

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет комп'ютерних наук
Кафедра програмної інженерії

ЗВІТ
З ПЕРЕДАТЕСТАЦІЙНОЇ ПРАКТИКИ

Місце проходження практики:
ХНУРЕ, кафедра ПІ, м.Харків
у період з «7» квітня по «18» травня 2025 р.

Тема індивідуального завдання:
Ігровий програмний застосунок у жанрі покрокової дуельної стратегії. Ігрові механіки та UI/UX.

Студент гр. ПЗПІ-21-9 _____ (Артем Дешевих)
(підпис)

Керівник практики _____ (ст. викл. В. Сковороднікова)
(підпис)

Роботу захищено з оцінкою _____ «17» травня 2025 р.

Комісія:

_____ (доц. О. Мазурова)
(підпис)

_____ (проф. А. Руткас)
(підпис)

_____ (ст. викл. В. Сковороднікова)
(підпис)

Харків 2025

РЕФЕРАТ / ABSTRACT

Звіт з передатестаційної практики: 42 стор., 8 рис., 7 джерел.

ПОКРОКОВА ДУЕЛЬНА СТРАТЕГІЯ, ІГРОВІ МЕХАНІКИ, UI/UX, ПРОЦЕДУРНА ГЕНЕРАЦІЯ РІВНІВ, BLUEPRINTS, C++, UNREAL ENGINE 5, VISUAL STUDIO, ADOBE PHOTOSHOP.

Об'єкт дослідження — програмний ігровий продукт у жанрі покрокової дуельної стратегії з акцентом на розробку механік бою та інтерфейсу користувача.

Мета роботи — розробити та реалізувати прототип гри, що поєднує глибокі тактичні дуелі в структурі покрокового протистояння з інтуїтивно зрозумілим UI/UX.

Методика виконання — використання середовища Unreal Engine 5 разом із Visual Studio 2022; програмування на C++ та створення логіки через Blueprints; оформлення графічних матеріалів у Adobe Photoshop.

Результатом дослідження став ігровий застосунок, що демонструє базові бойові механіки покрокових дуелей, повноцінний користувацький інтерфейс та адаптивний дизайн ігрових елементів.

TURN-BASED DUEL STRATEGY, GAME MECHANICS, UI/UX, PROCEDURAL LEVEL GENERATION, BLUEPRINTS, C++, UNREAL ENGINE 5, VISUAL STUDIO, VISUAL STUDIO, ADOBE PHOTOSHOP.

Object of study — a software game product in the genre of turn-based duel strategy, with a focus on designing combat mechanics and user interface.

Purpose of the work — to design and implement a prototype that merges deep tactical duels within a turn-based framework with an intuitive and accessible UI/UX.

Methodology — development carried out in Unreal Engine 5 and Visual Studio 2022; core logic coded in C++ and Blueprints; graphical assets prepared in Adobe Photoshop.

Outcome — a playable game application showcasing foundational turn-based duel mechanics, a full-featured user interface, and adaptive design of in-game elements.

ЗМІСТ

Вступ.....	5
1 Опис підприємства	6
2 Аналіз предметної галузі	12
2.1 Аналіз предметної галузі	12
2.2 Виявлення проблем та актуалізація рішень	13
2.3 Цільова аудиторія	15
2.4 Виявлення проблем.....	16
2.5 Постановка задачі.....	17
3 Формування вимог до програмної системи.....	19
3.1 Постановка мети.....	19
3.2 Загальний опис	19
3.3 Загальні обмеження	21
3.4 Припущення та залежності	22
4 Архітектура та проектування програмного забезпечення	24
4.1 UML проектування ПЗ.....	24
4.2 Реалізація ігрового поля на основі клітинної сітки	25
4.3 Проектування дизайну рівня.....	29
Висновки.....	31
Перелік джерел посилання	32
Додаток А. Концепт документ до гри	33

ВСТУП

Інтерактивна індустрія стрімко розвивається, породжуючи нові формати й трансформуючи класичні жанри. Серед них особливу увагу здобувають покрокові стратегії, що поєднують глибокий тактичний аналіз із компактним протистоянням двох гравців.

У таких проєктах ключовими є продумана система бою, баланс механік активацій та переміщень, а також адаптивний інтерфейс, який допомагає гравцю швидко зорієнтуватися в ходах суперника й власних можливостях. Зручність UI/UX у покрокових дуелях не менш важлива, ніж самі алгоритми розподілу дій — від чистоти подання інформації залежить стратегічна глибина та швидкість прийняття рішень.

Метою цього дослідження є створення прототипу комп'ютерної гри в жанрі покрокової дуельної стратегії з акцентом на бойові механіки й дружній інтерфейс. Для реалізації проєкту будуть застосовані Unreal Engine 5 та Visual Studio 2022, програмна логіка реалізується мовою C++ та через Blueprints, а графічні матеріали готуються в Adobe Photoshop.

В результаті роботи планується провести аналіз існуючих рішень у цьому жанрі, сформулювати технічні й геймдизайнерські вимоги, а також підготувати повний геймдизайн-документ, що ляже в основу подальшої розробки та тестування ігрового застосунку.

1 ОПИС ПІДПРИЄМСТВА

Кафедра програмної інженерії готує студентів за трьома ступенями вищої освіти: бакалаврів за спеціальністю 121 «Інженерія програмного забезпечення» (освітня програмою «Програмна інженерія»); магістрів за спеціальністю 121 «Інженерія програмного забезпечення» (за освітньо-науковою програмою «Інженерія програмного забезпечення»); докторів філософії – за спеціальністю 121 «Інженерія програмного забезпечення». Навчальні плани узгоджено з міжнародними стандартами підготовки програмістів.

Форми навчання: очна, заочна та заочно-дистанційна.

Студенти мають широкі можливості вільного вибору дисциплін. Деякі предмети викладаються англійською мовою в окремо сформованих групах.

Починаючи з 3 курсу студенти проходять практику в провідних ІТ-компаніях Харкова та України. Договір про співробітництво між кафедрою ПІ ХНУРЕ та компаніями Global Logic, EPAM Systems, NIX Solutions та Sigma Software дозволив створити на кафедрі лабораторії з розробки сучасного програмного забезпечення. Відкрито академічний бізнес-інкубатор «YEP!» та YEP!-клуб, які на базі ХНУРЕ є майданчиком для підтримки студентських ініціатив та стартапів.

Практикується залучення провідних спеціалістів фірм до викладання професійно-орієнтованих дисциплін, здійснюється цільова підготовка студентів старших курсів у формі безкоштовних факультативних занять від компаній.

Під час навчання студенти підвищують свій професійний рівень, беруть участь у роботі семінарів і гуртків, організованих на базі наукових лабораторій. Так, у гуртку «Програміст» студенти здобувають знання та навички з виконання завдань з програмування на різних олімпіадах, у лабораторії GameDev навчають студентів розробляти ігрові проекти, а в проекті «IT Talents» поглиблено вивчають складні алгоритми з практичним їх застосуванням у різних галузях ІТ.

Неодноразово студенти кафедри ІІІ посідали призові місця на українських та міжнародних змаганнях з програмування, гідно представляли кафедру та університет на виставках, конкурсах, конференціях та форумах.

