

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ І ПРИКЛАДНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ТУПІЦЬКИЙ ВЛАДИСЛАВ ВІТАЛІЙОВИЧ

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій,
д-р. техн. наук, професор
_____ Наталія ВЕСЕЛОВСЬКА
« ____ » _____ 2026 р.

РОЗРОБКА КОМП'ЮТЕРНОЇ ГРИ В ЖАНРІ ПОКРОКОВОЇ СТРАТЕГІЇ

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:
Надія ПОТАПОВА, доцент кафедри
інформаційних технологій,
к. е. н., доцент

Оцінка: _____ / _____ / _____
(бали за шкалою ЄКТС / за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця - 2026

АНОТАЦІЯ

Тупіцький В. В. Розробка комп'ютерної гри в жанрі покрокової стратегії. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця 2026.

У кваліфікаційній (бакалаврській) роботі досліджено особливості проєктування та розробки комп'ютерних ігор жанру покрокова стратегія. Створено прототип комп'ютерної гри жанру покрокова історична стратегія, події якої присвячені періоду Української революції 1917–1921 років. У програмному продукті реалізовано систему держав і провінцій, механіки управління ресурсами, будівництва, дипломатії, військових підрозділів, а також систему покрокового перебігу часу та користувацький інтерфейс для взаємодії

Ключові слова: покрокова стратегія, Комп'ютерна гра, Unity, C#, ігровий рушій, дипломатія, економічна система, військова система, історична стратегія.

58 ст., 18 рис., 7 табл., 2 дод., 47 джерел.

ABSTRACT

Tupitsky V. V. Development of a Computer Game in the Turn-Based Strategy Genre. Specialty 122 «Computer Science», educational program «Computer Science». Vasyl Stus Donetsk National University, Vinnytsia 2026.

The qualification (bachelor's) thesis examines the specifics of designing and developing turn-based strategy video games. A prototype of a turn-based historical strategy video game was created, set during the Ukrainian Revolution of 1917–1921. The software product implements a system of states and provinces, mechanics for resource management, construction, diplomacy, and military units, as well as a turn-based time progression system and a user interface for interaction

Keywords: turn-based strategy, computer game, Unity, C#, game engine, diplomacy, economic system, military system, historical strategy.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ТЕОРЕТИКО-КОНЦЕПТУАЛЬНІ ЗАСАДИ РОЗРОБКИ ІГОР СТРАТЕГІЇ.....	6
1.1. Сутність та поняття ігрових додатків типу стратегія.....	6
1.2 Концептуальні засади розробки стратегічних ігор	9
1.3 Основні механіки стратегічних ігор	13
РОЗДІЛ 2. ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ РОЗРОБКИ ІГРОВОГО ДОДАТКУ	20
2.1 Архітектура комп'ютерної гри жанру стратегія.....	20
2.2 Обґрунтування інструментальних та програмних засобів розробки стратегічної гри	22
2.3 Алгоритми розробки рушія гри	27
2.4 Вимоги до ігрового додатку	32
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІГРОВОГО ДОДАТКУ ЖАНРУ СТРАТЕГІЯ	37
3.1 Розробка дизайну комп'ютерної гри	37
3.2 Програмна реалізація основних компонентів ігрового додатку	41
3.3 Тестування додатку жанру стратегія	46
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ	55
ДОДАТКИ.....	58

ВСТУП

Сучасна індустрія комп'ютерних ігор є однією з найбільш динамічних галузей інформаційних технологій. Особливе місце серед різноманітних жанрів займають стратегічні ігри, які поєднують у собі елементи управління, планування, моделювання складних систем та прийняття рішень. Такі ігри дозволяють користувачу не лише розважатися, а й аналізувати економічні, політичні та військові процеси.

Одним із найскладніших та наймасштабніших піджанрів стратегічних ігор є grand strategy. Ігри цього типу моделюють функціонування держав, економік, армій, дипломатичних систем та внутрішньої політики протягом значних історичних періодів. Grand strategy поєднує в собі елементи глобального управління державою, економічного розвитку, військового планування та дипломатичної взаємодії.

Особливий інтерес для дослідження становлять історичні стратегії, що відтворюють реальні події та дозволяють моделювати альтернативні сценарії розвитку історії. Події 1917–1921 років на території України є надзвичайно складним та багатогранним історичним періодом, що включає революції, боротьбу за незалежність, громадянську війну, зовнішнє втручання та політичну нестабільність.

У сучасній ігровій індустрії існує обмежена кількість ігор, які детально моделюють зазначений історичний період та приділяють значну увагу українському контексту. Більшість існуючих grand strategy-ігор або поверхнево відображають події Української революції, або зосереджуються на значно ширших глобальних конфліктах.

Актуальність теми полягає у необхідності дослідження методів створення складних simulation-систем для стратегічних ігор, а також у можливості популяризації історичних подій Української революції засобами інтерактивних цифрових технологій.

Метою дипломної роботи є проектування та розробка комп'ютерної гри жанру grand strategy, присвяченої подіям 1917–1921 років, із реалізацією політичної, економічної, дипломатичної та військової симуляції.

Для досягнення поставленої мети необхідно виконати такі завдання:

- дослідити теоретико-концептуальні засади розробки ігор жанру стратегія;
- провести аналіз існуючих ігор-аналогів в жанрі стратегія;
- обґрунтувати вибір технологій та інструментів розробки ігрового додатку;
- спроектувати архітектуру стратегічної гри
- реалізувати систему провінцій та карти;
- створити основні системи управління державою в рамках гри: економічну, політичну та дипломатичну, військову та систему постачання.

Об'єктом дослідження є процес створення комп'ютерних стратегічних ігор.

Предметом дослідження є методи, алгоритми та програмні засоби реалізації grand strategy simulation.

Практичне значення роботи полягає у створенні функціонального програмного продукту, який може бути використаний як основа для подальшого розвитку повноцінної історичної grand strategy-гри.

Апробація результатів дослідження. Основні положення та результати кваліфікаційної роботи були апробовані на VII Всеукраїнській науково-практичній конференції здобувачів вищої освіти та молодих вчених «Прикладні інформаційні технології» (м. Вінниця, 15 травня 2026 р.).

РОЗДІЛ 1

ТЕОРЕТИКО-КОНЦЕПТУАЛЬНІ ЗАСАДИ РОЗРОБКИ ІГОР СТРАТЕГІЇ

1.1. Сутність та поняття ігрових додатків типу стратегія

Комп'ютерні ігри є одним із найпоширеніших видів цифрових програмних продуктів сучасності. Вони поєднують у собі елементи програмування, комп'ютерної графіки, математичного моделювання, штучного інтелекту та інтерактивного дизайну. У процесі розвитку інформаційних технологій комп'ютерні ігри перетворилися із простих розважальних програм на складні програмні системи, які здатні моделювати економічні, політичні, військові та соціальні процеси[1].

Серед великої кількості жанрів особливе місце займають стратегічні ігри. Вони орієнтовані не стільки на швидкість реакції користувача, скільки на його здатність аналізувати ситуацію, планувати дії та приймати ефективні рішення. Саме тому стратегічні ігри часто використовуються не лише як розважальний продукт, але й як інструмент навчання, розвитку аналітичного мислення та моделювання складних систем.

Поняття «стратегічна гра» походить від терміна «стратегія», який традиційно використовується у військовій справі та означає мистецтво планування і керування ресурсами для досягнення визначеної мети. У комп'ютерних іграх стратегія являє собою жанр, у якому гравець керує певною державою, цивілізацією, організацією або групою персонажів, приймаючи рішення щодо розвитку, економіки, дипломатії та військових дій[2].

Основною особливістю стратегічних ігор є високий рівень абстракції керування. Гравець зазвичай не контролює окремого персонажа безпосередньо, а здійснює управління великими структурами. Це можуть бути держави, армії, міста, економічні системи або навіть цілі цивілізації. Відповідно, успіх у грі залежить від здатності прогнозувати наслідки власних рішень та ефективно використовувати доступні ресурси.

Стратегічні ігри можуть суттєво відрізнятися між собою за масштабом.

Одні моделюють окремі бойові операції або розвиток невеликих поселень, тоді як інші дозволяють керувати цілими державами протягом десятків або навіть сотень років історичного розвитку. Саме тому даний жанр характеризується значною різноманітністю механік та підходів до реалізації ігрового процесу[3].

Таблиця 1.1 – Класифікація стратегічних ігор

Тип стратегій	Характеристика	Приклади
RTS	Дії відбуваються в реальному часі	StarCraft II, Age of Empires IV
TBS	Покрокове управління	Civilization VI, Old World
Grand Strategy	Управління державою в глобальному масштабі	Hearts of Iron IV, Europa Universalis IV
4X	Дослідження, розширення, експлуатація, знищення	Endless Space 2, Stellaris

З точки зору теорії комп'ютерних ігор стратегічні ігри можна визначити як інтерактивні програмні системи, у яких результат діяльності користувача залежить переважно від процесу планування, аналізу інформації та прийняття управлінських рішень. Такі системи моделюють взаємодію різних елементів складного середовища та створюють умови для довгострокового прогнозування наслідків дій гравця[4].

Важливою характеристикою стратегічних ігор є наявність ресурсної системи. Ресурси виступають основою економічного розвитку та визначають можливості гравця. До найбільш поширених ресурсів належать продовольство, деревина, метал, золото, нафта, людські ресурси та інші показники, які можуть

бути використані для розвитку держави або створення військових формувань[5].

Ще однією характерною особливістю є система прийняття рішень. На відміну від багатьох інших жанрів, у стратегіях гравець постійно стикається з необхідністю вибору між кількома альтернативами. Кожне рішення може мати як позитивні, так і негативні наслідки, що створює складну систему причинно-наслідкових зв'язків.

Сучасні стратегічні ігри також активно використовують механізми дипломатії. Взаємодія між державами або фракціями дозволяє укласти союзи, вести переговори, встановлювати торговельні угоди або оголошувати війни. Завдяки цьому формується складна політична система, яка значно підвищує рівень реалістичності ігрового процесу[6].

Особливу роль у стратегічних іграх відіграє військова складова. Бойові дії можуть бути представлені як у вигляді окремих тактичних битв, так і у формі глобального управління арміями. При цьому військова система зазвичай тісно пов'язана з економікою, логістикою та політикою, що дозволяє створювати більш реалістичну модель функціонування держави[7].

Одним із важливих напрямків розвитку жанру є історичні стратегічні ігри. Такі проекти відтворюють реальні історичні події, дозволяючи користувачу впливати на перебіг історії та створювати альтернативні сценарії розвитку подій. Саме до цієї категорії належить більшість сучасних grand strategy ігор[8].

Для дипломного проєкту особливий інтерес становлять покрокові стратегічні ігри. На відміну від стратегій реального часу, вони дозволяють гравцю детально аналізувати ситуацію та приймати рішення без часових обмежень. Це особливо важливо під час моделювання складних політичних, економічних та військових процесів[9].

Покрокова система передбачає, що всі учасники гри виконують свої дії послідовно в межах визначених ходів. Після завершення ходу відбувається обробка результатів, оновлення стану ігрового світу та починається наступний цикл управління. Такий підхід забезпечує високий рівень контролю над процесом гри та сприяє реалізації складних механік державного управління.

