лутбоксів та системи крафту;
- перевірити час відгуку на взаємодії з елементами стратегічного інтерфейсу;
- перевірити коректність відображення різних варіацій юнітів;
- перевірити можливість перемикання локалізації інтерфейсу.

Тестування продуктивності та навантажувальне тестування:

- перевірити продуктивність при максимальній армії з 15 юнітів на дошці;
- перевірити продуктивність при складних формах стін та великій кількості перешкод;

Рисунок Д.9 – Сторінка 9 плану тестування

- перевірити продуктивність системи обчислення шляхів руху юнітів;
- перевірити час обробки ходу з множинними токенами активації;
- перевірити продуктивність системи валідації розміщення стін;
- перевірити час завантаження колекції з великою кількістю юнітів;
- перевірити швидкість розрахунку балансу армії та вартості;
- перевірити стабільність при довготривалих партіях (понад 30 хвилин);
- перевірити збереження стану гри при закритті додатку;
- перевірити продуктивність інтерфейсу при перегляді великої колекції.

#### Тестування балансу юнітів:

- перевірити правильність формули розрахунку вартості (20 очок за поінт);
- перевірити баланс між шкодою та здоров'ям (стандартне співвідношення);
- перевірити вплив додаткових атак на вартість юніта (+10 очок за атаку);
- перевірити вплив особливих здібностей на вартість (+10 очок за здібність);
- перевірити компенсацію додаткового здоров'я для дорогих юнітів;
- перевірити збалансованість різних варіацій одного юніта;
- перевірити вплив форми стіни на загальну корисність юніта;
- перевірити обмеження лідерів та їх вплив на армію;
- перевірити можливість створення різноманітних стратегій в межах 15 очок;
- перевірити відсутність домінуючих стратегій або "broken" комбінацій.

#### Тестування конфігурації та інсталяції:

- перевірити працездатність гри на різних версіях операційної системи Android;
- перевірити працездатність ПК-версії на різних конфігураціях;
- перевірити працездатність гри на пристроях з малою кількістю оперативної пам'яті;
- перевірити працездатність гри при перевстановленні зі збереженням

Рисунок Д.10 – Сторінка 10 плану тестування

- прогресу;
- перевірити сумісність збережених даних між версіями гри.

### 3. Стратегія тестування (Test Strategy)

#### 3.1 Типи тестування (Testing Types)

##### 3.1.1 Дані і БД Інтеграційне тестування (Data and Database Integrity Testing)

База даних відсутня, однак в грі наявна можливість збереження прогресу гравця, включаючи колекцію юнітів, налаштування армій та результати партій. Збереження відбувається локально з можливістю синхронізації в хмарі. При цьому зберігаються налаштування різних армій, прогрес в системі квестів та накопичені валюти.

Мета випробування (Test Objective)	Забезпечення методів доступу до збереження та відтворення даних гравця, включаючи колекцію юнітів та прогрес, без пошкодження даних.
Технічний прийом (Technique)	Виклик кожного метода доступу до даних колекції та армій, відбір дійсних та недійсних конфігурацій. Перевірка даних для забезпечення цілісності збережених армій та відповідності обмеженням. Тестування синхронізації між пристроями та відновлення після переустановки.
Критерії завершення (Completion Criteria)	Виконання всіх методів доступу до даних гравця без пошкодження колекції юнітів або налаштувань армій.
Спеціальні рекомендації (Special Consideration)	- Тестування може вимагати імітації різних станів збережених даних. Процеси синхронізації повинні бути перевірені в різних мережевих умовах.