У галузі освіти та науки кафедра ІІІ співпрацює з Ліннеус університетом м. Вакхо (Linnaeus University), Швеція, а також з Університетом Економіки (WSG) м. Бидгощ, Польща. Студенти мають можливість отримати два дипломи – український і європейський рівнів бакалавра або магістра, а також проходити стажування та брати участь у літніх і зимових школах з програмування.

Основні наукові дослідження кафедри виконуються за напрямками:

- створення нового покоління обчислювальних систем і технологій;
- педагогіка в нових освітніх середовищах, дистанційне навчання;
- проектування систем штучного інтелекту;
- розробка математичних моделей механізмів людського інтелекту;
- розробка формального апарату методів логіки, алгебри, лінгвістичної алгебри і системи логічної підтримки проектування нових інформаційних технологій;
- сучасні технології інтеграції гетерогенних розподілених джерел даних і парадигма якості програмного забезпечення автоматизованих систем обробки інформації;
- програмні засоби автоматизованого формування інформаційного простору навчального процесу;
- інформаційні технології дистанційного навчання та електронної комерції; побудова, розробка і реалізації білінгових систем;
- інтелектуальний аналіз даних;
- розвиток основ теорії сегментації та ідентифікації геометричних об'єктів у режимі реального часу для прикладних завдань обробки цифрової інформації;
- семантичний аналіз зображень;

- розробка моделей, методів і алгоритмів розпізнавання для біометричних систем;
- знання-орієнтовані технології класифікації, діагностики і прогнозування ситуацій;
- розробка підсистем аналізу зображень для системи обробки й аналізу технічної інформації в галузі медичного прогнозування;
- розробка програмного забезпечення для систем відеоконтролю;
- розробка моделі навчання і програмного середовища для проведення навчання та перевірки знань у довільній предметній галузі;
- розробка систем мобільного навчання.

Кафедра програмної інженерії існує вже 55 років, хоч і не завжди носила цю назву. На кафедрі працюють більше 60 викладачів, у числі яких 8 докторів наук, 14 професорів, 31 доцент, 39 кандидатів наук.

За час свого існування кафедра підготувала більше 500 кандидатів та докторів наук.

Завідує кафедрою Смеляков К.С., член НТР, заступник голови секції 2 НТР, доктор технічних наук, професор.

У 1963 році Харківський гірничий інститут було реорганізовано у Інститут гірничого машинобудування, автоматики та обчислювальної техніки. У цей період створюється кафедра автоматики та телемеханіки, завідувачем якої стає Іванченко Є.Я.

У цьому ж році кафедра автоматики та телемеханіки була перетворена у кафедру промислової електроніки та обчислювальної техніки, завідувачем був назначений доцент Волков А.А. Відокремившись від цієї кафедри, згідно з наказом №62 від 20 вересня 1963 р. ХІГМАОТа створюється кафедра Обчислювальної техніки.

Першим завідувачем кафедри став Рвачов Володимир Логвинович. Разом із ним на кафедру прийшли працювати Шабанов-Кушнарєнко Юрій Петрович, Мурашко Анрі Гаврилович, Шклярєв Леонід Йосипович, Путятін Євген Петрович

та Тищенко Володимир Володимирович. Кафедра Обчислювальної техніки здійснювала підготовку інженерів зі спеціальності Математичні та лічильно-вирішальні прилади та пристрої. У науковому плані кафедра займалася розробкою засобів обчислювальної техніки (ОТ), нових методів обчислень, математичним моделюванням властивостей зору людини..

У 1964 г. стався перший розподіл кафедри ОТ. З'являється кафедра Математичного програмування та моделювання та кафедра Математичних машин.

У 1966 г. кафедра Математичного програмування та моделювання розділилася на кафедри Математичного моделювання (ММ) та кафедру Обчислювальної математики. Завідуючим кафедри ММ стає Шабанов-Кушнарєнко Ю.П. У цей час поширюється тематика математичного моделювання психічних функцій людини та приходять на кафедру такі фахівці як Дюбко Геннадій Федорович, Качко Олена Григорівна, Марченко Юрій Сергійович, Ловицький Володимир Адріанович.

Починаючи з 1969 року кафедра Математичного моделювання здійснює підготовку спеціалістів зі спеціальності Прикладна математика, спеціалізація – математичне забезпечення ЕОМ і АСУ.

У 1975 р. наказом №86 від 15 травня 1975 р. ХІРЕ кафедру Математичного моделювання перейменовано у кафедру Програмного забезпечення ЕОМ (ПЗ ЕОМ). Період 1969-1984 рр. можна охарактеризувати як період розвитку нових наукових напрямків на кафедрі: розпізнавання зорових образів, реалізація біонічних моделей засобами обчислювальної техніки, паралельні обчислення. З'являються такі кафедри як кафедра Застосування ЕОМ та кафедра Обчислювальної техніки.

З 1974 по 1984 г. кафедрою завідував Мурашко А.Г.

З 1984 по 1987 г. кафедрою завідував Дюбко Г.Ф.

У 1987 р. знову відбувається реорганізація. Кафедри ПЗ ЕОМ та ОТ об'єднуються під назвою ПЗ ЕОМ, завідувачем кафедри стає Бондаренко Михайло

Федорович. У 1994 році Бондаренко М.Ф. був назначений ректором ХІРЕ, і його замісником і вірним помічником стала Дудар Зоя Володимирівна.

У 2001 г. за ініціативою ХНУРЕ створюється Українська асоціація дистанційної освіти, Михайло Федорович стає її президентом. При центрі післядипломної освіти (директор Дудар Зоя Володимирівна) кафедрою вперше в університеті розпочато підготовку кадрів по дистанційній формі навчання (спеціальність ПЗАС), за ініціативою кафедри (Каук В.І.) у 2002 р. в ХНУРЕ створюється Центр технологій дистанційної освіти.

У 2003 р. знову відбувається розділення кафедри: з'являється кафедра Соціальної інформатики, що влилася у напрямок підготовки Прикладна математика, завідувачою якої стала Соловйова Катерина Олександрівна.

У теперішній час кафедра займається як науковою діяльністю, так і готує бакалаврів за напрямком «Програмна інженерія» (ПІ), магістрів та спеціалістів зі спеціальності «Інженерія програмного забезпечення» (ПЗ) та «Програмне забезпечення систем» (ПЗС). Напрямок «Програмна інженерія» - єдиний напрям у ХНУРЕ, який повністю відповідає європейському стандарту Software Engineering.

З 11 грудня 2011 кафедра ПЗ ЕОМ була перейменована у кафедру Програмної інженерії.

Ось вже понад 15 років студенти кафедри беруть активну участь у командних та особистих першостях та олімпіадах з програмування (тренер – Вечур О.В.), де часто стають переможцями, або займають призові місця: півфінал України (Донецьк, Харків), фінал України (Вінниця), Кубок України з програмування (м.Одеса), півфінал Світу з програмування (Румунія, м. Бухарест), відкрита особиста командна першість Південного Кавказу (Грузія, м. Батумі).

Щороку кафедра Програмної інженерії ХНУРЕ запрошує на навчання велику кількість абітурієнтів, з першого дня занурюючи їх у атмосферу професіоналізму та невпинного самовдосконалення. Щоб відповідати власним високим стандартам, кафедра залучає провідних спеціалістів до викладання професійно-орієнтованих

дисциплін, організовує факультативи на базі наукових лабораторій та надає безліч інших можливостей підвищити свій професійний рівень.