Таким чином, стратегічні ігри є складними програмними системами, що поєднують механізми управління ресурсами, економікою, політикою, дипломатією та військовою діяльністю. Вони створюють умови для моделювання складних соціально-економічних процесів та розвитку навичок стратегічного мислення[10]. Саме тому даний жанр є одним із найбільш перспективних напрямків розробки сучасних Комп'ютерних ігор.

1.2 Концептуальні засади розробки стратегічних ігор

Розробка стратегічних комп'ютерних ігор є складним багатокомпонентним процесом, який поєднує принципи програмної інженерії, системного аналізу, математичного моделювання, геймдизайну та інтерактивного проєктування. На відміну від багатьох інших жанрів комп'ютерних ігор, стратегічні ігри орієнтовані на створення складних взаємопов'язаних систем, у межах яких кожне рішення користувача може впливати на подальший розвиток ігрового світу[1, 2].

Основою будь-якої стратегічної гри є концепція управління. Гравець виступає не безпосереднім виконавцем окремих дій, а керівником певної системи – держави, цивілізації, міста, армії або економічної структури. Саме тому під час проєктування стратегічної гри особливу увагу приділяють побудові механізмів прийняття рішень та відображенню наслідків цих рішень у межах ігрового процесу[3].

Одним із ключових принципів розробки стратегічних ігор є принцип системності. У стратегічній грі всі елементи повинні бути взаємопов'язаними між собою. Економіка впливає на військову сферу, військові події змінюють політичну ситуацію, політичні рішення впливають на дипломатичні відносини, а дипломатія визначає можливості економічного розвитку. Завдяки такому підходу формується цілісна модель функціонування ігрового світу[4].

Важливою концептуальною основою стратегічних ігор є використання ресурсної моделі. Ресурси виступають фундаментом розвитку будь-якої держави або організації. У більшості сучасних стратегій використовується

декілька типів ресурсів, кожен із яких виконує окрему функцію в економічній системі гри[5].

До найбільш поширених ресурсів належать продовольство, фінанси, промислові матеріали, енергетичні ресурси та людський потенціал. Гравець повинен не лише накопичувати ресурси, але й ефективно розподіляти їх між різними напрямками розвитку. Саме обмеженість ресурсів створює необхідність стратегічного планування та визначає складність прийняття рішень[5].

Наступним важливим принципом є баланс між ризиком та винагородою. Будь-яке рішення у стратегічній грі повинно супроводжуватися як потенційними перевагами, так і можливими негативними наслідками. Наприклад, збільшення військових витрат може підвищити обороноздатність держави, але водночас зменшити фінансування економічного розвитку. Завдяки цьому створюється необхідність оцінювання ризиків та довгострокового планування[3].

Однією з найважливіших складових стратегічних ігор є механізм розвитку. Гравець повинен відчувати прогрес та результати власних дій. Розвиток може проявлятися у збільшенні території держави, зростанні населення, покращенні економічних показників, технологічному прогресі або підвищенні військової могутності[2].

Суттєве значення під час розробки стратегічних ігор має механіка досліджень та технологічного розвитку. Технологічне дерево дозволяє моделювати науковий прогрес та поступове відкриття нових можливостей для гравця. Подібний підхід активно використовується у багатьох сучасних стратегіях, оскільки створює додаткову мотивацію для розвитку та забезпечує довгострокову динаміку ігрового процесу[6].

Не менш важливою складовою є система дипломатії. У сучасних стратегічних іграх взаємодія між державами відіграє не менш важливу роль, ніж військові дії. Дипломатичні механіки дозволяють укладати союзи, здійснювати торговельні операції, покращувати або погіршувати відносини між країнами, а також формувати міжнародні коаліції[4].

Наявність дипломатичної системи значно підвищує рівень реалістичності гри, оскільки дозволяє досягати поставлених цілей не лише військовим шляхом. Гравець отримує можливість використовувати політичні інструменти для досягнення стратегічних переваг[4].

Важливим елементом концепції стратегічних ігор є військова система. Вона повинна забезпечувати баланс між простотою використання та глибиною моделювання. Надмірно складні механіки можуть ускладнити сприйняття гри, тоді як надто спрощені системи знижують рівень стратегічного планування[7].

У сучасних стратегічних іграх військові формування зазвичай характеризуються набором параметрів, серед яких чисельність особового складу, рівень підготовки, бойовий дух, мобільність, показники атаки та захисту. Взаємодія цих параметрів дозволяє створювати реалістичні бойові сценарії та підвищує варіативність ігрового процесу[7].

Таблиця 1.2 – Основні ігрові механіки стратегій

Механіка	Призначення
Економіка	Виробництво ресурсів
Політика	Управління державою
Дипломатія	Взаємодія між країнами
Армія	Захоплення територій
Дослідження	Розвиток технологій

Окрему увагу необхідно приділити концепції покрокового управління. На відміну від стратегій реального часу, покрокові стратегії дозволяють гравцеві ретельно аналізувати ситуацію перед прийняттям рішення. Усі дії виконуються послідовно в межах визначеного ходу, після чого відбувається оновлення стану ігрового світу[8].

Покрокова модель має низку переваг. По-перше, вона значно спрощує реалізацію складних механік економіки та дипломатії. По-друге, дозволяє користувачу концентруватися на аналізі ситуації, а не на швидкості реакції. По-

третє, забезпечує більш передбачуване функціонування програмної системи та спрощує процес тестування[8].

Для розробки історичних стратегічних ігор важливим є принцип історичної достовірності. Ігровий світ повинен відображати характерні особливості відповідної історичної епохи. Це стосується політичного устрою держав, економічних умов, військових технологій, дипломатичних відносин та культурних особливостей[9].

Разом із тим сучасні історичні стратегії часто використовують концепцію альтернативної історії. Вона дозволяє користувачеві змінювати перебіг історичних подій залежно від власних рішень. Такий підхід суттєво підвищує реіграбельність та створює додатковий інтерес до ігрового процесу[9].

Ще одним важливим принципом є модульність архітектури програмного забезпечення. Сучасні стратегічні ігри містять велику кількість взаємопов'язаних механік, тому використання модульної структури дозволяє спростити підтримку та подальший розвиток програмного продукту. Кожна підсистема – економіка, дипломатія, армія, політика, населення або інфраструктура – реалізується як окремий функціональний модуль[10].

Модульний підхід забезпечує масштабованість програмного продукту та дозволяє поступово розширювати функціональні можливості гри без необхідності повної переробки існуючої архітектури[10].

Особливого значення під час проєктування стратегічних ігор набуває штучний інтелект. Його основним завданням є моделювання поведінки Комп'ютерних супротивників. Ефективний штучний інтелект повинен аналізувати поточний стан гри, оцінювати ризики, планувати розвиток держави та приймати рішення відповідно до власних стратегічних цілей[11].

Для реалізації подібних систем використовуються алгоритми прийняття рішень, дерева поведінки, системи пріоритетів та інші методи штучного інтелекту. Якість реалізації ШІ безпосередньо впливає на складність та цікавість ігрового процесу[11].

Таким чином, концептуальні засади розробки стратегічних ігор базуються

на принципах системності, взаємозалежності ігрових механік, управління ресурсами, покрокового прийняття рішень, дипломатії, військового планування та історичного моделювання. Використання зазначених принципів дозволяє створювати складні інтерактивні системи, здатні забезпечити високий рівень реалістичності, варіативності та залучення користувача до процесу управління ігровим світом [11].

1.3 Основні механіки стратегічних ігор

Стратегічні комп'ютерні ігри є одним із найбільш технологічно складних напрямів сучасної ігрової індустрії. На відміну від багатьох інших жанрів, вони поєднують велику кількість взаємопов'язаних систем, серед яких економіка, дипломатія, військова справа, розвиток територій, внутрішня політика та управління ресурсами[12, 13]. Протягом останніх десятиліть стратегічні ігри пройшли значний шлях розвитку – від простих економічних симуляторів до складних глобальних моделей державного управління[1, 14].

Сучасний ринок стратегічних ігор представлений великою кількістю проєктів, які відрізняються підходами до реалізації ігрових механік, масштабами моделювання та ступенем історичної достовірності. Аналіз існуючих рішень дозволяє визначити основні тенденції розвитку жанру та виявити найбільш ефективні підходи до побудови стратегічних систем[10, 13].

Однією з найвідоміших сучасних стратегічних ігор є Hearts of Iron IV, розроблена компанією Paradox Interactive[22]. Гра присвячена подіям Другої світової війни та дозволяє керувати будь-якою державою світу в період з 1936 по 1948 роки. Основною особливістю Hearts of Iron IV є надзвичайно детальна військово-політична модель (рис. 1.1)[22]. Гравець може управляти економікою країни, здійснювати дипломатичну діяльність, розвивати промисловість, формувати армії та проводити масштабні військові кампанії. Значна увага приділяється логістиці, виробництву озброєння та плануванню бойових операцій[22].

Перевагою гри є висока глибина стратегічного планування та значна

кількість можливих сценаріїв розвитку подій. Водночас складність інтерфейсу та велика кількість взаємопов'язаних механік створюють високий поріг входу для нових користувачів[22].



Рисунок. 1.1 – Hearts of Iron IV

Іншим представником жанру є Europa Universalis IV[23]. Гра охоплює період з XV до XIX століття та дозволяє керувати державами в епоху Великих географічних відкриттів, колонізації та становлення національних держав.



Рисунок 1.2 – Europa Universalis IV

Особливістю Europa Universalis IV є складна дипломатична система. У грі реалізовано широкий спектр міжнародних взаємодій, серед яких укладання союзів, династичні шлюби, торговельні угоди, створення коаліцій та ведення переговорів між державами[23]. Значна увага також приділяється розвитку економіки та міжнародної торгівлі.

Система управління державою в Europa Universalis IV дозволяє моделювати історичний розвиток різних країн світу та створювати альтернативні історичні сценарії[23]. Завдяки цьому гра стала одним із найбільш успішних прикладів реалізації глобальної історичної стратегії.

Важливе місце серед сучасних стратегічних ігор займає Victoria 3 (рис. 1.3). На відміну від багатьох інших представників жанру, вона орієнтована насамперед на економічне та соціальне моделювання[14].



Рисунок 1.3 – Victoria 3

Основою ігрового процесу Victoria 3 виступає система населення, яка враховує соціальні групи, рівень добробуту, політичні погляди та економічні потреби громадян. Гравець повинен підтримувати баланс між економічним розвитком, політичною стабільністю та соціальними реформами[14].

Особливістю Victoria 3 є високий рівень деталізації економічних процесів.

У грі моделюються виробничі ланцюги, міжнародна торгівля, ринок праці та фінансова система. Завдяки цьому проєкт демонструє сучасний підхід до побудови економічних симуляцій у стратегічних іграх[14].

Ще одним популярним представником жанру є Crusader Kings III (рис. 1.4)[24]. Дана гра відрізняється від більшості стратегій тим, що основна увага приділяється не державам, а окремим історичним персонажам та династіям.