Рисунок Д.11 – Сторінка 11 плану тестування

### 3.1.2 Функціональне тестування (Function Testing)

Мета випробування (Test Objective)	Забезпечити належне тестування функціональності покрокової стратегії, включаючи: збірку армії з урахуванням обмежень, систему тераформування, бойові механіки з токенами активації, взаємодію юнітів та умови перемоги.
Технічний прийом (Technique)	Виконати перевірку кожного сценарію геймплею з використанням дійсних і недійсних конфігурацій армій та дій, щоб перевірити: <ul style="list-style-type: none"> <li>- Очікувані результати збірки армії в межах 15 очок.</li> <li>- Коректність роботи системи тераформування та валідації стін.</li> <li>- Правильність функціонування токенів активації та бойової системи.</li> <li>- Відповідні помилки при порушенні правил гри.</li> </ul>
Критерії завершення (Completion Criteria)	<ul style="list-style-type: none"> <li>- Всі заплановані сценарії геймплею було протестовано.</li> <li>- Усі виявлені дефекти балансу та механік були розглянуті.</li> </ul>
Спеціальні рекомендації (Special Consideration)	Особлива увага повинна приділятися тестуванню унікальних механік: формули вартості юнітів, системи варіацій та взаємодії обмежень лідерів.

### 3.1.3 Бізнес-цикл тестування (Business Cycle Testing)

В проєкті будуть наявні щоденні квести та події за розкладом, мета яких заохочувати гравця регулярно запускати гру та використовувати систему лутбоксів. Зокрема, буде система щоденних винагород за входи в гру та тижневі виклики для отримання додаткових ресурсів.

Мета випробування (Test Objective)	Забезпечити належне тестування системи квестів, щоденних винагород та періодичних подій відповідно до бізнес-моделі гри.
Технічний прийом (Technique)	Тестування буде імітувати кілька циклів активності гравця: <ul style="list-style-type: none"> <li>- Тести щоденних входів та нарахування винагород.</li> <li>- Перевірка тижневих викликів та спеціальних подій.</li> </ul>

Рисунок Д.12 – Сторінка 12 плану тестування

Мета випробування (Test Objective)	Забезпечити належне тестування системи квестів, щоденних винагород та періодичних подій відповідно до бізнес-моделі гри.
	- Тестування системи лутбоксів та монетизації. - Всі функції будуть виконуватися з використанням дійсних і недейсних часових періодів.
Критерії завершення (Completion Criteria)	- Всі заплановані бізнес-цикли було протестовано. - Система монетизації працює коректно без можливості експлоїтів.
Спеціальні рекомендації (Special Consideration)	Важливо перевірити баланс економіки гри та відсутність способів обходу платних механік.

### 3.1.4 Тестування інтерфейсу користувача (User Interface Testing)

Інтерфейс стратегічної гри має бути інтуїтивно-зрозумілим для нових гравців, але достатньо інформативним для досвідчених стратегів. Особливо важливим є зручність збірки армії, читабельність характеристик юнітів та зрозумілість бойового інтерфейсу з токенами активації.

Через складність стратегічних механік, інтерфейс повинен надавати достатньо інформації для прийняття обґрунтованих рішень, але не перевантажувати новачків. Система підказок та навчання має пояснювати унікальні механіки гри.

Мета випробування (Test Objective)	Перевірити наступне: - Зручність навігації по екранах збірки армії та колекції юнітів. - Читабельність та зрозумілість інформації про характеристики та здібності. - Інтуїтивність бойового інтерфейсу та системи токенів активації. - Якість візуалізації стін юнітів та процесу тераформування.
Технічний прийом (Technique)	Створення та виконання тестових сценаріїв для кожного екрану гри, перевірка зручності виконання типових дій гравця та відповідності інтерфейсу потребам стратегічного геймплею.

Рисунок Д.13 – Сторінка 13 плану тестування

Мета випробування (Test Objective)	Перевірити наступне: - Зручність навігації по екранах збірки армії та колекції юнітів. - Читабельність та зрозумілість інформації про характеристики та здібності. - Інтуїтивність бойового інтерфейсу та системи токенів активації. - Якість візуалізації стін юнітів та процесу тераформування.
Критерії завершення (Completion Criteria)	Кожен екран та сценарій взаємодії успішно перевірений, інтерфейс забезпечує комфортний досвід як для новачків, так і для досвідчених гравців.
Спеціальні рекомендації (Special Consideration)	Особлива увага має приділятися тестуванню інтерфейсу на різних розмірах екранів, враховуючи підтримку як мобільних пристроїв, так і ПК.