Кафедра Програмної інженерії готує програмістів з 1988 року. Впродовж історії розвитку спеціальність мала декілька поступово змінених назв – ПЗОТАС – ПЗАС – ПІ – ПІЗ.

Інженерія програмного забезпечення – це інженерна дисципліна, яка пов'язана зі всіма аспектами виробництва програмного забезпечення від початкових стадій створення специфікації до підтримки системи після здачі в експлуатацію.

Індустрія програмування – одна з найбільш перспективних і динамічних галузей. Сучасні ІТ- компанії забезпечують цікаві проекти, професійне зростання та гідну зарплату. Прогнози економічного розвитку галузі в нашій країні стримуються дефіцитом кваліфікованих кадрів.

Під час навчання студенти опановують різні технологічні підходи до інженерії програмного забезпечення, платформи, операційні системи, середовища та мови програмування для систем різного призначення. Опановують навички роботи в команді та навчаються володіти інструментами колективної розробки програмного забезпечення.

Але якнайкраще про якість здобутої освіти говорять професійні успіхи й досягнення її випускників. Вони – успішні та висококваліфіковані спеціалісти у найрізноманітніших галузях комп'ютерних наук, що мають попит на Батьківщині та за кордоном.

2 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

2.1 Аналіз предметної галузі

Індустрія стратегічних ігор демонструє стійке зростання: у 2024 році світовий ринок стратегій оцінювався приблизно в USD 18 млрд і прогнозується досягнути USD 35 млрд до 2033 року з середньорічним темпом зростання 7,8 %[1]. Серед піджанрів особливу популярність здобувають покрокові тактичні ігри (turn-based tactics), які надають гравцям можливість глибокого стратегічного аналізу кожного ходу.

Зокрема, дуельні формати для двох гравців (duel strategy) поєднують елементи карткових і настільних ігор із класичними тактичними механіками, акцентуючи увагу на балансі між позиціюванням фігур і ресурсним менеджментом. Цей підхід створює великий спектр стратегічних варіантів і повторюваності матчів, що підвищує залучення аудиторії та потенціал монетизації через колекційні механіки й мікротранзакції.

Процедурна генерація рівнів у стратегічних проєктах дозволяє значно розширити ігровий контент та збільшити реіграбельність, знижуючи витрати на ручне створення карт і сценаріїв. Методи генерації — від алгоритмів клітинкових автоматів до графових підходів — застосовуються для випадкового компонування ландшафтів, розташування перешкод і бонусів, що створює унікальні виклики в кожній сесії.

Важливим аспектом успішності покрокових дуелей є якість UI/UX: інтерфейс має балансувати між наданням повної інформації для прийняття рішень та мінімізацією когнітивного навантаження на гравця. Удосконалені елементи зворотного зв'язку — анімації підсвічування ходів, аудіоефекти та вібрації — підвищують відчуття «живості» бою та сприяють емоційній залученості.

Огляд найбільш продаваних покрокових ігор 2024 р. демонструє лідерство як традиційних серій (наприклад, XCOM та Fire Emblem), так і нових гібридних проєктів, які експериментують з міським сеттінгом чи науковою фантастикою[2].

Аналіз відгуків користувачів на мобільних платформах свідчить про значущий попит на швидкі дуелі тривалістю до 20 хвилин, поєднані із зручним сенсорним управлінням[3].

Таким чином, предметна галузь покрокових дуельних стратегій характеризується:

- Швидке розширення ринку з високим CAGR та інвесторами, зацікавленими в інноваціях;
- Поєднання карткових, настільних та тактичних механік, що забезпечує глибину та різноманітність ігрового процесу;
- Активне впровадження процедурної генерації для створення непередбачуваних ігрових середовищ;
- Критичну роль UI/UX, де інтуїтивність і якість зворотного зв'язку визначають успіх проєкту;
- Зростаючий попит на короткі, але стратегічно насичені дуелі як на ПК, так і на мобільних пристроях.

Ці тенденції слугують опорою для розробки нашого прототипу, закладаючи фундамент для інноваційної системи бою та адаптивного інтерфейсу в межах жанру покрокової дуельної стратегії.

2.2 Виявлення проблем та актуалізація рішень

У процесі аналізу поточних покрокових дуельних стратегій виявлено низку ключових викликів, що перешкоджають оптимальному ігровому досвіду та подальшому розвитку жанру:

а) Нестабільний баланс бою:

- 1) Занадто сильні або, навпаки, надто слабкі одиниці суттєво спотворюють тактичну глибину, знижуючи мотивацію до подальших дуелей;
- 2) Різниця в наборі карт чи фракцій між гравцями призводить до відчуття «нечесної гри».

б) Складність освоєння механік:

- 1) Велика кількість спеціальних вмінь та умов активації часто плутає новачків, збільшуючи поріг входження;
- 2) Інтерфейс з безліччю кнопок і вкладок заважає швидкому прийняттю рішень під час матчу.

в) Низька реіграбельність карт і мап:

- 1) Статичні рівні швидко набридають через одноманітність розміщення перешкод та ресурсів;
- 2) Обмежена варіативність сценаріїв призводить до повторення одних і тих самих стратегій.

г) Недостатня система навчання:

- 1) Відсутність покрокового туторіалу та інтерактивних підказок ускладнює перший контакт із грою;
- 2) Нечіткі пояснення механік змушують гравців шукати інформацію на сторонніх ресурсах.

д) Одноманітність візуального оформлення UI/UX:

- 1) Статичні елементи інтерфейсу без реактивної анімації знижують залученість;
- 2) Недостатня кастомізація відображення інформації (наприклад, позначення траєкторій руху) уповільнює ігровий процес.

Для подолання зазначених проблем запропоновано наступні рішення:

а) Автоматизований баланс через метрики:

- 1) Впровадити систему аналітики, що відстежує показники перемог/поразок для кожного типу одиниць і автоматично пропонує корекції їхніх характеристик;
- 2) Забезпечити динамічне коригування вартості набору одиниць при побудові армії, враховуючи їхню популярність та ефективність.

б) Спрощений та контекстний інтерфейс:

- 1) Запровадити адаптивні підказки, що з'являються у разі неочікуваної дії користувача, пояснюючи доступні опції;

- 2) Організувати UI у вигляді мінімальних панелей з можливістю розгортання додаткових меню за потреби.

в) Процедурна генерація та модульні сценарії:

- 1) Використовувати алгоритми випадкового компоновання (модульні «плитки») для створення різноманітних карт із мінімальною ручною розробкою;
- 2) Залучити параметризовані сценарії (місії з різними цілями: захопити точку, знищити ворожу вежу), що поєднуються з процедурною топологією.

г) Інтерактивне навчання та прогресивний туторіал:

- 1) Розробити покроковий туторіал із демонстрацією базових механік у формі коротких місій;
- 2) Додати систему «схованих підказок», які гравець зможе звернутися в будь-який момент під час матчу.