Рисунок 1.4 – Crusader Kings III

У межах ігрового процесу користувач управляє правителем та його родиною, здійснює політичні рішення, організовує дипломатичні переговори та впливає на розвиток власної держави через систему міжособистісних відносин[24]. Подібний підхід дозволяє поєднати елементи рольової гри зі складними стратегічними механіками.

Суттєвий вплив на розвиток жанру справила серія Sid Meier's Civilization VI (рис. 1.5)[11]. На відміну від більшості grand strategy-проєктів, Civilization використовує покрокову систему управління.

У грі користувач розвиває цивілізацію від стародавнього світу до сучасності. Перемогу можна здобути різними шляхами: військовим, культурним, науковим, дипломатичним або релігійним[11]. Такий підхід забезпечує високу варіативність проходження та значно підвищує

реіграбельність.



Рисунок 1.5 – Civilization VI

Покрокова система Civilization VI дозволяє гравцю ретельно аналізувати ситуацію перед кожним рішенням. Саме ця особливість зробила серію одним із найбільш успішних прикладів покрокових стратегій у світовій ігровій індустрії[11].

Серед покрокових стратегій уваги заслуговує Old World. Гра поєднує класичні механіки розвитку цивілізації з елементами династичного управління[14].



Рисунок 1.6 – Old World

Особливістю Old World є система наказів, яка обмежує кількість дій, що можуть бути виконані протягом одного ходу. Завдяки цьому гравець змушений постійно визначати пріоритети та оптимізувати використання доступних ресурсів[14].

Таблиця 1.3 – Переваги та недоліки проаналізованих ігор

Гра	Переваги	Недоліки
HoI4	Глибока військова система	Високий поріг входження
EU4	Потужна дипломатія	Складність освоєння
Victoria 3	Детальна економіка	Складний інтерфейс
Civilization VI	Простота освоєння	Менша історична деталізація
Old World	Якісна покрокова система	Менший масштаб

Важливим напрямом розвитку сучасних стратегічних ігор є використання складних моделей штучного інтелекту. Розробники прагнуть створити Комп'ютерних супротивників, здатних адаптуватися до дій користувача та приймати обґрунтовані стратегічні рішення[2, 7].

Сучасні алгоритми штучного інтелекту використовують системи пріоритетів, дерева поведінки, оцінювання станів та інші методи автоматичного прийняття рішень[2, 7]. Це дозволяє підвищити складність гри без необхідності штучного збільшення ресурсів Комп'ютерних суперників.

Ще однією важливою тенденцією є використання модульної архітектури програмного забезпечення. Великі стратегічні проекти складаються з десятків взаємопов'язаних підсистем, тому розробники активно застосовують принципи об'єктно-орієнтованого програмування та компонентного підходу[3, 5, 8].

У сучасних умовах значна частина стратегічних ігор створюється з використанням універсальних ігрових рушіїв. Найбільш поширеними серед них є Unity Technologies та Epic Games Unreal Engine[17, 18]. Використання готових рушіїв дозволяє суттєво скоротити час розробки та зосередити увагу на реалізації ігрових механік[4, 17].

Аналіз сучасних стратегічних ігор дозволяє виділити кілька основних тенденцій розвитку жанру:

- збільшення складності економічних та політичних моделей;
- інтеграція дипломатичних механік у загальну систему управління державою;
- використання історично достовірних сценаріїв;
- розширення можливостей альтернативного історичного розвитку;
- застосування модульної архітектури програмного забезпечення;
- удосконалення алгоритмів штучного інтелекту;
- розвиток покрокових механік управління для підвищення глибини стратегічного аналізу[7, 13, 14].

На основі проведеного аналізу можна зробити висновок, що сучасні стратегічні ігри прагнуть до максимально детального моделювання процесів управління державою та суспільством[13, 14]. Найбільш перспективним напрямом розвитку є поєднання історичної достовірності, складних економічних і політичних систем, а також покрокового механізму прийняття рішень[9, 15].

Отже, зазначені принципи доцільно використати під час розробки власного ігрового додатку жанру стратегія, присвяченого подіям Української революції 1917–1921 років[9, 15]. Використання покрокової моделі дозволить реалізувати складні механіки економіки, політики, дипломатії та військового управління, забезпечивши високий рівень стратегічної глибини та історичної достовірності[13, 14].

РОЗДІЛ 2

ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ РОЗРОБКИ ІГРОВОГО ДОДАТКУ

2.1 Архітектура комп'ютерної гри жанру стратегія

Архітектура програмного забезпечення є одним із ключових факторів успішної розробки сучасних стратегічних ігор. Від якості проектування архітектури залежить продуктивність системи, можливість подальшого розширення функціоналу, підтримка програмного коду та швидкість розробки нових механік[4, 5].

Для розроблюваної Комп'ютерної гри було обрано модульний підхід до побудови архітектури. Даний підхід дозволяє розділити функціональність гри на окремі незалежні підсистеми, кожна з яких відповідає за певну групу механік[3, 5].

Основою гри є покрокова модель управління, в якій кожен ігровий хід відповідає певному проміжку історичного часу. У межах одного ходу гравець може виконувати економічні, політичні, дипломатичні та військові дії. Після завершення ходу відбувається обробка результатів усіх систем та оновлення стану світу[13, 14].

Архітектура гри реалізується на основі компонентного підходу, який підтримується ігровим рушієм Unity[17]. Усі об'єкти гри складаються з незалежних компонентів, що дозволяє спростити розширення функціональності та повторне використання програмного коду[4, 17].

Для забезпечення масштабованості проєкту архітектура поділена на декілька основних модулів:

- модуль світу;
- модуль економіки;
- модуль населення;
- модуль інфраструктури;
- модуль політики;
- модуль дипломатії;

- модуль військової системи;
- модуль штучного інтелекту;
- модуль подій;
- модуль користувацького інтерфейсу.

Центральним елементом системи виступає модуль світу. Він відповідає за управління картою, провінціями, територіями держав та географічними характеристиками місцевості. Кожна провінція містить інформацію про населення, ресурси, інфраструктуру та належність до певної держави[22, 23].

Економічний модуль забезпечує моделювання виробництва ресурсів, податкової системи, доходів бюджету та розвитку промисловості. У грі використовуються базові економічні ресурси: продовольство, вугілля, метал, нафта та грошові кошти[14, 16].

Модуль населення відповідає за демографічні процеси та соціальну структуру суспільства. Населення поділяється на окремі соціальні групи, які мають власні потреби, рівень лояльності та політичні уподобання[14].

Модуль інфраструктури забезпечує управління транспортною мережею, залізницями, дорогами, промисловими об'єктами та іншими спорудами. Розвинена інфраструктура підвищує ефективність економіки та швидкість переміщення військ[23].

Політичний модуль реалізує систему державного управління. Він включає механізми роботи уряду, політичних партій, реформ, внутрішньої стабільності та громадської підтримки влади[9, 15].

Дипломатичний модуль забезпечує взаємодію між державами. У його межах реалізуються дипломатичні відносини, укладання союзів, оголошення війни, підписання мирних договорів та проведення переговорів[23, 24].

Військовий модуль відповідає за створення армій, пересування військових підрозділів, бойові дії та систему постачання. Кожна військова частина характеризується чисельністю, рівнем підготовки, бойовим досвідом та запасами ресурсів[22, 25].

Модуль штучного інтелекту використовується для керування країнами, які

не контролюються гравцем. Система аналізує поточний стан держави та приймає рішення щодо економічного розвитку, дипломатії та військових дій[2, 7].

Модуль подій реалізує історичні та випадкові події. Події можуть змінювати політичну ситуацію, впливати на економіку або викликати військові конфлікти[13, 14].

Для взаємодії користувача з грою використовується модуль інтерфейсу. Він забезпечує відображення інформації про провінції, держави, ресурси, армії та дипломатичні відносини[1, 10].

Взаємодія між модулями здійснюється через централізовану систему менеджерів. Такий підхід дозволяє уникнути надмірної залежності між окремими компонентами та спрощує підтримку програмного коду[3, 5].

Для зберігання даних використовуються ScriptableObject-об'єкти Unity[20, 21]. Вони містять інформацію про держави, провінції, ресурси, будівлі та інші елементи гри. Використання ScriptableObject дозволяє відокремити дані від програмної логіки та спростити їх редагування[20, 21].

Запропонована архітектура забезпечує можливість поступового розширення функціоналу гри та відповідає вимогам до сучасних покрокових стратегій[4, 14]. Модульний підхід дозволяє незалежно розвивати окремі системи без необхідності кардинальної зміни структури проєкту[5, 8].

2.2 Обґрунтування інструментальних та програмних засобів розробки стратегічної гри

Визначимо роль ігрового рушія в сучасній розробці стратегічних ігор. Ігровий рушій є фундаментом будь-якої сучасної Комп'ютерної гри. Саме він забезпечує взаємодію між усіма компонентами програмного продукту, включаючи графічне відображення, обробку подій користувача, виконання програмного коду, керування ресурсами та збереження ігрового стану[4].

У минулому більшість ігор створювалися на власних рушіях, які розроблялися спеціально під конкретний проєкт. Такий підхід вимагав значних

витрат часу та ресурсів, оскільки розробникам доводилося самотійно реалізовувати базові механізми роботи гри[4].

Сучасні ігрові рушії дозволяють зосередитися безпосередньо на створенні ігрової логіки та контенту. Вони надають готові засоби роботи з графікою, фізикою, інтерфейсами та системою керування даними[17].

Особливо важливим це є для жанру стратегічних ігор, де значна частина складності припадає не на графічні ефекти, а на реалізацію великої кількості взаємопов'язаних механік. Для покрокової стратегії необхідно одночасно підтримувати роботу багатьох систем:

- політики;
- дипломатії;
- економіки;
- військової справи;
- ресурсного забезпечення;
- управління територіями;
- будівництва;
- наукового розвитку.

Тому використання сучасного рушія дозволяє суттєво скоротити час розробки та підвищити якість кінцевого програмного продукту[4].

Перед вибором Unity було проведено аналіз декількох популярних рушіїв.

Серед найбільш поширених платформ можна виділити:

- Unity;
- Unreal Engine;
- Godot Engine.

Unreal Engine є одним із найпотужніших рушіїв сучасності та широко використовується для створення високобюджетних проєктів. Його головною перевагою є високоякісна графіка та потужна система візуального програмування Blueprint. Проте для розробки проєкту використання Unreal Engine є менш доцільним через високі системні вимоги та орієнтацію переважно на тривимірні ігри.

Godot Engine є безкоштовним рушієм з відкритим кодом, який активно розвивається останніми роками. Він відрізняється простотою використання та невеликими системними вимогами.

Однак Godot поступається Unity за кількістю навчальних матеріалів, готових рішень та рівнем поширення серед професійних розробників[17]. Результати порівняння наведені в таблиці 2.1.

Таблиця 2.1 – Порівняння популярних ігрових рушіїв

Критерій	Unity	Unreal Engine	Godot
Простота освоєння	Висока	Середня	Висока
Підтримка 2D	Висока	Середня	Висока
Підтримка 3D	Висока	Дуже висока	Висока
Документація	Дуже велика	Велика	Середня
Спільнота розробників	Дуже велика	Велика	Середня
Підходить для стратегій	Так	Так	Так
Інтеграція з C#	Повна	Обмежена	Часткова

На основі проведеного аналізу було обрано Unity як найбільш збалансоване рішення для реалізації поставлених завдань[17, 18].