### 3.1.5 Тестування продуктивності (Performance Profiling)

Мета випробування (Test Objective)	Перевірка продуктивності при типових та граничних умовах стратегічного геймплею: максимальна кількість юнітів на дошці, складні конфігурації стін, довготривалі партії.
Технічний прийом (Technique)	- Використання тестових сценаріїв з максимальним навантаженням. - Тестування обчислення шляхів руху та валідації дій. - Перевірка продуктивності системи балансу та розрахунку вартості. - Моніторинг використання пам'яті при великих колекціях юнітів.
Критерії завершення (Completion Criteria)	Стабільна робота гри при максимальному навантаженні без критичних падінь FPS або зависань.
Спеціальні рекомендації (Special Consideration)	Тестування має проводитися на пристроях з мінімальними системними вимогами для забезпечення доступності гри.

### 3.1.6 Завантажувальне тестування (Load Testing)

Через специфіку покрокової стратегії та відсутність реального мультиплеєру (система "походив-передав"), завантажувальне тестування серверів не є критичним. Основна увага приділяється локальній продуктивності.

Рисунок Д.14 – Сторінка 14 плану тестування

### 3.1.7 Стресове тестування (Stress Testing)

Стресове тестування фокусується на граничних умовах стратегічного геймплею: максимальні армії, найскладніші конфігурації стін, довготривалі партії з великою кількістю ходів.

### 3.1.8 Навантажувальне тестування (Volume Testing)

Мета випробування (Test Objective)	Перевірка стабільної роботи при великих обсягах даних: розширені колекції юнітів, множинні збережені армії, велика кількість завершених партій в історії гравця.
Технічний прийом (Technique)	Використання тестових даних з максимальними обсягами колекцій та збережених конфігурацій, імітація довготривалого використання гри.
Критерії завершення (Completion Criteria)	Стабільна робота всіх систем гри при максимальних обсягах користувацьких даних.
Спеціальні рекомендації (Special Consideration)	Важливо врахувати потенційне зростання колекцій гравців з часом та забезпечити масштабованість.

### 3.1.9 Тестування безпеки і контролю доступу (Security & Access Control Testing)

Тестування безпеки фокусується на захисті збережених даних гравця, валідації покупок в системі лутбоксів та запобіганні можливим експлойтам економіки гри.

### 3.1.10 Тестування відмовостійкості та відновлення (Failover and Recovery Testing)

Перевірка коректного збереження прогресу при несподіваному закритті гри, відновлення стану незавершених партій та цілісності даних колекцій.

Рисунок Д.15 – Сторінка 15 плану тестування

### 3.1.11 Тестування конфігурації (Configuration Testing)

Мета випробування (Test Objective)	Забезпечення коректної роботи гри на різних платформах (Android, PC) та конфігураціях пристроїв, з урахуванням вимог стратегічного геймплею.
Технічний прийом (Technique)	Тестування на мінімальних та рекомендованих системних вимогах, перевірка адаптації інтерфейсу до різних розмірів екранів та платформ.
Критерії завершення (Completion Criteria)	Стабільна робота на всіх цільових конфігураціях з прийнятною продуктивністю та якістю відображення.
Спеціальні рекомендації (Special Consideration)	Особлива увага до оптимізації для мобільних пристроїв з обмеженими ресурсами та адаптації інтерфейсу для ПК.

### 3.1.12 Тестування інсталяції (Installation Testing)

Мета випробування (Test Objective)	Коректне встановлення гри на цільові платформи при різних умовах: <ul style="list-style-type: none"> <li>- Нові установки на чисті пристрої.</li> <li>- Оновлення з попередніх версій зі збереженням прогресу.</li> <li>- Відновлення після переустановки з хмарного збереження.</li> </ul>
Технічний прийом (Technique)	Тестування процесу інсталяції на різних платформах, перевірка збереження та відновлення даних гравця, валідація цілісності файлів гри.
Критерії завершення (Completion Criteria)	Успішна установка та запуск гри на всіх цільових платформах зі збереженням прогресу користувача.
Спеціальні рекомендації (Special Consideration)	Важливо забезпечити надійність синхронізації даних між пристроями та платформами.