д) Динамічні елементи UI/UX:

- 1) Реалізувати анімації підсвічування клітин руху та атаки, а також звукові й вібраційні сигнали під час критичних подій;
- 2) Надати можливість гравцю налаштовувати розташування інформаційних панелей і склад відображених показників (наприклад, прибирати лічильники енергії, якщо вони не використовуються).

Завдяки впровадженню цих рішень прототип гри отримає збалансовану систему бою, зрозумілий та гнучкий інтерфейс, високий рівень реіграбельності й доступне навчання, що сприятиме залученню як новачків, так і досвідчених гравців.

2.3 Цільова аудиторія

Покрокові стратегії традиційно приваблюють геймерів старшої вікової когорти (порівняно з швидкими шутерами чи МОБА), найактивніше серед групи

25–44 роки[4]. При цьому 65–69 % стратегічних гравців — чоловіки, тоді як частка жінок становить близько 11 % у піджанрі turn-based, хоча останні роки спостерігається зростання інтересу завдяки таким франшизам, як Fire Emblem із більшою жіночою аудиторією. На мобільних платформах найбільша група геймерів (27 %) — це користувачі віком 25–34 роки, що підтверджує актуальність орієнтації на цю демографію і в покрокових дуелях[5].

Ключовими сегментами є як досвідчені любителі глибокої тактики, так і новачки, які цінують нетерміновий геймплей і можуть виділити 15–30 хвилин на матч. Аудиторія охоплює PC-геймерів, орієнтованих на детальний контроль і модифікацію налаштувань, та мобільних користувачів, що шукають короткі, але насичені стратегією сесії. Незалежно від платформи, для цієї аудиторії важливі чітка подача інформації, адаптивний UI/UX і поступове введення нових механік через інтерактивний туторіал.

2.4 Виявлення проблем

У ході аналізу жанру покрокових дуельних стратегій і огляду практик розробників виокремилися такі основні проблеми:

- а) Дисбаланс одиниць і колод. Нерівномірна сила юнітів та «метові» набори карт призводять до домінування одних стратегій і зниження різноманітності боїв;
- б) Надмірна складність механік для новачків. Велика кількість спеціальних умінь і умов їх активації збільшує когнітивне навантаження на гравця, ускладнюючи перші матчі та відлякуючи потенційних новачків;
- в) Обмежена реіграбельність статичних карт. Повторюваність локацій та фіксовані сценарії знижують цікавість до багаторазових ігор; відсутність варіативних цілей (наприклад, «захопити точку» або «утримати базу») робить поєдинки передбачуваними;

- г) Перевантажений інтерфейс. Надлишок кнопок і показників без можливості налаштування або приховування непотрібного елементу створює візуальний шум та гальмує процес прийняття рішень;
- д) Недостатня система навчання. Перші кроки гравця часто не підтримуються інтерактивними підказками чи динамічними туторіалами, тому користувачі змушені шукати керівництва на сторонніх ресурсах;
- е) Мережеві затримки в мультиплеєрі. Хоча затримка для покрокових ігор менш критична, ніж для реального часу, візуальні підтвердження ходів і анімації можуть «підвисати», створюючи враження нестабільності з'єднання;
- ж) Непрозорість процедурної генерації. Без наочного пояснення алгоритмів складання карти гравець не може адаптувати стратегію під можливі варіанти поля, а «плейси» іноді створюються несправедливими або занадто однотипними.

Урахування й виправлення цих проблем на початкових етапах проектування забезпечить балансовану, доступну та захопливу покрокову дуельну стратегію з високим рівнем залученості та задоволеності аудиторії.

2.5 Постановка задачі

У рамках розробки прототипу покрокової дуельної стратегії необхідно реалізувати низку ключових компонентів, що забезпечать цілісність геймплею та інтерактивність. Передусім, слід сформулювати механіку черговості ходів із чітким алгоритмом активації й відліку ресурсів дії для кожної одиниці. Паралельно необхідно розробити адаптивний інтерфейс, який відображатиме всі критичні параметри (здоров'я, енергію, доступні вміння) у зручному форматі та підказуватиме гравцю можливі варіанти дій без надмірного навантаження екранного простору.

Окрім цього, важливо впровадити базовий рівень процедурної генерації дуельної арени — модульної картки, що випадково поєднуватиме ігрові секції й

перешкоди, аби забезпечити різноманітність матчів без додаткових витрат на ручне створення кожного поля. Завдання включає також створення простого інтерактивного туторіалу для першого запуску гри, який познайомить новачків із основними правилами та інструментами керування, підвищивши поріг входження та загальну привабливість проєкт.

3 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

3.1 Постановка мети

Метою розробки програмної системи є створення робочого прототипу комп'ютерної гри в жанрі покрокової дуельної стратегії, який забезпечить:

- Здійснення повноцінного циклу дуелі. Забезпечити механізм черговості ходів із чітким алгоритмом активації одиниць, розрахунку ресурсів дії та обробки результатів атак і переміщень;
- Інтуїтивність і гнучкість інтерфейсу. Реалізувати адаптивний UI/UX, що відображає всі необхідні параметри юнітів (здоров'я, енергія, доступні вміння) та надає контекстні підказки, мінімізуючи візуальний шум і полегшуючи прийняття рішень;
- Процедурну генерацію ігрових арен. Розробити систему модульної побудови карти, що випадково комбінує заздалегідь підготовлені «плитки» з перешкодами й зоною взаємодії, забезпечуючи різноманітність і високий рівень реіграбельності;
- Інтерактивне навчання. Додати вбудований туторіал із покроковими місіями, який ознайомлює користувача з базовими механіками та інструментами керування, а також дозволяє в будь-який момент звернутися до підказок під час гри;
- Мультиплатформність. Спроектувати систему таким чином, щоб базова логіка та UI працювали як на ПК, так і на мобільних пристроях, із можливістю масштабування інтерфейсу під різні роздільні здатності екранів.

Досягнення цих цілей забезпечить створення збалансованого, зрозумілого та технічно надійного рішення, яке стане основою подальшої розробки й тестування повноцінної версії гри.

3.2 Загальний опис

У цьому розділі окреслено основні компоненти та функціональні блоки програмної системи, необхідні для реалізації прототипу покрокової дуельної стратегії. З огляду на відмову від мережевого мультиплеєра, гра передбачає режим “передай і ходи” на одній машині. Процедурна генерація рівнів не використовується: локації формуються вручну відповідно до заздалегідь продуманих сценаріїв.

Ігрові механіки

- Чергування ходів: реалізація чергового циклу, де гравці по черзі вибирають одиниці, активують їхні вміння, рухаються та атакують;
- Система дій: кожна фігура має ліміт активацій (рух, атака, спеціальні уміння), який відображається у вигляді лічильника на UI;
- Баланс юнітів: налаштовані показники здоров’я, атаки та захисту так, щоб жодна одиниця не давала надлишкової переваги;
- Міні-AI для противників: простий алгоритм вибору ходів (наприклад, оцінка найближчих цілей за пріоритетом), що дозволяє протистояти одиночному гравцю.

Інтерфейс користувача (UI/UX)

- Головне меню та екран вибору бою: кнопки “Нова дуель”, “Налаштування”, “Туторіал” та “Вихід”.
- Ігровий HUD: мінімалістичні панелі для відображення ходу гравця, стану обраної одиниці (здоров’я, вміння, активації) та кнопка “Передати хід”.
- Контекстні підказки: підсвічування доступних клітин руху та мішеней для атаки; спливаючі підказки при наведенні курсора на іконку вміння.
- Туторіал: інтерактивні послідовні підказки під час першої гри, що демонструють базові механіки руху, атаки та передачі ходу.