У проєкті бакалаврської роботи Unity використовується не лише як графічний рушій, але й як основна платформа керування всіма компонентами гри. Однією з найважливіших переваг Unity є використання концепції сцен[17].

Сцена являє собою окремий етап роботи програми та містить набір об'єктів, необхідних для реалізації певного функціоналу.

У межах проєкту використовуються такі сцени:

- головне меню;
- меню вибору держави;
- основна ігрова карта;
- службові сцени тестування.

Кожна сцена містить власний набір об'єктів та інтерфейсних елементів.

Таке розділення дозволяє оптимізувати використання пам'яті та спростити підтримку програмного коду[17].

Під час розробки програмного продукту використовується об'єктно-орієнтована парадигма програмування. Основна ідея цього підходу полягає у представленні елементів гри у вигляді окремих об'єктів[5, 6]. Наприклад:

- країна представляється класом `CountryData`;
- провінція представляється класом `ProvinceData`;
- військова частина представляється класом `UnitData`;
- дипломатичні відносини представляються окремими структурами даних.

Подібний підхід дозволяє створювати логічно незалежні компоненти та значно полегшує масштабування проєкту. У майбутньому до гри можна буде додавати нові механіки без необхідності повного переписування існуючого коду[5].

Саме тому об'єктно-орієнтований підхід є стандартом при розробці сучасних стратегічних ігор[6].

Обґрунтованими є переваги використання `ScriptableObject`. У стратегічних іграх кількість даних може бути дуже великою. Наприклад, для кожної держави необхідно зберігати:

- назву;
- прапор;
- форму правління;
- економічні показники;
- військову силу;

- дипломатичний статус.

Для кожної провінції необхідно зберігати:

- населення;
- ресурси;
- інфраструктуру;
- рівень розвитку;
- власника території.

Якщо всі ці дані зберігати безпосередньо у коді, підтримка проєкту стане складною. Тому в Unity використовується механізм ScriptableObject[20]. Він дозволяє винести дані в окремі файли ресурсів та працювати з ними незалежно від програмної логіки. У результаті розробник може змінювати характеристики об'єктів без модифікації програмного коду. Це особливо важливо для стратегічних ігор, де балансування показників є постійним процесом[20, 21].

Visual Studio Code використовується як основне середовище написання програмного коду. Популярність даного редактора пояснюється його гнучкістю та високою продуктивністю. На відміну від повноцінних IDE, Visual Studio Code має невелике споживання оперативної пам'яті та швидко запускається навіть на відносно слабких комп'ютерах. Під час розробки проєкту використовуються такі можливості редактора:

- автоматичне доповнення коду;
- підсвічування синтаксису;
- навігація між класами;
- швидкий пошук помилок;
- інтеграція з Unity;
- аналіз структури проєкту.

Завдяки цьому значно підвищується швидкість розробки та зменшується кількість програмних помилок[19].

Сформуємо вимоги до технологій розробки стратегічної гри. До них належать:

1. Підтримка великих обсягів даних.

2. Можливість реалізації складної ігрової логіки.
3. Простота масштабування проєкту.
4. Наявність документації та навчальних матеріалів.
5. Підтримка об'єктно-орієнтованого програмування.
6. Можливість створення великої кількості інтерфейсних елементів.
7. Підтримка системи збережень.
8. Висока стабільність роботи.

Проведений аналіз показав, що обрана комбінація Unity, C#, ScriptableObject та Visual Studio Code повністю відповідає зазначеним вимогам[17-21].

2.3 Алгоритми розробки рушія гри

Одним із ключових етапів створення будь-якої комп'ютерної гри є проєктування та реалізація ігрового рушія. Саме рушій забезпечує функціонування основних механік гри, обробку взаємодії користувача з ігровим світом, управління ігровими об'єктами, виконання розрахунків та оновлення стану системи[4].

На відміну від динамічних ігор у реальному часі, покрокові стратегії мають власні особливості побудови внутрішньої логіки. Основне навантаження припадає не на графічні обчислення чи фізичне моделювання, а на обробку великої кількості взаємопов'язаних систем, які повинні коректно працювати між собою[1, 14].

У процесі розробки гри було реалізовано спеціалізований ігровий рушій, орієнтований на управління такими системами:

- система держав;
- система провінцій;
- економічна система;
- система ресурсів;
- військова система;
- дипломатична система;

- система будівництва;
- система ігрового часу;
- система подій;
- система користувацького інтерфейсу.

Розглянемо загальну схему роботи рушія. Після запуску гри відбувається послідовна ініціалізація всіх основних компонентів. На першому етапі завантажуються глобальні менеджери, що відповідають за роботу окремих підсистем[5, 8]. До них належать:

- GameManager;
- CountryManager;
- TimeManager;
- TurnManager;
- ResourceManager;
- DiplomacyManager;
- ConstructionManager.

Після створення менеджерів відбувається завантаження даних про держави та провінції із файлів ScriptableObject[20, 21].

На наступному етапі створюються ігрові об'єкти карти та відображаються початкові дані поточної кампанії. Схему роботи рушія можна представити таким чином: Запуск гри → Ініціалізація менеджерів → Завантаження даних → Створення карти → Запуск ігрового циклу → Обробка дій гравця → Завершення ходу → Оновлення стану світу[4].

Однією з найважливіших систем покрокової стратегії є система часу. На відміну від стратегій реального часу, усі зміни у світі відбуваються після завершення ходу. Для цього було реалізовано окремий менеджер часу[17].

Основними функціями системи є:

- зберігання поточної дати;
- облік ходів;
- зміна місяців та років;
- запуск економічних розрахунків;

- оновлення дипломатичних відносин;
- активація історичних подій.

Після натискання кнопки завершення ходу виконується наступний алгоритм:

1. Перевірка коректності завершення ходу.
2. Збереження поточного стану.
3. Оновлення дати.
4. Виконання економічних розрахунків.
5. Оновлення ресурсів держав.
6. Оновлення будівництва.
7. Оновлення військових підрозділів.
8. Перевірка дипломатичних змін.
9. Запуск подій.
10. Передача управління наступному ходу.

Такий підхід забезпечує передбачувану та контрольовану логіку роботи всієї гри[5, 6].

Розглянемо алгоритм управління провінціями. Провінція є базовою територіальною одиницею ігрового світу. Кожна провінція містить набір параметрів:

- назва;
- власник;
- населення;
- столиця;
- міста;
- виробництво продовольства;
- виробництво вугілля;
- виробництво сталі;
- дохід.

Усі характеристики провінції зберігаються у відповідному об'єкті ProvinceData[20]. При виборі провінції виконується наступний алгоритм:

1. Отримання натискання користувача.
2. Визначення об'єкта провінції.
3. Завантаження даних провінції.
4. Передача даних до панелі інтерфейсу.
5. Відображення інформації користувачу.

Такий механізм дозволяє швидко працювати навіть із великою кількістю територій на карті[17].

Алгоритм економічної системи. Економічна система відповідає за формування ресурсної бази держави. Під час завершення кожного ходу виконується розрахунок виробництва ресурсів[14]. Загальний алгоритм виглядає наступним чином:

1. Отримання списку провінцій держави.
2. Підрахунок виробництва кожної провінції.
3. Врахування побудованих споруд.
4. Розрахунок бонусів.
5. Формування підсумкових показників.
6. Додавання ресурсів до державного балансу.

У поточній версії гри використовуються такі ресурси:

- продовольство;
- вугілля;
- сталь;
- фінанси.

Надалі перелік ресурсів може бути розширений.

Алгоритм військової системи відтворює управління військами. Військова система забезпечує створення та управління військовими підрозділами[7].

Кожна армія має власний набір характеристик:

- чисельність;
- атака;
- захист;
- очки здоров'я;

- очки пересування;
- приналежність до держави.

При створенні нового підрозділу виконується алгоритм:

1. Вибір провінції.
2. Перевірка ресурсів.
3. Створення військового об'єкта.
4. Ініціалізація характеристик.
5. Відображення армії на карті.

Під час ходу армія може виконувати пересування між провінціями в межах доступних очок руху[7].

Алгоритм дипломатичної системи. Дипломатична система відповідає за взаємовідносини між державами. Для кожної пари країн зберігається окремий рівень відносин[14]. Він може змінюватися залежно від:

- дипломатичних дій;
- історичних подій;
- воєнних конфліктів;
- політичних рішень.

Алгоритм зміни відносин виглядає наступним чином:

1. Вибір дипломатичної дії.
2. Визначення країни-цілі.
3. Розрахунок впливу дії.
4. Оновлення значення відносин.
5. Перевірка порогових значень.
6. Зміна дипломатичного статусу.

Подібна структура дозволяє у майбутньому реалізувати союзи, гарантії незалежності, пакти про ненапад та інші дипломатичні механіки[22, 23, 24].

Алгоритм системи будівництва. Будівництво є одним із ключових інструментів розвитку держави. Для кожної провінції зберігається список побудованих споруд[14]. При зведенні нової будівлі виконується наступна послідовність дій:

1. Вибір провінції.
2. Вибір типу споруди.
3. Перевірка ресурсів.
4. Перевірка доступності будівництва.
5. Додавання будівлі до списку.
6. Зміна економічних показників провінції.

Результат будівництва починає впливати на економіку після завершення поточного ходу.

Алгоритм роботи користувацького інтерфейсу. Користувацький інтерфейс реалізований на основі системи Canvas рушія Unity[17]. Основними елементами інтерфейсу є:

- панель провінції;
- панель держави;
- панель дипломатії;
- панель будівництва;
- панель військ;
- верхня інформаційна панель.

При взаємодії користувача з ігровими об'єктами відбувається автоматичне оновлення відповідних елементів інтерфейсу. Завдяки використанню модульної архітектури кожна панель працює незалежно від інших компонентів гри[5, 8].

Отже, реалізовані алгоритми формують основу спеціалізованого рушія покрокової історичної стратегії. Запропонована архітектура забезпечує модульність, масштабованість та можливість подальшого розширення функціональних можливостей гри шляхом додавання нових механік та підсистем[4, 5, 8].

2.4 Вимоги до ігрового додатку

Під час розробки програмного забезпечення важливим етапом є формування вимог до майбутнього продукту. Саме вимоги визначають

функціональні можливості системи, її технічні характеристики, обмеження та критерії якості[5,6].

Для стабільної роботи програмного продукту було сформовано комплекс функціональних, програмних та апаратних вимог.

Функціональні вимоги – визначають перелік можливостей, які повинен забезпечувати ігровий додаток[4]. Розроблена гра повинна надавати користувачу можливість:

- створювати нову ігрову кампанію;
- обирати державу для проходження;
- взаємодіяти з картою світу;
- переглядати інформацію про провінції;
- переглядати інформацію про держави;
- здійснювати будівництво споруд;
- керувати ресурсами держави;
- створювати військові підрозділи;
- переміщувати армії між провінціями;
- взаємодіяти з дипломатичною системою;
- завершувати хід;
- отримувати оновлення стану держави після завершення ходу;
- працювати з користувацьким інтерфейсом без необхідності;
- використання стороннього програмного забезпечення.

