

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія аналітики комп'ютерної гри Dota 2»

Виконав: студент 2 курсу, групи 1КН-22м
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)


_____ Тишко В.А.
(прізвище та ініціали)

Керівник: PhD, професор каф. КН
_____ Савчук Т.О.
(прізвище та ініціали)

« 07 » _____ 12 _____ 2023 р.

Опонент: д.т.н., професор, завідувач кафедри АІТ
_____ Бісікало О.В.
(прізвище та ініціали)


« 07 » _____ 12 _____ 2023 р.

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
(прізвище та ініціали)

« 08 » _____ 12 _____ 2023 р.

Вінниця ВНТУ – 2023 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра Комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»


ЗАТВЕРДЖУЮ
Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.
"12" 08 2023 року

**ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Тишку Владиславу Анатолійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Інформаційна технологія аналітики комп'ютерної гри Dota 2.
Керівник роботи: Савчук Тамара Олександрівна, PhD, проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "18" 09 року №247

2. Строк подання студентом роботи 13.11.2023

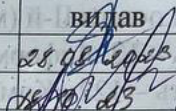
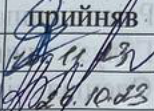
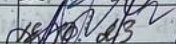
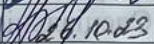
3. Вихідні дані до роботи: потужність множини персонажів – не менше 10,
потужність множини спорядження – не менше 10, потужність множини правил
– не менше 4, мова програмування – об'єктно-орієнтована.

4. Зміст текстової частини

Сучасний рівень розвитку аналітики комп'ютерної гри Dota 2; Розробка
математичної моделі, алгоритму та структури аналітики комп'ютерної гри Dota 2;
Розробка модулів інформаційної технології комп'ютерної гри Dota 2;
Економічна частина; Висновки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень):
UML-діаграма узагальненого алгоритму аналітики комп'ютерної гри Dota 2;
UML-діаграма удосконаленого алгоритму аналітики комп'ютерної гри Dota 2;
Структура інформаційної технології аналітики комп'ютерної гри Dota 2; UML-
діаграма роботи модуля збору даних; UML-діаграма роботи модуля обробки
даних; UML-діаграма роботи модуля статистичного аналізу; UML-діаграма
роботи модуля виведення даних.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Савчук Т. О., проф. каф. КН		
4	Адлер О. О., доц. каф. ЕПВМ		

7. Дата видачі завдання 29. 08. 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Сучасний рівень розвитку аналітики комп'ютерної гри Dota 2	01.09.23 - 08.09.23	
2	Розробка математичної моделі, алгоритму та структури аналітики комп'ютерної гри Dota 2	09.09.23 - 29.09.23	
3	Розробка модулів інформаційної технології аналітики комп'ютерної гри Dota 2	25.09.23 - 17.10.23	
4	Підготовка економічної частини	18.10.23 - 26.10.23	
5	Апробація та/або впровадження результатів дослідження	27.10.23 - 01.11.23	
6	Оформлення матеріалів до захисту МКР	02.11.23 - 10.11.23	

Студент


(підпис)

Тишко В. А.

Керівник роботи


(підпис)

Савчук Т. О.

АНОТАЦІЯ

УДК 004.8

Тишко В. А. Інформаційна технологія аналітики комп'ютерної гри Dota 2. Магістерська кваліфікаційна робота зі спеціальності 122 «Комп'ютерні науки», освітня програма «Системи штучного інтелекту». Вінниця: ВНТУ, 2023. 96 с.

Укр. мовою. Бібліогр.: 26 назв; рис.: 22; табл.: 12.

Робота присвячена розробці інформаційної технології аналітики комп'ютерної гри Dota 2. Проведено аналіз сучасних методів та засобів аналітики комп'ютерної гри Dota 2, виявлено їх переваги та недоліки в результаті чого було знайдено проблему у необхідності створення інформаційної технології, яка б виконувала аналітику комп'ютерної гри Dota 2 з розширеним функціоналом. В результаті чого було виконано постановку задачі.

Розроблено алгоритм аналітики комп'ютерної гри Dota 2, сформовано структурну схему інформаційної технології. Після чого було проаналізовано та обрано мову програмування та середовище розробки для реалізації інформаційної технології та описано її реалізацію.

Після цього було протестовано інформаційну технологію відповідно до вхідних даних та констатовано розширення функціоналу, а саме вибір комп'ютерного персонажа та спорядження за рахунок статистичного аналізу.

Графічна частина складається з 7 плакатів із результатами моделювання.