Візуальні та аудіоеlementи

- Стиль оформлення: вибір єдиного художнього напрямку (наприклад, *fantasy* або *sci-fi*), із уніфікованими моделями юнітів та шрифтами UI.

- Анімації: плавні переходи для руху фігур, ефекти атаки та спеціальних умінь; прості анімації реакцій при отриманні чи завданні ушкодження.
- Звукові ефекти: короткі дії під час кліку по юніту, підтвердження ходу, а також фонова музика для меню та бою.

Структура рівнів

- Фіксовані карти: дві-три попередньо розроблені дуельні арени з різною топологією (відкриті поля, звужені проходи, укриття).
- Розташування перешкод: кам'яні блоки чи бар'єри, що змінюють траєкторію руху та зонування бою.
- Зони старту: чітко позначені зони для кожного гравця, щоб уникнути перекриття під час початку дуелі.

Цей опис деталізує постановки технічних та геймдизайнерських вимог, які ляжуть в основу наступного етапу розробки.

3.3 Загальні обмеження

Гра має бути створена на базі Unreal Engine 5 із використанням Visual Studio 2022, програмної логіки на C++ та Blueprint.

Системні вимоги:

- Операційна система:

Підтримуються Windows 10 (64-bit, версії 1909 або вище) або Windows 11.

- Процесор:

Мінімально — чотириядерний Intel або AMD із тактовою частотою від 2,5 ГГц. Для комфортної роботи рушія рекомендуються CPU рівня Intel Core i5-8400 або AMD Ryzen 3 3300X та вище.

- Оперативна пам'ять:

Не менше 16 ГБ RAM, хоча для середніх та великих проєктів UE5 оптимально мати 32 ГБ.

- Графічний адаптер:

Відеокарта з підтримкою DirectX 12 та мінімум 8 ГБ відеопам'яті. Для запуску сучасних UE5-ігор на середніх налаштуваннях підходять AMD Radeon RX 5700 або NVIDIA GeForce GTX 1080

– Сховище:

SSD-диск з вільним місцем не менше 50 ГБ для файлів рушія та проекту.

3.4 Припущення та залежності

У цьому проєкті ми виходимо з низки передумов, які спрощують реалізацію та фокусують увагу на ключових механіках, а також визначаємо зовнішні компоненти, від яких залежить успіх розробки.

Припущення:

- Передбачається, що користувач працює на ПК із мінімальними характеристиками, вказаними в розділі 3.3; мобільна версія не розглядається на початковому етапі.
- Гравець знайомий із базовими принципами покрокових стратегій (наприклад, шахи, XCOM), але не обов'язково має досвід складних TCG-проєктів.
- Механіка “передай і ходи” передбачає, що двоє гравців по черзі грають на одному пристрої; онлайн-функціонал не враховується.
- Карти арени створюються вручну та залишаються без змін між сесіями; динамічна генерація не застосовується.
- На старті реалізується обмежений набір юнітів (до 10 типів) та дві-три арени, розширення контенту відбуватиметься на наступних етапах.

Залежності:

- Проєкт цілком покладається на можливості Unreal Engine 5 (система черг Blueprint, рендеринг, керування подіями).
- Візуальні скрипти та код розробляються у Visual Studio 2022; відсутність сумісності з іншими IDE може ускладнити залучення нових розробників.
- Усі 2D/3D-елементи та UI-іконки створюються в Adobe Photoshop і — за

необхідності — у сторонніх пакетах (Blender, Substance Painter); затримки з постачанням арту можуть відтермінувати тестування.

- Використовуються офіційні плагіни Epic Games (наприклад, Enhanced Input, Online Subsystem), хоча мережевий модуль залишається неактивним; оновлення рушія може порушити їхню сумісність.
- Передбачається, що в команді є хоча б один Unreal-програміст із досвідом C++ і Blueprints, а також художник із навичками UI/UX; брак таких спеціалістів може затягнути терміни.
- Доступний ресурс часу обмежений одним семестром або чотирма місяцями; ця залежність визначає обсяг реалізованого прототипу та пріоритети завдань.

Отже, враховуючи сформульовані припущення та зовнішні залежності, можна зробити висновок, що проєкт має чітко окреслені рамки та реалістичний масштаб для початкового етапу розробки. Обмеження, встановлені на рівні платформи, контенту та функціоналу, дозволяють зосередитися на головних механіках геймплею без розпорошення ресурсів. Водночас критична роль окремих інструментів та спеціалістів вказує на необхідність ретельного управління ризиками та часовими обмеженнями, щоб забезпечити стабільне створення життєздатного ігрового прототипу в межах наявних можливостей.

4 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 UML проектування ПЗ

Визначено основні функції, до яких гравець повинен мати доступ. Тож було створено Use-case діаграму, яка зображена на рисунку 4.1