Кожна з перелічених функцій повинна працювати незалежно від інших підсистем та коректно взаємодіяти із загальною логікою гри[5].

Нефункціональні вимоги визначають характеристики якості програмного продукту. До основних нефункціональних вимог належать:

- стабільність роботи;
- швидке завантаження гри;
- коректне відображення графічного інтерфейсу;
- масштабованість архітектури;
- зручність користування;

- можливість подальшого розширення функціоналу;
- підтримка сучасних версій операційної системи Windows.

Особлива увага приділяється модульності програмної архітектури, що дозволяє розширювати гру новими механіками без необхідності повної переробки існуючого коду[5, 8].

Вимоги до програмного забезпечення. Для запуску гри користувач повинен мати встановлене програмне забезпечення, що підтримує виконання застосунків, створених за допомогою рушія Unity[17, 18]. Програмні вимоги до ігрового додатку наведено в таблиці 2.2.

Таблиця 2.2 – Програмні вимоги до ігрового додатку

Компонент	Мінімальна вимога
Операційна система	Windows 10
Архітектура ОС	64-bit
Графічний API	DirectX 11
.NET Runtime	Вбудований у Unity
Драйвер відеокарти	Актуальна версія

Для коректної роботи рекомендується використовувати останні оновлення операційної системи та драйверів обладнання.

Оскільки розроблювана гра є двовимірною покроковою стратегією, її системні вимоги є порівняно невисокими. Мінімальні системні вимоги до апаратного забезпечення наведено в таблиці 2.3.

Таблиця 2.3 – Мінімальні системні вимоги

Компонент	Характеристика
Процесор	Intel Core i3 або аналог
Оперативна пам'ять	4 ГБ
Відеокарта	1 ГБ VRAM
Місце на диску	2 ГБ
Роздільна здатність екрана	1366×768

Рекомендовані системні вимоги наведені в таблиці 2.4. Вказані характеристики забезпечують комфортну роботу гри та стабільну частоту кадрів під час взаємодії з великою кількістю ігрових об'єктів.

Таблиця 2.4 – Рекомендовані системні вимоги

Компонент	Характеристика
Процесор	Intel Core i5 або аналог
Оперативна пам'ять	8 ГБ
Відеокарта	2 ГБ VRAM
Місце на диску	5 ГБ
Роздільна здатність екрана	1920×1080

Інтерфейс користувача повинен забезпечувати швидкий доступ до всіх основних функцій гри. Основними вимогами до інтерфейсу користувача є [1, 10]:

- зрозуміла структура меню;
- логічне розташування елементів керування;
- швидкий доступ до інформації про провінції;
- швидкий доступ до інформації про державу;
- можливість роботи з дипломатичними функціями;
- можливість роботи з будівництвом;
- зручне управління військами;
- відображення ресурсів у режимі реального часу.

Інтерфейс повинен бути інтуїтивно зрозумілим навіть для користувачів, які раніше не грали в стратегічні ігри [13].

Вимоги до зберігання даних. Для забезпечення коректної роботи гри необхідно організувати централізоване зберігання даних. У розробленому проєкті для цього використовуються об'єкти ScriptableObject [20, 21]. У такому форматі зберігаються:

- дані держав;

- дані провінцій;
- характеристики будівель;
- параметри військових підрозділів;
- дипломатичні налаштування.

Такий підхід дозволяє спростити модифікацію контенту без внесення змін до програмного коду.

Вимоги до надійності та масштабованості. Однією з ключових вимог є можливість подальшого розвитку гри[4, 5]. Архітектура проєкту повинна забезпечувати:

- додавання нових держав;
- додавання нових провінцій;
- додавання нових типів будівель;
- додавання нових військових формувань;
- розширення дипломатичної системи;
- реалізацію нових історичних сценаріїв.

Завдяки модульному підходу кожна підсистема може розвиватися незалежно від інших частин проєкту[6, 8].

Отже, сформований комплекс вимог визначає функціональні можливості, технічні характеристики та критерії якості розроблюваного програмного продукту. Виконання зазначених вимог забезпечує стабільну роботу гри, зручність використання та можливість подальшого розширення функціоналу в процесі розвитку проєкту[4, 5, 17].

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ІГРОВОГО ДОДАТКУ ЖАНРУ СТРАТЕГІЯ

3.1 Розробка дизайну комп'ютерної гри

Одним із ключових етапів створення будь-якого програмного продукту є розробка його дизайну та користувацького інтерфейсу. Для стратегічних ігор інтерфейс відіграє особливо важливу роль, оскільки користувач постійно взаємодіє з великою кількістю інформації, що стосується управління державою, провінціями, військами та ресурсами[1, 13].

Під час розробки гри було обрано концепцію мінімалістичного двовимірного інтерфейсу, орієнтованого на зручність використання та швидкий доступ до основних ігрових механік. Основна увага приділялася читабельності інформації та зменшенню кількості зайвих елементів на екрані[10, 16].

Гра реалізована у вигляді політичної карти, розділеної на окремі провінції. Кожна провінція є незалежним об'єктом, що містить власний набір характеристик, включаючи назву, населення, столицю, міста та економічні показники[14].

Після запуску гри користувач потрапляє до головного меню, яке забезпечує доступ до створення нової гри та вибору держави.

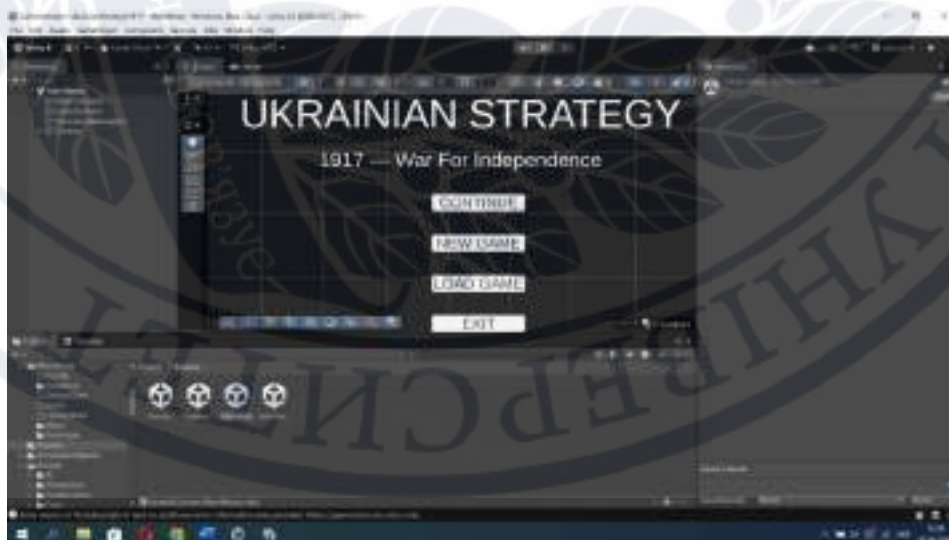


Рисунок 3.1 – Головне меню гри

Після натискання кнопки початку гри відкривається меню вибору держави. У даному меню користувач може обрати одну із доступних країн для подальшого проходження кампанії.

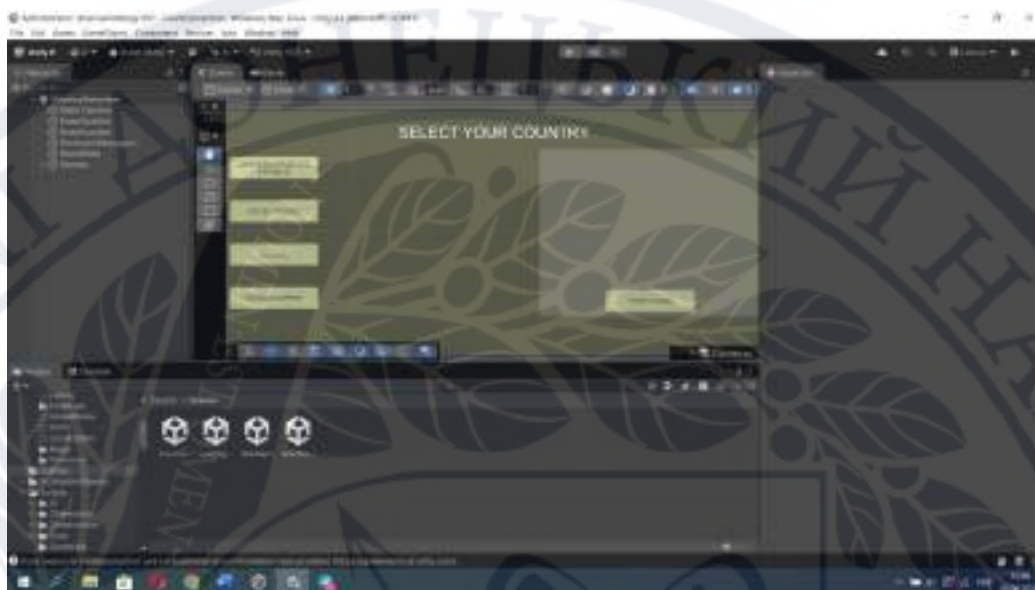


Рисунок 3.2 – Вікно вибору держави

Після завершення вибору держави відбувається завантаження основної ігрової сцени. Центральним елементом інтерфейсу є політична карта. Кожна провінція відображається окремим графічним об'єктом та має власний колір відповідно до держави-власника.



Рисунок 3.3 – Основна ігрова карта

Для відображення інформації про провінцію реалізовано окрему інформаційну панель. Після вибору провінції користувач отримує доступ до основних статистичних показників.

На панелі відображаються:

- назва провінції;
- держава-власник;
- населення;
- столиця;
- перелік міст;
- побудовані споруди.

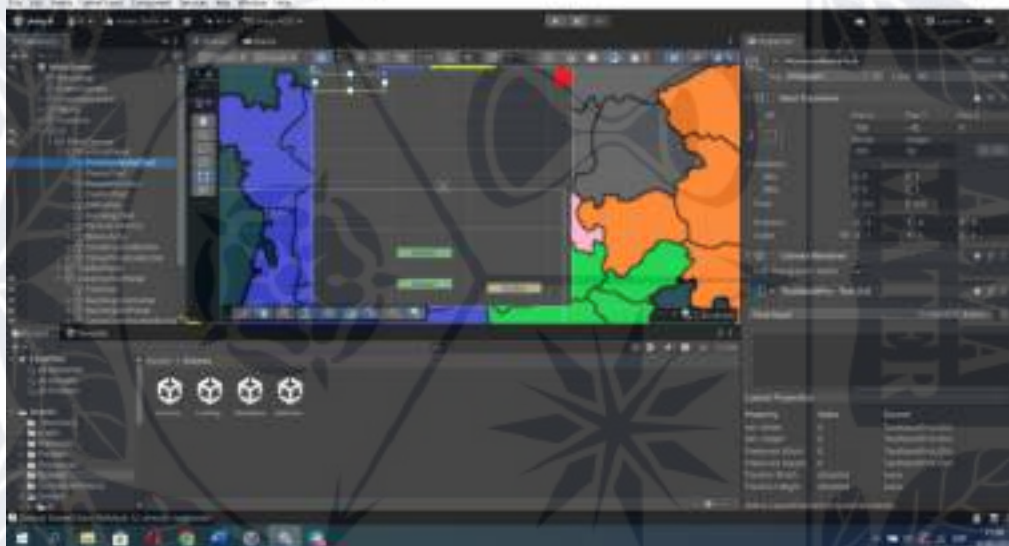


Рисунок 3.4 – Панель інформації про провінцію

Окремою складовою інтерфейсу є система будівництва. Вона дозволяє здійснювати будівництво нових споруд у вибраній провінції.