### 3.2 Інструменти (Tools)

Для проведення тестування покрокової стратегії будуть використані наступні інструменти:

Рисунок Д.16 – Сторінка 16 плану тестування

Інструмент (Tool)	Постачальник (Vendor / In-house)	Версія (Version)
Управління тестами (Test Management)	Azure DevOps	Microsoft 2023
Відслідковування дефектів (Defect Tracking)	Jira	Atlassian 2023
Функціональне тестування (Functional Testing)	Automation Testing Framework (GAFT/Functional Tests)	Epic Games / In-house
Тестування продуктивності (Performance Testing)	Unreal Insights	Epic Games
Тестування балансу (Balance Testing)	Custom Game Analytics	In-house
Управління проектами (Project Management)	Jira	Atlassian 2023
Тестування інтерфейсу (UI Testing)	Unreal Automation Tool + Functional UI Tests	Epic Games / In-house

Рисунок Д.17 – Сторінка 17 плану тестування

#### 4 Ресурси (Resources)

##### 4.1 Ролі (Roles)

У таблиці нижче наведено кадрове забезпечення для проекту.

Працівник	Рекомендований мінімальний обсяг осіб (повний робочий день)	Конкретні обов'язки або коментарі
Тест-менеджер, менеджер з тестування	1	Забезпечує управління наглядом. Обов'язки: технічна підтримка тестування стратегічних механік; придбання відповідних ресурсів для тестування балансу; забезпечення управлінської звітності щодо якості геймплею
Конструктор тестів	1	Визначення пріоритетів та реалізація тестів покрокової стратегії. Обов'язки: створення плану тестування унікальних механік; генерація тестових моделей для системи збірки армії; оцінка ефективності тестових зусиль балансу юнітів
Тестувальник геймплею	2	Виконання тестів стратегічних механік. Обов'язки: виконання тестів збірки армії та бойової системи; журнал результатів тестування балансу; документування дефектів механік та інтерфейсу; тестування різних стратегій та комбінацій юнітів
Тестувальник системи	1	Технічне тестування та перевірка продуктивності. Обов'язки: тестування продуктивності при максимальних армія; перевірка системи збереження та синхронізації; тестування інсталяції на різних платформах
Тестовий системний адміністратор	1	Забезпечує тестове середовище та управління активами. Обов'язки: адміністрування тестової системи управління; встановлення та управління доступом до тест-системи;

Рисунок Д.18 – Сторінка 18 плану тестування

Працівник	Рекомендований мінімальний обсяг осіб (повний робочий день)	Конкретні обов'язки або коментарі
		підтримка тестових конфігурацій для Android та PC

#### 4.2 Система (System)

Ресурси	Назва/тип
Клієнт випробувань ПК	DeshevykhArtemPC
Тестові Android пристрої	Iqoo Neo 10
Репозиторій тестування	<a href="https://artemdeshevykh@bitbucket.org/legions-of-the-vault/legions_of_the_vault.git">https://artemdeshevykh@bitbucket.org/legions-of-the-vault/legions_of_the_vault.git</a>
Система багтрекінгу	Jira – Atlassian 2023
Інструменти тестування	Azure DevOps, Unreal Automation Tools

#### 5 Етапи проекту (Project Milestones)

Цільове завдання	Обсяг робіт	Дата початку	Дата закінчення
План випробувань	15	09.12.25	12.12.25
Тест-дизайн механік гри	20	13.12.25	17.12.25
Реалізація випробувань	50	18.12.25	28.12.25
Виконання тестів балансу	40	29.12.25	10.01.26
Виконання функціональних тестів	60	11.01.26	25.01.26
Тестування продуктивності	25	26.01.26	31.01.26
Оцінка випробувань	15	01.02.26	05.02.26

Рисунок Д.19 – Сторінка 19 плану тестування

## 6 Кінцевий продукт (Deliverables)

### 6.1 Тестова модель (Test Model)

В ході виконання тестування будуть отримані такі елементи:

а) план тестування – опис цілей та стратегій тестування покрокової стратегії, методів реалізації процесу тестування унікальних механік гри;

б) тест-кейси збірки армії – документи, що відповідають тестовим сценаріям системи збірки армії в межах 15 очок. Містять: унікальний номер, опис механіки, кроки відтворення, пріоритет, важливість для балансу гри та очікуваний результат;

в) тест-кейси бойових механік – сценарії тестування системи трансформації, токенів активації та взаємодії юнітів на бойовій дошці;

г) тест-кейси балансу – спеціалізовані тести для перевірки формул розрахунку вартості юнітів та збалансованості різних стратегій;

д) тестовий журнал – запис всіх дій пов'язаних з проведенням тестування стратегічних механік, послідовності операцій та результатів тестування геймплею;

е) баг-репорти – звіти про знайдені недоліки в системі збірки армії, бойових механіках та балансі гри з зазначенням рівня серйозності для геймплею.