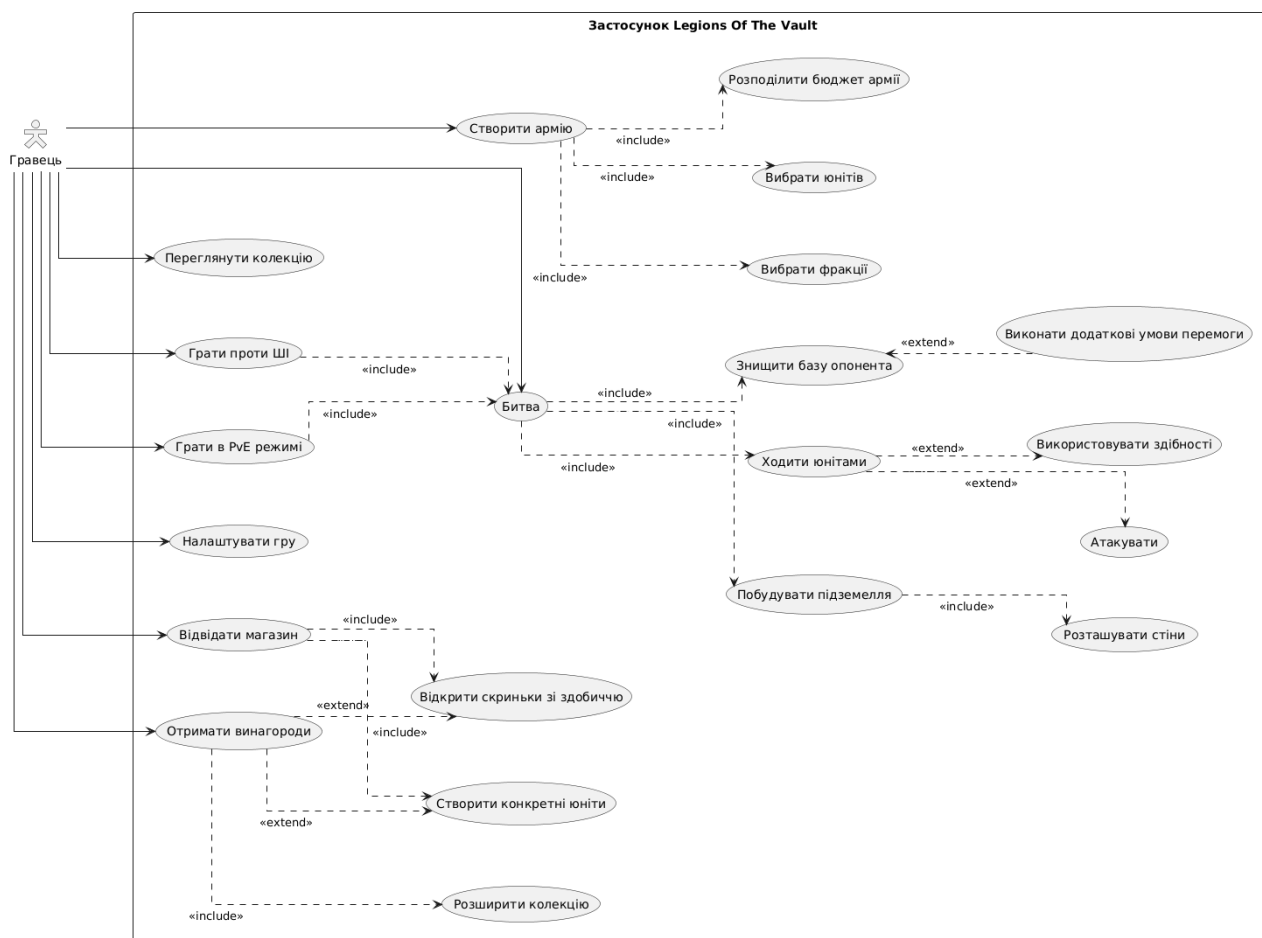


Рисунок 4.1 – Use-case діаграма застосунку Legions Of The Vault

Гравець є користувачем ігрового застосунку «Legions Of The Vault». Він може створити армію, вибравши фракції, юнітів та розподіливши бюджет армії між ними. Для участі в битві гравець повинен побудувати підземелля, розташувати стіни, а також ходити юнітами, які можуть атакувати або використовувати спеціальні здібності. Метою битви є знищення бази опонента, хоча іноді можуть бути й додаткові умови перемоги.

Після успішних битв гравець отримує винагороди, які дозволяють відкривати скриньки зі здобиччю, створювати конкретні юніти та розширювати свою колекцію. У вільний від битв час гравець може переглядати свою колекцію, відвідувати магазин для придбання нових предметів чи юнітів, а також налаштовувати гру під власні потреби.

Застосунок підтримує різні режими гри: гравець може грати в PvE режимі проти ігрового середовища або змагатися проти штучного інтелекту в режимі ШІ. Кожен режим включає битви з унікальними особливостями та стратегічними викликами.

4.2 Реалізація ігрового поля на основі клітинної сітки

У розробленій грі ігрове поле реалізовано як двовимірну клітинну сітку, що є фундаментальною структурою для просторової організації ігрового простору.

Основні характеристики сітки включають:

- Динамічне налаштування розміру сітки (кількість рядків та стовпців);
- Підтримка різних розмірів клітинок;
- Можливість виділення та підсвічування клітинок;
- Система координат для переміщення та взаємодії з ігровими об'єктами.

Клас `ABP_Grid` є центральним компонентом реалізації ігрового поля. Він успадковується від базового класу `AActor` в Unreal Engine і містить наступні ключові властивості:

а) Параметри сітки:

- 1) `NumRows`: Кількість рядків;
- 2) `NumColumns`: Кількість стовпців;
- 3) `TileSize`: Розмір однієї клітинки;
- 4) `LineThickness`: Товщина ліній сітки;
- 5) `LineColor`: Колір ліній сітки.

б) Компоненти візуалізації:

- 1) TileHighlight: Компонент для підсвічування клітинок;
- 2) SelectionProceduralMesh: Процедурна mesh для виділення клітинок;
- 3) LinesProceduralMesh: Процедурна mesh для малювання ліній сітки.

Клас забезпечує наступний функціонал:

а) Перетворення координат:

- 1) LocationToTile(): Перетворення світових координат на координати клітинки;
- 2) TileToGridLocation(): Перетворення координат клітинки на світові координати;

б) Управління станом клітинок:

- 1) SetSelectedTile(): Встановлення вибраної клітинки;
- 2) AddToGridMap(): Додавання об'єкта до сітки;
- 3) RemoveFromGridMap(): Видалення об'єкта з сітки.

в) Алгоритми pathfinding:

- 1) FindPath(): Пошук шляху між клітинками;
- 2) GetNeighborPositions(): Отримання сусідніх клітинок;
- 3) FilterValidPositions(): Фільтрація прохідних клітинок.

На наданих блок-схемах показано складну логіку обробки та маніпуляцій з grid-системою. Перша схема демонструє процес створення ліній та меш для сітки з використанням процедурної генерації. На рисунку 4.2 видно складну логіку з'єднання вузлів, що відповідають за генерацію геометрії сітки.

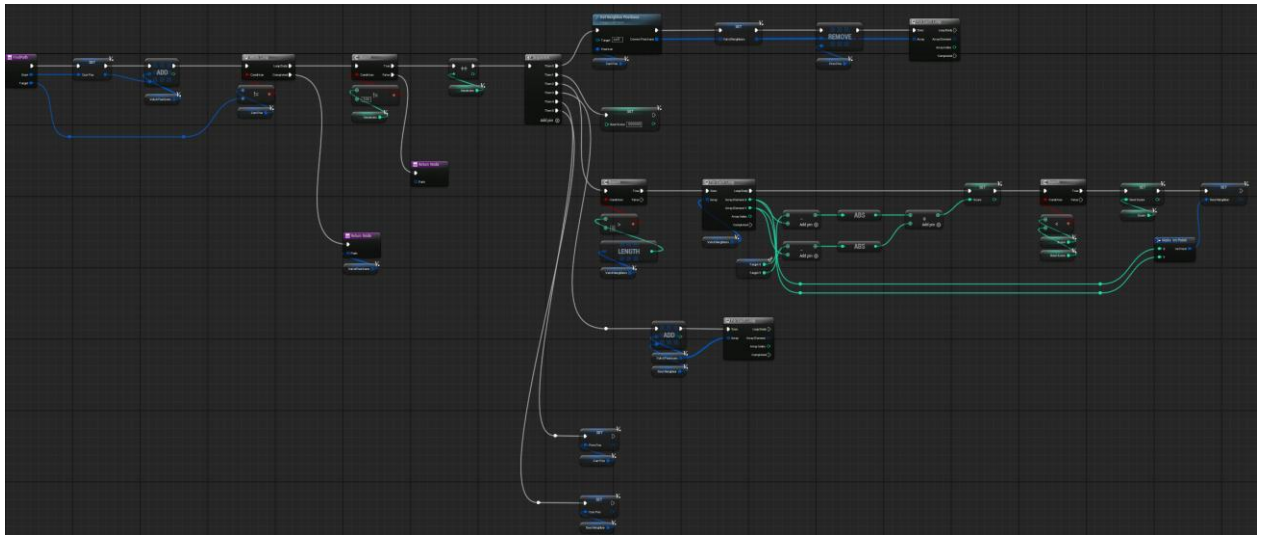


Рисунок 4.3 – Функція для знаходження шляху

Третя схема розкриває додаткові утиліти для роботи з grid-системою, включаючи створення різних типів mesh та обробку геометрії. Рисунок 4.4 демонструє модульність та гнучкість реалізованого підходу.