Користувач може переглядати список доступних будівель та запускати процес їх спорудження (рис. 3.5).

Для реалізації військової складової було створено окрему систему відображення армій на карті. Військові підрозділи представлені окремими об'єктами, що можуть бути вибрані користувачем.

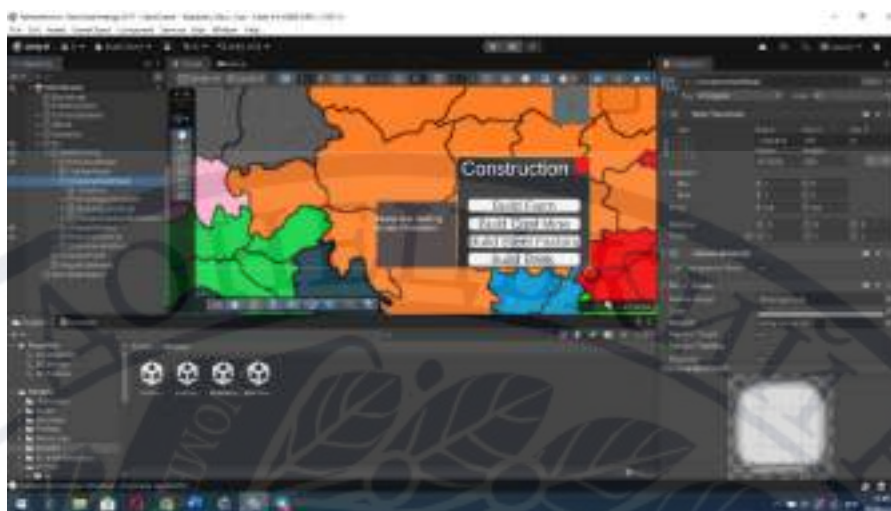


Рис. 3.5 – Панель будівництва

Після вибору армії відображається панель із характеристиками військового формування. На панелі можуть відображатися:

- назва підрозділу;
- чисельність особового складу;
- показники атаки;
- показники захисту;
- запас здоров'я;
- залишок очок пересування.

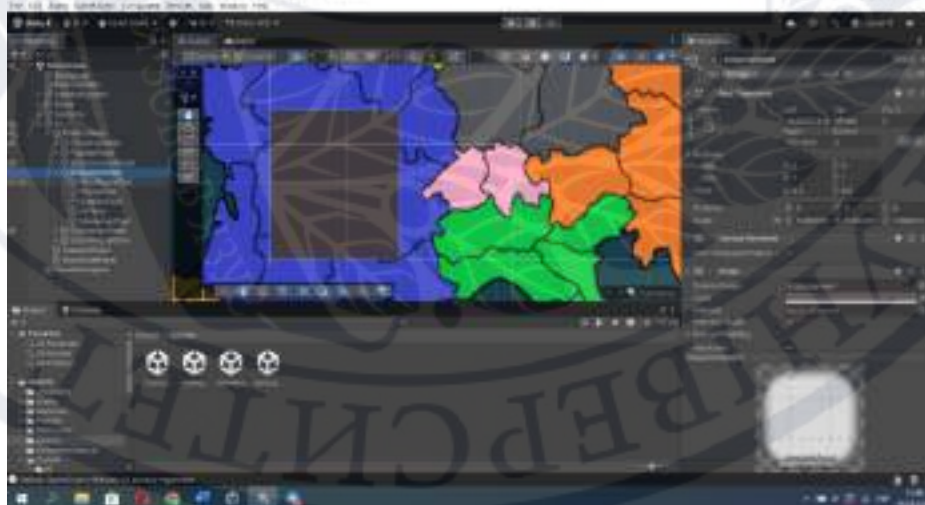


Рисунок 3.6 – Інформаційна панель армії

Для керування міжнародними відносинами реалізовується дипломатична система. Вона забезпечує відображення держав та поточних відносин між ними.

Користувач може переглядати дипломатичний статус інших країн та взаємодіяти з ними через спеціальне дипломатичне меню[22-24].

При розробці інтерфейсу використовувалися стандартні компоненти Unity UI, що дозволило забезпечити адаптивність та простоту подальшого розширення функціоналу[17, 18]. Усі інформаційні панелі реалізовані за єдиним стилем оформлення. Це забезпечує візуальну цілісність проєкту та покращує сприйняття інформації користувачем.

Таким чином, у процесі розробки було створено базовий дизайн стратегічної гри, що забезпечує зручний доступ до всіх реалізованих ігрових механік та дозволяє користувачу ефективно взаємодіяти з елементами ігрового світу[13, 16].

3.2 Програмна реалізація основних компонентів ігрового додатку

Після завершення етапу проєктування було розпочато програмну реалізацію основних механік гри. Розробка виконувалася із використанням рушія Unity та мови програмування C#[17-20].

Основною вимогою до архітектури програмного забезпечення була модульність. Такий підхід дозволяє ізолювати окремі підсистеми гри та забезпечує можливість подальшого розширення функціоналу без внесення суттєвих змін до вже існуючого програмного коду[4-6].

Структура проєкту побудована за принципом розділення відповідальностей між окремими модулями. Кожна підсистема відповідає за конкретну частину ігрової логіки[5, 6].

Основними підсистемами проєкту є:

- система управління державами;
- система управління провінціями;
- система будівництва;
- система військ;

- система дипломатії;
- система інтерфейсу користувача;
- система керування грою.

Реалізація системи зберігання даних. Для зберігання інформації про країни, провінції, будівлі та інші об'єкти використовуються `ScriptableObject`[20, 21].

`ScriptableObject` є спеціальним типом об'єктів Unity, який дозволяє зберігати дані незалежно від сцен та ігрових об'єктів[20].

Використання такого підходу дозволяє:

- централізовано керувати даними;
- зменшити кількість дублювання інформації;
- спростити редагування параметрів через Inspector;
- підвищити масштабованість проєкту[20, 21].

Для кожної держави створюється окремий об'єкт `CountryData`, що містить основні характеристики країни. До таких характеристик належать:

- назва держави;
- прапор;
- кількість населення;
- стартові ресурси;
- дипломатичні параметри.

Аналогічно реалізовано структуру `ProvinceData` для зберігання інформації про провінції. Провінція містить:

- назву;
- власника;
- населення;
- столицю;
- перелік міст;
- економічні показники;
- виробничі характеристики.



Рисунок 3.7 – Приклад налаштування об'єкта ProvinceData

Реалізація системи управління державами. Для централізованого управління країнами використовується клас CountryManager. Даний компонент автоматично завантажує всі об'єкти CountryData із каталогу Resources під час запуску гри. Після завантаження формується внутрішній список держав, який використовується іншими підсистемами[17, 20]. Основними функціями менеджера є:

- завантаження даних держав;
- пошук держав за назвою;
- отримання списку провінцій держави;
- забезпечення доступу до інформації про країни.

Застосування патерна Singleton дозволяє отримувати доступ до менеджера з будь-якої частини програми через єдиний екземпляр[3, 8].

Реалізація системи управління провінціями. Однією з центральних механік гри є взаємодія з провінціями. Для реалізації цієї функціональності створено клас ProvinceView. Кожна провінція представлена окремим об'єктом на карті та має власний компонент ProvinceView[4, 17].

Після натискання на провінцію виконується:

- визначення вибраного об'єкта;
- завантаження даних провінції;
- відкриття інформаційної панелі;
- оновлення відображення даних.



Рисунок 3.8 – Робота системи управління державами

Також реалізовано механізм зміни кольору провінції відповідно до держави-власника. Колір використовується для візуального розмежування територій різних країн[10, 13].



Рисунок 3.9 – Взаємодія користувача з провінцією

Реалізація системи будівництва. Для розвитку провінцій у грі реалізована система будівництва. Система дозволяє створювати нові споруди в межах вибраної провінції. Кожна будівля містить набір характеристик:

- назву;
- вартість;
- час будівництва;
- економічний ефект.

Після завершення будівництва споруда додається до списку побудованих об'єктів провінції. Інформація про побудовані споруди відображається в інформаційній панелі провінції[14, 16].



Рисунок 3.10 – Робота системи будівництва

Таким чином було реалізовано основні компоненти гри, що забезпечують управління державами, провінціями та економічним розвитком територій. Побудована архітектура дозволяє легко інтегрувати нові механіки та розширювати функціональність проєкту в майбутньому[4-6, 21].

3.3 Тестування додатку жанру стратегія

Однією з найважливіших складових будь-якої стратегічної гри є військова система. Саме вона забезпечує можливість ведення бойових дій, захисту територій та реалізації зовнішньополітичних цілей держави. У розробленому програмному додатку військові формування представлені окремими ігровими об'єктами, що розміщуються безпосередньо на карті світу.

Для реалізації військових підрозділів було створено клас ArmyView, який відповідає за візуальне представлення армії та взаємодію користувача з військовими формуваннями. Кожна армія має власний набір характеристик, що зберігається в структурі UnitData. До основних характеристик військового підрозділу належать:

- назва підрозділу;
- країна-власник;
- поточна провінція розташування;
- чисельність особового складу;
- показники атаки;
- показники захисту;
- запас здоров'я;
- максимальна кількість очок пересування;
- поточний запас очок пересування.

При створенні нового підрозділу виконується його ініціалізація та встановлення базових характеристик. Для кожної країни передбачено власне кольорове позначення армій на карті, що дозволяє користувачу швидко визначати приналежність військ[17].

Після вибору армії її колір змінюється, а інформація про підрозділ передається до інформаційної панелі інтерфейсу (рис. 3.11). Панель інформації про армію наведена на рисунку 3.12. Такий підхід забезпечує зручну взаємодію між користувачем та військовими об'єктами[1, 10].

Реалізація дипломатичної системи. Для відображення міжнародних відносин між державами у проєкті реалізовується базова дипломатична система.

Основним завданням дипломатичного модуля є забезпечення взаємодії між державами та відображення їхнього поточного статусу[13, 14].

У структурі дипломатичної системи кожна країна може підтримувати відносини з іншими державами[22, 23, 24].