### 6.2 Тестовий журнал (Test Logs)

Ведення тестового журналу буде відбуватися з використанням:

а) Azure DevOps для управління тестовими сценаріями стратегічних механік;

б) Jira для відслідковування дефектів балансу та геймплею;

в) Excel для аналізу статистики тестування різних конфігурацій армій;

г) Unreal Insights для журналювання тестів продуктивності.

Рисунок Д.20 – Сторінка 20 плану тестування

### 6.3 Звіти з дефектів (Defect Reports)

Звіти з дефектів будуть створені з використанням:

- а) Jira для централізованого управління баг-репортами;
- б) Excel для аналізу статистики дефектів за категоріями (баланс, механіки, UI);
- в) Azure DevOps для інтеграції дефектів з процесом розробки;
- г) спеціалізовані шаблони для дефектів балансу юнітів та стратегічних механік.

### 7 Додаток А Задачі проекту (Appendix A Project Tasks)

Нижче наводяться завдання тесту покрокової стратегії:

#### 7.1 План тестування включає:

- а) визначення вимог тестування стратегічних механік та системи балансу;
- б) оцінку ризиків для геймплею та збалансованості;
- в) розробку стратегії тестування унікальних механік гри;
- г) визначення тест-ресурсів для покрокової стратегії;
- д) створення графіку тестування з урахуванням специфіки геймплею;
- е) створення плану тестування збірки армії та бойових механік.

#### 7.2 Дизайн випробувань включає:

- а) підготовку аналізу навантаження для максимальних конфігурацій армій;
- б) визначення та опис тестів системи тераформування та токенів активації;
- в) визначення тестових процедур для перевірки балансу юнітів;
- г) огляд та оцінку покриття тестування всіх стратегічних механік.

#### 7.3 Впровадження випробувань включає:

- а) створення автоматизованих сценаріїв для тестування балансу;
- б) визначення тестових функцій для системи збірки армії;
- в) встановлення тестових наборів даних з різними конфігураціями юнітів;

Рисунок Д.21 – Сторінка 21 плану тестування

г) підготовку тестових армій для перевірки різних стратегій.

#### 7.4 Виконання тесту:

- а) виконання процедур тестування стратегічних механік;
- б) оцінку виконання тестів балансу та геймплею;
- в) перевірку результатів взаємодії юнітів та системи вартості;
- г) дослідження неочікуваних результатів в механіках гри;
- д) журнал дефектів з фокусом на баланс та зручність гри.

#### 7.5 Оцінка випробувань:

- а) оцінку покриття тестування всіх стратегічних механік;
- б) оцінку покриття тестування різних комбінацій юнітів та стратегій;
- в) аналіз дефектів балансу та їх впливу на геймплей;
- г) визначення критеріїв завершення для збалансованої та стабільної гри;
- д) оцінку досягнення критеріїв успіху для покрокової стратегії.

Рисунок Д.22 – Сторінка 22 плану тестування

## ДОДАТОК Е

### Фрагменти программного коду

#### 3.1 Код системы завершения хода

```

#include <vector>
#include <memory>

class Hero;
class MovementCard;
class StatusEffect;

class TurnBasedPlayerController {
private:
    class Grid* targetGrid;
    bool isPreparationPhase;
    int currentPlayerNumber;
    int turnCount;
    std::vector<int> actedUnits;
    int actionsRemaining;
    class WBPPlacementUnitCards* placementCards;
    Hero* friendlyHero;
    Hero* newHero;
    Hero* selectedHero;

public:
    void ZeroOutReferences() {
        friendlyHero = nullptr;
        newHero = nullptr;
        selectedHero = nullptr;
    }

    void RestoreActionsApplyEffects() {
        std::vector<Hero*> allHeroes = GetAllHeroes();

        for (Hero* hero : allHeroes) {
            if (hero && !hero->IsDead()) {
                RestoreActions(hero);
                UpdateActionPoints(hero);
                SetHealth(hero);
                SetVisibility(hero);
                UpdateStatusEffects(hero);
            }
        }

        HighlightMesh();
    }

    void EndTurn() {
        if (!isPreparationPhase) {