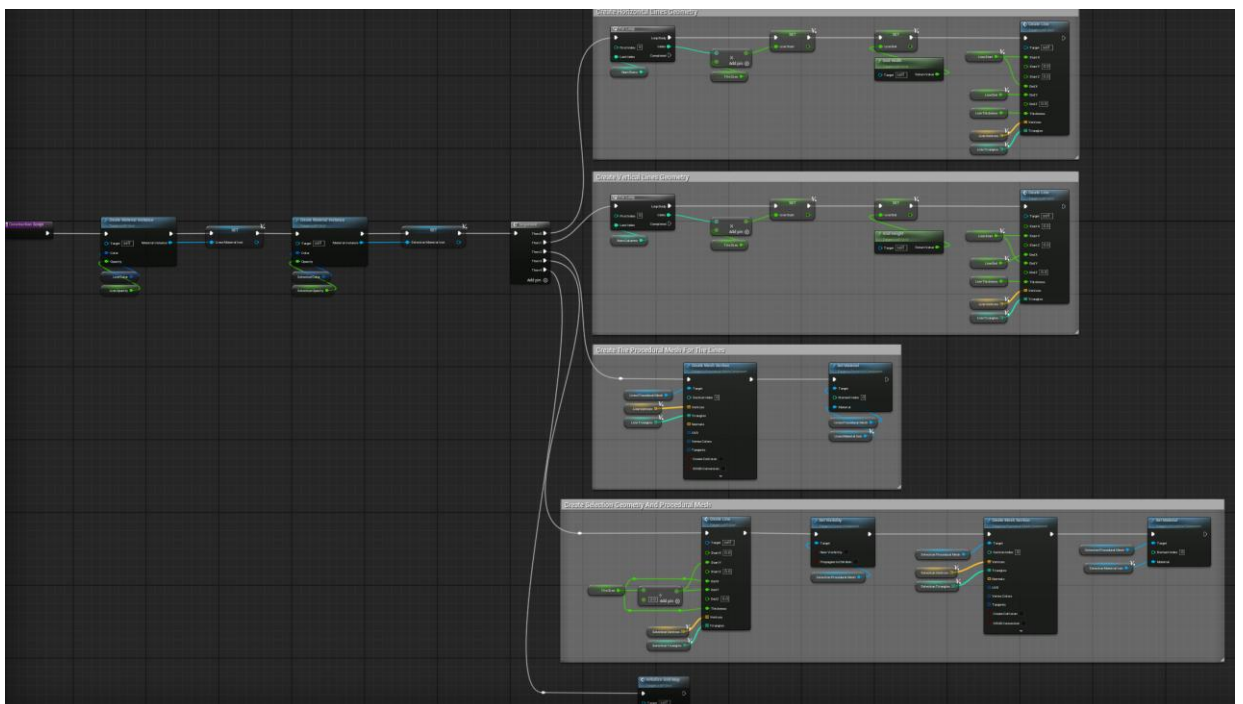


Рисунок 4.4 – Функція ініціалізації системи клітин

4.3 Проектування дизайну рівня

Представлена локація є типовим прикладом вестерн-стилізованого міського середовища з акцентом на нічне освітлення. На рисунку 4.5 зображено фрагмент головної вулиці невеликого поселення в стилі Дикого Заходу з характерними архітектурними рішеннями.

Ключовим об'єктом локації виступає двоповерховий салун, пофарбований у синій колір, що відразу привертає увагу гравця завдяки контрасту з темним оточенням та яскравому внутрішньому освітленню, яке проникає крізь вікна. На фасаді розміщена вивіска "SALOON", що чітко ідентифікує призначення будівлі та слугує орієнтиром для навігації гравця. Салун обладнано просторою верандою та сходами, що створюють додаткові можливості для тактичного позиціювання під час ігрових дуелей.

На протилежному боці вулиці розташовано дерев'яні будівлі з балконами другого поверху, що формують класичну композицію західного містечка та створюють асиметричний баланс локації. Такий дизайн надає гравцям різноманітні варіанти для руху та укриття, а також створює цікаві просторові сценарії для ігрового процесу.

Центральна частина вулиці містить інтерактивний елемент у вигляді напівпрозорого куба з блакитним символом всередині, що, ймовірно, слугує точкою активації певних ігрових механік або візуальним індикатором зони особливого призначення. Це може бути як зона спавну персонажів, так і місце для проведення покрокових дуелей у рамках геймплею.

Освітлення локації реалізовано через комбінацію теплого світла з вікон будівель та холодного місячного сяйва, що створює атмосферний контраст та підкреслює глибину простору. Вдалині видніються силуети гір, що обмежують ігрову зону та створюють відчуття ізоляваності поселення.

У дизайні рівня також передбачено елементи міської інфраструктури — ліхтарі, з'єднані дротами, що додають автентичності та слугують додатковими

джерелами освітлення для ігрових сцен.