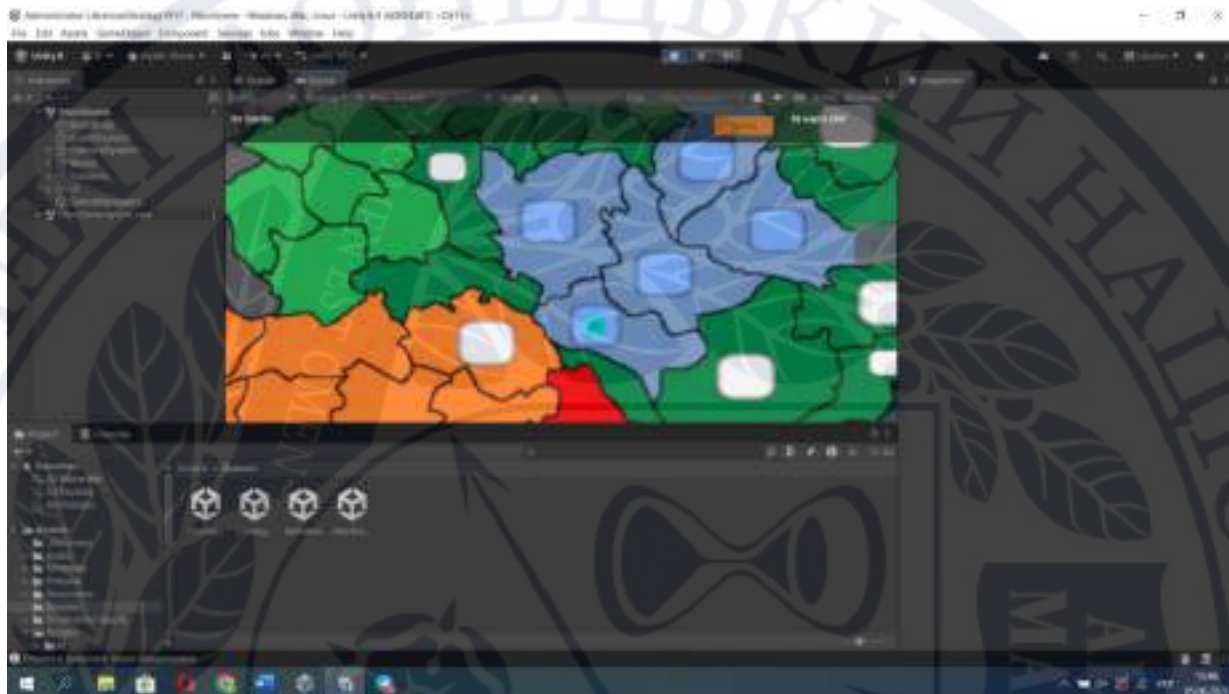


Рисунок 3.11 – Відображення армій на карті

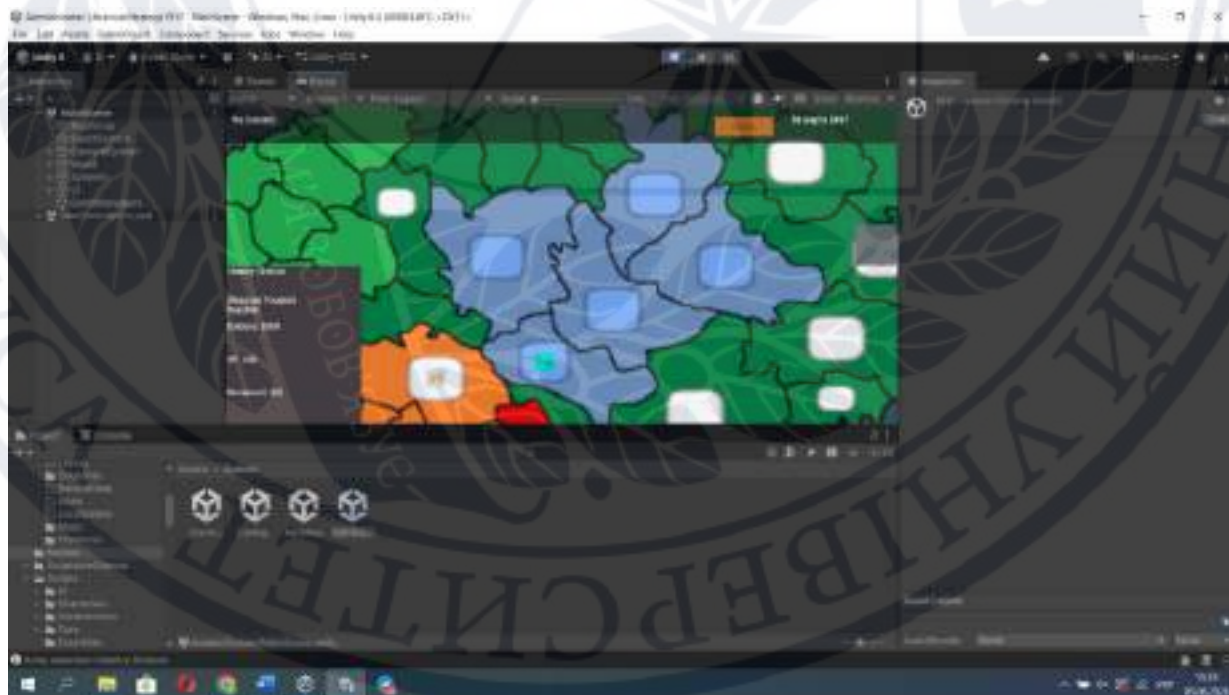


Рисунок 3.12 – Панель інформації про армію

Для зручності користувача створено окрему дипломатичну панель, яка відображає список доступних країн та інформацію про дипломатичні відносини. Після вибору країни користувач отримує доступ до таких параметрів:

- назва держави;
- рівень дипломатичних відносин;
- поточний дипломатичний статус;
- можливі дипломатичні дії.

У подальшому дипломатична система може бути розширена шляхом додавання механік союзів, оголошення війни, підписання мирних договорів, торговельних угод та інших інструментів міжнародної політики[11, 12, 22].

Базова реалізація дипломатичного модуля вже забезпечує необхідний фундамент для подальшого розвитку функціоналу гри[13].

Реалізація системи керування грою. Для централізованого керування основними процесами гри використовується набір спеціалізованих менеджерів.

Головним координатором ігрового процесу є GameManager. Даний компонент відповідає за ініціалізацію ключових підсистем гри та забезпечує їхню спільну роботу. Під час запуску гри відбувається створення та налаштування основних менеджерів, після чого система переходить до робочого режиму[4]. Для виконання початкової ініціалізації використовується BootstrapManager.

Основними функціями цього компонента є:

- запуск процесу завантаження гри;
- створення менеджерів;
- перевірка коректності ініціалізації систем;
- підготовка ігрового середовища.

Для керування ігровим часом використовується TimeManager.

Компонент забезпечує зберігання поточної дати та дозволяє реалізовувати механіку зміни часу[4].

На основі TimeManager функціонує TurnManager. Даний модуль

відповідає за реалізацію покрокової механіки гри. Після завершення ходу виконується оновлення ігрового стану та перерахунок необхідних параметрів.

Застосування покрокової системи дозволяє користувачу ретельно аналізувати ситуацію та приймати стратегічні рішення без часових обмежень. Саме тому покрокова модель була обрана як основа розроблюваної гри[10, 14].

Структура програмного проєкту. Для забезпечення масштабованості програмного забезпечення було використано модульну структуру каталогів. Основні програмні модулі розміщені в каталозі Scripts та поділені на окремі функціональні підсистеми. Серед них:

- Core;
- World;
- Factions;
- Economy;
- Military;
- Diplomacy;
- Politics;
- UI.

Такий підхід дозволяє ізолювати функціональність окремих модулів та спрощує подальшу підтримку програмного коду (додаток А, рис. А.1)[8].

Таким чином у процесі розробки було реалізовано основні програмні компоненти стратегічної гри, включаючи систему військ, дипломатію, керування ігровим процесом та модульну архітектуру проєкту. Створене програмне забезпечення має гнучку структуру та може бути розширене шляхом додавання нових механік без суттєвої зміни існуючого коду[13].

Після завершення реалізації основних компонентів гри було проведено тестування програмного забезпечення з метою перевірки працездатності реалізованих механік та виявлення можливих помилок під час взаємодії між окремими підсистемами проєкту.

Тестування виконувалося безпосередньо в середовищі Unity із використанням вбудованих інструментів налагодження та журналювання подій

(Console). Основна увага приділялася перевірці коректності роботи систем управління провінціями, будівництва, військових формувань, дипломатії та інтерфейсу користувача.

У процесі тестування було перевірено коректність роботи системи вибору провінцій. Після натискання на провінцію повинно відбуватися відкриття інформаційної панелі та відображення актуальних характеристик вибраної території. Проведені випробування підтвердили правильність завантаження даних із об'єктів ProvinceData та їх коректне відображення у користувацькому інтерфейсі.

Також було протестовано систему будівництва. Перевірялися сценарії створення споруд у різних провінціях, додавання побудованих об'єктів до внутрішнього списку провінції та оновлення інформації на інформаційній панелі. У результаті тестування підтверджено коректну роботу механізму будівництва та відсутність критичних помилок під час взаємодії користувача із системою[20].

Окремо було проведено тестування військової системи. Перевірялася можливість створення військових формувань, вибору армій на карті та відображення інформації про них у спеціальній інформаційній панелі. Результати тестування підтвердили правильність роботи системи відображення військових підрозділів та передачі даних між компонентами ArmyView і ArmyInfoUI.

Для дипломатичної системи було перевірено відображення списку держав та можливість перегляду інформації про дипломатичні відносини. Система коректно завантажує інформацію про держави та забезпечує її відображення у дипломатичному інтерфейсі.

Крім функціонального тестування виконувалося тестування користувацького інтерфейсу. Перевірялася робота кнопок, відкриття та закриття інформаційних панелей, коректність відображення текстових даних та адаптивність інтерфейсу під різні роздільні здатності екрана.

Результати проведеного тестування свідчать про працездатність

реалізованих механік та готовність програмного забезпечення до подальшого розвитку і розширення функціональних можливостей.

Перспективи реалізації системи штучного інтелекту

Одним із найважливіших напрямів подальшого розвитку проєкту є впровадження системи штучного інтелекту для керування комп'ютерними державами[7].

У сучасних стратегічних іграх штучний інтелект виконує функції управління економікою, дипломатією, військами та внутрішньою політикою держави. Якість роботи AI безпосередньо впливає на складність гри та рівень зацікавленості користувача.

У межах розробленої архітектури вже передбачено можливість інтеграції окремого модуля AI без необхідності зміни існуючих механік. Модульна структура проєкту дозволяє реалізувати окремі підсистеми штучного інтелекту для кожного напрямку управління державою.

Економічний штучний інтелект може відповідати за розподіл ресурсів, розвиток інфраструктури та будівництво споруд у провінціях. Основною задачею такого модуля буде забезпечення стабільного розвитку держави та ефективного використання доступних ресурсів.

Військовий штучний інтелект може виконувати аналіз військової ситуації на карті, формувати нові підрозділи, здійснювати переміщення військ між провінціями та приймати рішення щодо ведення бойових дій. Для реалізації подібної системи можуть використовуватися алгоритми оцінки стану ігрового світу та прийняття рішень на основі набору пріоритетів[7].

Дипломатичний штучний інтелект може аналізувати відносини між державами та приймати рішення щодо укладання союзів, підписання торговельних угод або оголошення війни. Такий підхід дозволить створити більш динамічне міжнародне середовище та підвищити реалістичність симуляції.

Перспективним напрямом також є використання поведінкових моделей держав. У цьому випадку кожна країна може отримати власний набір

характеристик, що визначатиме її стиль поведінки на міжнародній арені. Наприклад, окремі держави можуть бути більш агресивними, інші — орієнтованими на дипломатію або економічний розвиток.