```

```

        CheckAllCardPlaced();

        if (AllPlaced()) {
            ExecuteBranch();
        } else {
            ShowWarningMessage();
            return;
        }
    }

    HighlightStartTiles();
    UpdateTurnCount();
    EndTurnSequence();
    ClearHighlightedTiles();
}

void InitializeController() {
    SetPreparationPhase(false);
    SetActionsRemaining(3);

    if (currentPlayerNumber == 1) {
        SwitchToPlayer2();
        SetCurrentPlayerNumber(2);
    } else if (currentPlayerNumber == 2) {
        SwitchToPlayer1();
        SetCurrentPlayerNumber(1);
    }

    SetCurrentPlayerNumber(currentPlayerNumber);
}

void FillPlacementCards() {
    if (!isPreparationPhase) {
        placementCards->FillCards();
        ZeroOutReferences();

        ChangeIndexUI();
        RestoreActionsApplyEffects();
        UpdateActivationPoints();

        SetActedUnits(std::vector<int>());
    }
}

private:
    std::vector<Hero*> GetAllHeroes() {
        return std::vector<Hero*>();
    }

    void RestoreActions(Hero* hero) {
}

```

```
void UpdateActionPoints(Hero* hero) {  
    }  
void SetHealth(Hero* hero) {  
    }  
void SetVisibility(Hero* hero) {  
    }  
void UpdateStatusEffects(Hero* hero) {  
    }  
void HighlightMesh() {  
    }  
void CheckAllCardPlaced() {  
    }  
bool AllPlaced() {  
    return true;  
}  
void ExecuteBranch() {  
    }  
void ShowWarningMessage() {  
    }  
void HighlightStartTiles() {  
    if (targetGrid) {  
        targetGrid->HighlightTiles();  
    }  
}  
void UpdateTurnCount() {  
    turnCount++;  
}  
void EndTurnSequence() {  
    }  
void ClearHighlightedTiles() {  
    if (targetGrid) {  
        targetGrid->ClearHighlights();  
    }  
}
```

```

    }
}

void SetPreparationPhase(bool phase) {
    isPreparationPhase = phase;
}

void SetActionsRemaining(int actions) {
    actionsRemaining = actions;
}

void SwitchToPlayer1() {

}

void SwitchToPlayer2() {

}

void SetCurrentPlayerNumber(int playerNum) {
    currentPlayerNumber = playerNum;
}

void ChangeIndexUI() {

}

void UpdateActivationPoints() {

}

void SetActedUnits(const std::vector<int>& units) {
    actedUnits = units;
}
};

```

### 3.2 Код системи завершення гри

```

#include "Engine/Engine.h"
#include "Components/WidgetComponent.h"
#include "Blueprint/UserWidget.h"
#include "GameFramework/PlayerController.h"
#include "Components/TextBlock.h"

void EndGameFunction()
{
    TArray<UUserWidget*> foundWidgets;
    UWidgetBlueprintLibrary::GetAllWidgetsOfClass(GetWorld(),
foundWidgets, UUserWidget::StaticClass(), true);

    for (UUserWidget* widget : foundWidgets)
    {

```

```

        if (widget && widget->IsInViewPort())
        {
            UUserWidget*          endGameWidget          =
CreateWidget<UUserWidget>(GetWorld()->GetFirstPlayerController(),
WBPEndGameClass);
            if (endGameWidget)
            {
                APlayerController*  playerController    =
GetWorld()->GetFirstPlayerController();
                if (playerController)
                {
                    UTextBlock*      textBlock          =
Cast<UTextBlock>(endGameWidget->GetWidgetFromName(TEXT("WinText")));
                    if (textBlock)
                    {
                        FString winnerText = FString("Player ")
+ FString("Winner!");
                        textBlock-
>SetText(FText::FromString(winnerText));
                    }

                    endGameWidget->AddToViewPort();

                    FInputModeUIOnly inputMode;
                    inputMode.SetWidgetToFocus(endGameWidget-
>TakeWidget());

                    inputMode.SetLockMouseToViewPortBehavior(EMouseLockMode::LockAlways)
;

                    playerController->SetInputMode(inputMode);
                    playerController->SetShowMouseCursor(true);
                }
            }
            break;
        }
    }
}