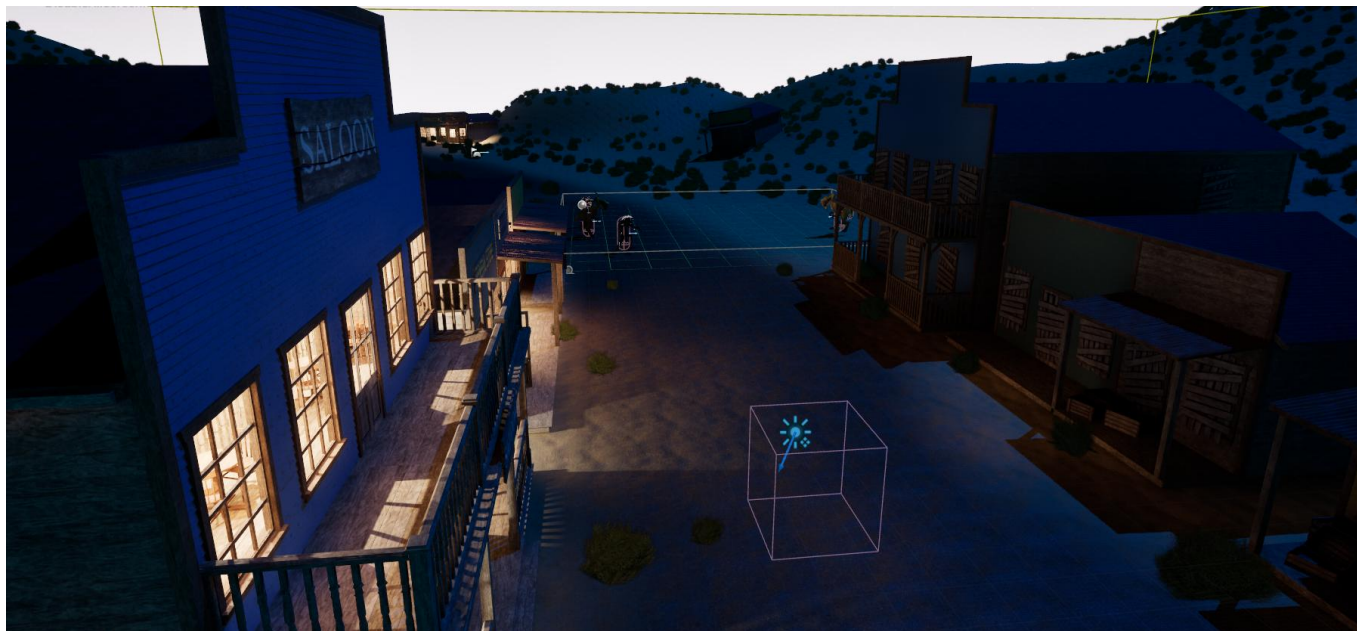


Рисунок 4.5 - Реалізація локації вестерн-містечка в Unreal Engine 5

Загальна композиція рівня спрямована на забезпечення різноманітності ігрових сценаріїв: вертикальна складність (балкони, веранди, сходи) дозволяє реалізувати тактичні переваги висоти, а горизонтальне планування вулиці сприяє маневруванню та стратегічному позиціюванню під час покрокових боїв.

ВИСНОВКИ

Підсумовуючи виконану роботу, було створено функціональний прототип покрокової дуельної стратегії з реалізованою системою черговості ходів, лімітом дій для кожної одиниці та базовим AI-супротивником. Ігровий застосунок розроблено на Unreal Engine 5 із використанням C++ та Blueprints, що забезпечило гнучке поєднання продуктивного коду та швидкого впровадження змін через візуальні скрипти[6]. Для створення графічних ресурсів та UI-елементів застосовано Adobe Photoshop, що дозволило опрацювати інтерфейс іконок та HUD відповідно до стандартів сучасного геймдизайну.

У ході розробки проведено аналіз предметної галузі й виявлено ключові проблеми балансування, доступності механік та реіграбельності, після чого запропоновано й апробовано рішення щодо адаптивного UI/UX із контекстними підказками та динамічними анімаціями[7]. Обмежений набір фіксованих карт і модульних арени дозволив закласти основу для подальшого розширення контенту, тоді як інтерактивний туторіал сприяв зниженню порогу входження для новачків.

У перспективі проєкт може бути вдосконалений через впровадження процедурної генерації рівнів для підвищення реіграбельності та варіативності ігрового процесу. Крім того, подальше доопрацювання UI/UX із урахуванням принципів гейміфікації та зворотного зв'язку допоможе підсилити залученість гравців і зробити бойову механіку ще більш насиченою й динамічною. Нарешті, розширення асортименту юнітів, арсеналу й сценаріїв із унікальними цілями створить більш глибокий і комплексний ігровий досвід, що відповідатиме очікуванням як новачків, так і ветеранів покрокових стратегій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Strategy Games Market Size [Електронний ресурс] – URL: <https://www.businessresearchinsights.com/market-reports/strategy-games-market-118765> (дата звернення 08.05.2025).
2. Top 10 Selling Turn-Based Games of 2024 [Електронний ресурс] – URL: <https://medium.com/%40ThatOneAJ/top-10-selling-turn-based-games-of-2024-cae68bb9ebc3> (дата звернення 07.05.2025).
3. Q2 2024 Turn-Based Tactics Game Trends on Android in the US [Електронний ресурс] – URL: <https://sensortower.com/blog/2024-q2-android-top-5-turn-based%20tactics%20games-revenue-ve-62546b22f5dff39390fba7c8> (дата звернення 06.05.2025).
4. Revisiting the Strategy Genre Map: Age, Audience Homogeneity, and the Lasso Effect [Електронний ресурс] – URL: quanticfoundry.com/2016/03/23/revisiting-the-strategy-genre-map (дата звернення 09.05.2025).
5. Share of mobile gamers worldwide as of December 2018, by age [Електронний ресурс] – URL: <https://www.statista.com/statistics/246577/age-distribution-of-mobile-gamers/> (дата звернення 10.05.2025).
6. How Unreal Engine is Transforming Game Development Industry [Електронний ресурс] – URL: <https://www.strivemindz.com/blog/how-unreal-engine-is-transforming-game-development-industry/> (дата звернення 10.05.2025).
7. Enhancing the Player Experience- The Significance of UI/UX in Gaming [Електронний ресурс] – URL: https://www.gamixlabs.com/blog/importance-of-ui-ux-in-gaming/?utm_source=chatgpt.com (дата звернення 10.05.2025).

ДОДАТОК А

Концепт-документ до гри

“Legions of the Vault” Game Design Documentation



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

Рисунок А.1 – Сторінка 1 геймдизайнерського документу

1. Вступ

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

Цей проєкт 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 лутбокси (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 А.2 – Сторінка 2 геймдизайнерського документу

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 всіх 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 повтілись 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...

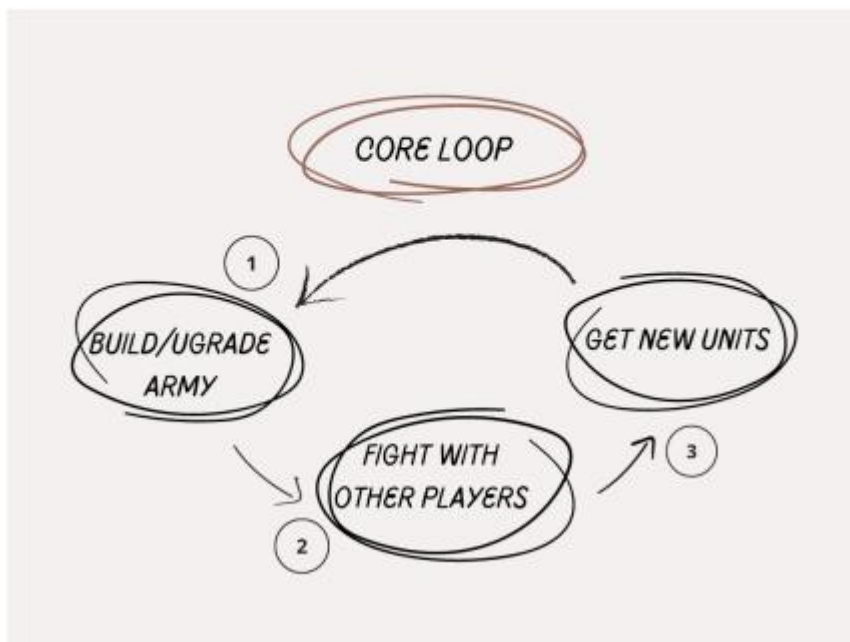
Чому нарратив це важливо?

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.

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 А.4 – Сторінка 4 геймдизайнерського документу

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).

Рисунок А.5 – Сторінка 5 геймдизайнерського документу

У вас буде кілька tokenів 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 редагуванню 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:**

Рисунок А.6 – Сторінка 6 геймдизайнерського документу



У покемонах можна зловити більше 100 пікачу, але всі вони будуть різні, бо від початку будуть мати різні характеристики. На екранах 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)
- Різна кількість 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 А.7 – Сторінка 7 геймдизайнерського документу

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

Єдина зміна в юніти 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.
- **Реклама:** чередується між показом геймплею та нарративом світу.

Рисунок А.10 – Сторінка 10 геймдизайнерського документу