В розділі економічної частини проведено оцінювання комерційного потенціалу розробки інформаційної технології аналізу комп'ютерної гри Dota 2, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, які склали 640651 грн, розраховано період окупності – 0,57 року.

Ключові слова: ігрова аналітика, статистичний аналіз, API, телеметрія, стратегія гри, електронний спорт, Dota 2

ABSTRACT

Tyshko V. A. Information technology of analytics of the computer game Dota 2. Master's thesis in the specialty 122 “Computer Sciences”, educational program “Artificial Intelligence Systems”. Vinnytsia: VNTU, 2023. 96 p.

In English. Bibliogr.: 26 titles; pics.: 22; Tables: 12.

The master's thesis is devoted to the development of information technology for analyzing the computer game Dota 2. An analysis of modern methods and tools for analyzing the computer game Dota 2 was carried out, their advantages and disadvantages were identified, as a result of which the problem of the need to create an information technology that would perform analytics of the computer game Dota 2 with advanced functionality was found. As a result, the problem was solved.

An algorithm for analyzing the Dota 2 computer game was developed, and a structural diagram of information technology was formed. After that, the programming language and development environment for the implementation of information technology were analyzed and selected, and its implementation was described.

After that, the information technology was tested in accordance with the input data and the extension of the functionality, namely the choice of a computer character and equipment through statistical analysis, was stated.

The graphic part consists of 7 posters with the results of modeling.

In the economic section, the commercial potential of the development of information technology for analyzing the computer game Dota 2 was assessed, the costs of research and implementation of the results were predicted, which amounted to 640651 UAH, and the payback period was calculated to be 0.57 years.

Keywords: game analytics, statistical analysis, API, telemetry, game strategy, e-sports, Dota 2

ЗМІСТ

ВСТУП.....	4
1 СУЧАСНИЙ РІВЕНЬ РОЗВИТКУ АНАЛІТИКИ КОМП'ЮТЕРНОЇ ГРИ DOTA 2.....	7
1.1 Аналіз сучасних методів аналітики комп'ютерної гри Dota 2	7
1.2 Аналіз сучасних засобів аналітики комп'ютерної гри Dota 2	9
1.3 Постановка задачі	19
1.4 Висновок до розділу 1	20
2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ, АЛГОРИТМУ ТА СТРУКТУРИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІТИКИ КОМП'ЮТЕРНОЇ ГРИ DOTA 2.....	21
2.1 Удосконалена математична модель вибору комп'ютерного персонажа	21
2.2 Удосконалений алгоритм аналітики комп'ютерної гри Dota 2.....	22
2.3 Розробка структури інформаційної технології аналітики комп'ютерної гри Dota 2	26
2.4 Висновок до розділу 2	27
3 РОЗРОБКА МОДУЛІВ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІТИКИ КОМП'ЮТЕРНОЇ ГРИ DOTA 2	28
3.1 Обґрунтування вибору мов програмування	28
3.2 Вибір середовища розробки.....	40
3.3 Розробка інтерфейсу інформаційної технології аналітики комп'ютерної гри Dota 2	44
3.4 Програмна реалізація модуля збору даних	48
3.5 Програмна реалізація модуля обробки даних	50
3.6 Програмна реалізація модуля статистичного аналізу	53
3.7 Програмна реалізація модуля виведення даних.....	55
3.8 Тестування інформаційної технології аналітики комп'ютерної гри Dota 2.....	57
3.9 Висновок до розділу 3	61

4 ЕКОНОМІЧНА ЧАСТИНА	62
4.1 Проведення комерційного та технологічного аудиту інформаційної технології аналітики комп'ютерної гри Dota 2.....	62
4.2 Розрахунок витрат на здійснення науково-дослідної роботи	63
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	71
4.4 Висновок до розділу 4	75
ВИСНОВКИ	76
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	78
ДОДАТКИ	80
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	81
Додаток Б (обов'язковий) Лістинг програми.....	81
Додаток В (обов'язковий) Ілюстративна частина	86
Додаток Г (довідниковий) Інструкція користувача.....	94

ВСТУП

Актуальність теми дослідження. Аналітика комп'ютерних ігор, зокрема Dota 2, має велику популярність для гравців, команд, тренерів та фанатів цієї популярної МОБА-гри. Тому, доцільною є розробка аналітики для самої гри. Вона надає можливість розуміти як влаштована гра, ефективно використовувати героїв та стратегії, а також покращувати особисті геймплейні навички. Гравці можуть аналізувати свої матчі, виявляти помилки, вдосконалювати тактику і прийоми гри, що допомагає досягати вищих результатів.