```

**ДОДАТОК Ж**

Фрагмент каталогу виставки технічної творчості молоді

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ**

**МАТЕРІАЛИ  
XXIX МІЖНАРОДНОГО МОЛОДІЖНОГО ФОРУМУ**

**«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ  
У XXI СТОЛІТТІ»**

16 – 19 квітня 2025 р.

**КАТАЛОГ ВИСТАВКИ  
ТЕХНІЧНОЇ ТВОРЧОСТІ МОЛОДІ**

Харків 2025

Рисунок Ж.1 – Обкладинка збірника

## 9. Ігровий проект “Dungeon Chess”

**Автори:** *Шапошніков Лев Ігорович*, ст. гр. ПЗПІ-21-5, *Дешевих Артем Миколайович, Pokora Mikita Іванович*, ст. гр. ПЗПІ-21-9, ХНУРЕ.

**Науковий керівник:** Новіков Юрій Сергійович, старший викладач, каф. ПІ, ХНУРЕ.

Розроблено PvP-стратегію, що поєднує елементи варгеймів і колекційних карткових ігор. У представленому прототипі гравці командують арміями у дуельному форматі, намагаючись знищити ворожого лідера та вибудувати найефективнішу тактику.

Геймплей зосереджений на стратегічному розміщенні юнітів, управлінні ресурсами та прийнятті тактичних рішень у кожному ході. Система балансу юнітів була математично розрахована, що дозволяє досягти справедливих ігрових умов та різноманіття бойових сценаріїв. Юніти мають великий різновид можливих дій, а лідери впливають на обмеження та підсилення кращих сторін армій, що створює умови для детального обдумування своїх ходів.

Нашою метою є створення глибокої, але доступної стратегічної гри, що дозволить гравцям відчувати задоволення від тактичного планування та динамічних битв.

## 10. Українські будні

**Автори:** *Кльонон Артем Сергійович, Колосовський Артем Сергійович*, ст. гр. ІТУ-24-1, ХНУРЕ.

**Науковий керівник:** Цвіркун Олександр Анатолійович, асист, каф. ІУС, ХНУРЕ.

Ця гра – динамічний інтерактивний проєкт, у якому гравцеві належить керувати літаком, маневруючи у повітряному просторі, ухиляючись від атакуючих ракет і збираючи бонуси, що приносять додаткові очки. Основна мета – вижити якомога довше, намагаючись набрати максимальну кількість балів. Чим довше літак залишається в повітрі, тим складнішими стають виклики: ракети стають швидшими, їх кількість зростає, а простір для маневру звужується. Гравець повинен швидко реагувати на загрози, продумувати траєкторію руху і використовувати зібрані бонуси для отримання переваги.

Гра створена на основі Python із використанням бібліотеки Pygame, що дозволяє реалізувати плавну анімацію, чітке керування та цікаві механіки. Під час розробки було приділено особливу увагу балансу між складністю та захопливістю геймплею, щоб кожен користувач, незалежно від досвіду, міг насолоджуватися грою і водночас відчувати виклик. Візуальний стиль ігрового середовища виконаний у мінімалістичному стилі, що дозволяє зосередитися на головному – виживанні та досягненні найкращого результату.

Проєкт створювався не лише як технічне завдання, а й як спосіб підтримати українців у складні часи. Гра може стати засобом для короткого відволікання від повсякденних турбот, нагадуючи про важливість витривалості, швидкості прийняття рішень і здатності адаптуватися до обставин. Крім того, розробка цього проєкту стала чудовою можливістю для вдосконалення навичок програмування та вивчення особливостей створення ігрових механік у Python. Сподіваємося, що наш застосунок принесе гравцям задоволення, викличе азарт і, можливо, надихне когось на створення власних ігрових світів.