Запровадження системи штучного інтелекту дозволить значно підвищити складність ігрового процесу та забезпечити більш реалістичне моделювання міжнародних відносин і розвитку держав.

Таким чином, створена архітектура проєкту вже зараз забезпечує можливість подальшого впровадження повноцінної системи штучного інтелекту, що стане одним із ключових етапів розвитку програмного продукту.

ВИСНОВКИ

У результаті виконання бакалаврської роботи було досліджено особливості розробки комп'ютерних ігор жанру стратегія та реалізовано прототип історичної покрокової стратегічної гри, присвяченої подіям Української революції 1917–1921 років. За результатами дослідження отримано наступні висновки:

1. Дослідження теоретико-концептуальних засад розробки ігор жанру стратегія показало, що жанр характеризується значною різноманітністю механік та підходів до реалізації ігрового процесу. Основними типами стратегій є: RTS (дії в реальному часі), TBS (покрокове управління), Grand Strategy (управління державою), 4X (дослідження, експлуатація, знищення). Важливою характеристикою стратегічних ігор є наявність ресурсної системи, системи прийняття рішень, військова складова. Покрокові стратегічні ігри дозволяють гравцю детально аналізувати ситуацію та приймати рішення без часових обмежень.

2. Проведено аналіз сучасних стратегічних ігор, розглянуто основні механіки ігрових проектів жанру grand strategy та покрокових стратегій, а також виконано порівняння існуючих програмних рішень. Найбільш поширеними на ринку ігор є стратегічні ігри: Hearts of Iron IV, Europa Universalis IV, Victoria 3, Crusader Kings III, Civilization VI, Old World. Найбільш поширеними серед універсальних ігрових рушіїв є Unity Technologies та Epic Games Unreal Engine. На основі проведеного аналізу було сформовано концепцію власного ігрового додатку. Найбільш перспективним напрямом розвитку стратегічних ігор можна визначити поєднання історичної достовірності, складних економічних і політичних систем, а також покрокового механізму прийняття рішень.

3. Обґрунтовано вибір програмних засобів та технологій розробки, спроектовано архітектуру програмного забезпечення і створено основні функціональні модулі гри.

4. Особливу увагу приділено побудові модульної архітектури, що

забезпечує можливість подальшого розширення функціональності та вдосконалення окремих компонентів гри. Розроблене програмне забезпечення демонструє можливість створення історичної покрокової стратегії із використанням сучасних інструментів розробки ігрових додатків.

5. Реалізовано систему провінцій, механізм вибору держави, базові елементи економіки, систему військових підрозділів, інтерфейс користувача, механізм ходів та початкову систему дипломатичної взаємодії між державами.

6. Подальший розвиток проєкту може передбачати розширення дипломатичної системи, удосконалення штучного інтелекту, впровадження додаткових історичних сценаріїв, покращення графічної складової, а також реалізацію багатокористувацького режиму.

Отже, створений програмний продукт може бути використаний як основа для подальшого розвитку повноцінної історичної grand strategy-гри.

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Adams E. Fundamentals of Game Design. 3rd ed. Berkeley : New Riders, 2014. 584 p.
2. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1994. 395 p.
3. Gregory J. Game Engine Architecture. 3rd ed. Boca Raton : CRC Press, 2018. 1024 p.
4. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston : Prentice Hall, 2017. 432 p.
5. Martin R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Boston : Prentice Hall, 2008. 464 p.
6. Millington I. AI for Games. 3rd ed. Boca Raton : CRC Press, 2019. 900 p.
7. Nystrom R. Game Programming Patterns. Genever Benning, 2014. 354 p.
8. Plokhly S. The Gates of Europe: A History of Ukraine. New York : Basic Books, 2021. 432 p.
9. Rogers S. Level Up! The Guide to Great Video Game Design. 2nd ed. Hoboken : Wiley, 2014. 552 p.
10. Romero B., Schreiber I. Challenges for Game Designers. 2nd ed. Boston : Pearson, 2021. 432 p.
11. Salen K., Zimmerman E. Rules of Play: Game Design Fundamentals. Cambridge : MIT Press, 2004. 688 p.
12. Schell J. The Art of Game Design: A Book of Lenses. 3rd ed. Boca Raton : CRC Press, 2020. 935 p.
13. Sellers M. Advanced Game Design: A Systems Approach. Boston : Addison-Wesley, 2017. 576 p.
14. Subtelny O. Ukraine: A History. 4th ed. Toronto : University of Toronto Press, 2009. 784 p.

15. Sylvester T. Designing Games: A Guide to Engineering Experiences. Sebastopol : O'Reilly Media, 2013. 352 p.
16. Unity Technologies. Unity Documentation. URL: <https://docs.unity3d.com/>.
17. Unity Technologies. Unity Learn. URL: <https://learn.unity.com/>.
18. Microsoft. C# Documentation. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/>
19. Unity Technologies. ScriptableObject Manual. URL: <https://docs.unity3d.com/Manual/class-ScriptableObject.html>
20. Unity Technologies. Architect your game code with ScriptableObjects . URL: <https://unity.com/how-to/architect-game-code-scriptable-objects>.
21. Paradox Interactive. Hearts of Iron IV. URL: <https://www.paradoxinteractive.com/games/hearts-of-iron-iv/about>.
22. Paradox Interactive. Europa Universalis IV. URL: <https://www.paradoxinteractive.com/games/europa-universalis-iv/about>.
23. Paradox Interactive. Crusader Kings III. URL: <https://www.paradoxinteractive.com/games/crusader-kings-iii/about>.
24. Creative Assembly. Total War Official Website. URL: <https://www.totalwar.com/>.
25. Canadian Institute of Ukrainian Studies. Encyclopedia of Ukraine . URL: <https://www.encyclopediaofukraine.com/>.
26. Internet Encyclopedia of Philosophy. URL: <https://iep.utm.edu/>.
27. Stanford Encyclopedia of Philosophy . URL: <https://plato.stanford.edu/>.
28. Freeman D. Creating Emotion in Games: The Craft and Art of Emotioneering. Indianapolis : New Riders, 2004. 832 p.
29. Fullerton T. Game Design Workshop: A Playcentric Approach to Creating Innovative Games. 5th ed. Boca Raton : CRC Press, 2024. 600 p.
30. Crawford C. The Art of Computer Game Design. Berkeley : McGraw-Hill, 1984. 256 p.
31. Koster R. A Theory of Fun for Game Design. 3rd ed. Scottsdale : ETC

Press, 2023. 320 p.

32. Rogers S. Level Up! The Guide to Great Video Game Design. 3rd ed. Wiley, 2024. 600 p.

33. Björk S., Holopainen J. Patterns in Game Design. Hingham : Charles River Media, 2004. 423 p.

34. Totten C. W. An Architectural Approach to Level Design. Boca Raton : CRC Press, 2014. 432 p.

35. Keith C. Agile Game Development with Scrum. Boston : Addison-Wesley, 2010. 328 p.

36. Novak J. Game Development Essentials. 4th ed. Delmar Cengage Learning, 2011. 640 p.

37. Rabin S. Introduction to Game Development. 2nd ed. Cengage Learning, 2014. 1024 p.

38. Chandler H. M. The Game Production Handbook. 3rd ed. Jones & Bartlett Learning, 2013. 450 p.

39. Unity Technologies. Unity Documentation. Official documentation. Unity Documentation

40. Unity Technologies. ScriptableObject Manual. ScriptableObject Manual

41. Unity Technologies. Introduction to ScriptableObjects. Unity Learn ScriptableObjects

42. Unity Technologies. Create modular game architecture with ScriptableObjects. Architect your game code with ScriptableObjects

43. Microsoft. C# Documentation. C# Documentation

44. Microsoft. .NET Documentation. .NET Documentation

45. Ukrainian Institute of National Memory. Ukrainian Revolution of 1917–1921. Ukrainian Institute of National Memory

46. Canadian Institute of Ukrainian Studies. Encyclopedia of Ukraine. Encyclopedia of Ukraine

47. Stanford Encyclopedia of Philosophy. Stanford Encyclopedia of Philosophy



ДОДАТКИ



Рисунок А.1 – Структура програмного проєкту

Додаток Б

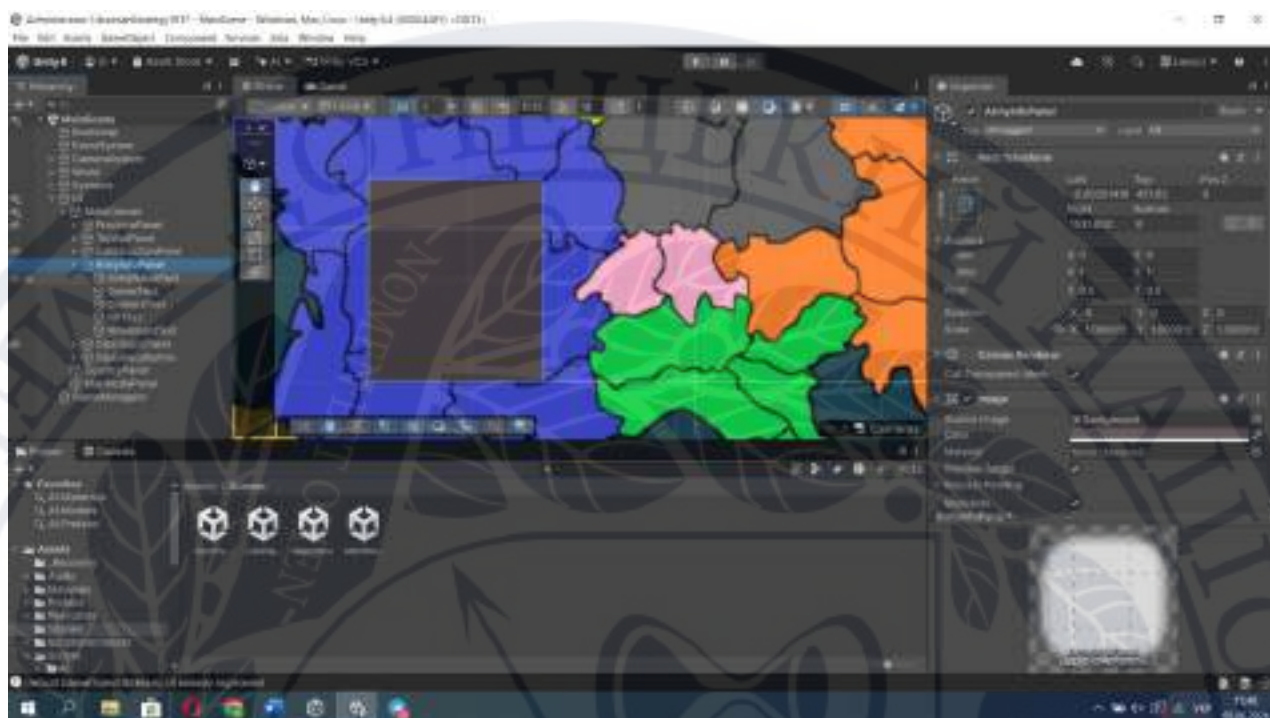


Рисунок Б.2 – Структура сцени MainScene