Сучасний рівень розвитку інформаційних технологій забезпечує можливість для реалізації аналітики комп'ютерної гри Dota 2. Завдяки постійному зростанню обчислювальної потужності комп'ютерів та доступності великих обсягів даних, аналітичні інструменти можуть швидко обробляти інформацію про гру. Статистичний аналіз дозволяє розпізнавати шаблони, прогнозувати ходи гравців та розвиток гри, що дає можливість командам та тренерам зробити більш обґрунтовані стратегічні рішення.

Таким чином, аналітика комп'ютерної гри Dota 2 є надзвичайно актуальною і корисною для всіх зацікавлених сторін. Вона сприяє розвитку геймінгу та підвищенню професійного рівня гравців і команд. Завдяки сучасним інформаційним технологіям, ми маємо можливість аналізувати та прогнозувати гру з високою точністю. Аналітика допомагає виявляти слабкі місця, вдосконалювати стратегії та прийоми гри. З урахуванням постійного розвитку геймінгу, роль аналітики в Dota 2 стає ще більш суттєвою і значущою для досягнення успіху у світі цієї гри.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 "Розробка прикладних інтелектуальних інформаційних

технологій та систем" та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження є розширення функціоналу сучасних засобів аналітики комп'ютерної гри Dota 2.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз існуючих методів аналітики комп'ютерної гри Dota 2;
- провести аналіз існуючих засобів аналітики комп'ютерної гри Dota 2;
- розробити удосконалену математичну модель вибору комп'ютерного персонажа комп'ютерної гри Dota 2;
- розробити удосконалений алгоритм аналітики комп'ютерної гри Dota 2;
- розробити структуру інформаційної технології аналітики комп'ютерної гри Dota 2;
- розробити інформаційну технології аналітики комп'ютерної гри Dota 2.

Об'єкт дослідження – це процес аналітики комп'ютерної гри Dota 2.

Предмет дослідження – це засоби аналітики комп'ютерної гри Dota 2.

Методи дослідження. У роботі використано такі методи наукових досліджень: методи системного аналізу, метод статистичного аналізу для аналітики комп'ютерної гри Dota 2.

Наукова новизна одержаних результатів:

1. Удосконалено математичну модель аналітики комп'ютерної гри Dota 2, яка на відміну від існуючих відрізняється введенням додаткових критеріїв аналітики, що дозволило розширити функціонал інформаційної технології.

2. Розроблено веб-орієнтовану інформаційну технологію аналітики комп'ютерної гри Dota 2, яка відрізняється від існуючих розширеним функціоналом, що забезпечує доцільність обрання певного персонажа та його спорядження.

Практичне значення одержаних результатів:

- розробка удосконаленого алгоритму аналітики комп'ютерної гри Dota 2;

– розробка структури інформаційної технології аналітики комп'ютерної гри Dota 2;

– програмна реалізація модулів збору даних, обробки даних, статистичного аналізу та виведення даних.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно.

Апробація

Основні результати роботи апробовано на Всеукраїнській науково-практичній інтернет-конференції «LII Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації», ВНТУ Вінниця, 2023 [1] та XXVIII Міжнародній науково-практичній конференції «Science and development of methods for solving modern problems», Мельбурн, Австралія, 2023 [2].

Публікації

За результатами роботи опубліковано дві тези доповіді «Савчук Т. О., Тишко В. А. Розробка удосконаленого алгоритму аналітики комп'ютерної гри Dota 2» [1] та «Савчук Т. О., Тишко В. А. Структура інформаційної технології аналітики комп'ютерної гри Dota 2» [2], отримано свідоцтво про реєстрацію авторського право на комп'ютерну програму «Інформаційна технологія аналітики комп'ютерної гри Dota 2» - (АП) с202308054 Вх-47007/2023 від 28.11 [3]. Крім того, було підготовлено статтю та направлено її до фахового журналу категорії «Б» «Наука і техніка сьогодні».

1 СУЧАСНИЙ РІВЕНЬ РОЗВИТКУ АНАЛІТИКИ КОМП'ЮТЕРНОЇ ГРИ DOTA 2

1.1 Аналіз сучасних методів аналітики комп'ютерної гри Dota 2

Серед сучасних методів аналітики [4] заслуговують уваги такі:

– Машинне навчання та штучний інтелект [5]: використання методів машинного навчання та штучного інтелекту дозволяє аналізувати великі обсяги даних з матчів Dota 2 [6] та знаходити складні зв'язки та шаблони. Переваги цього методу включають здатність до автоматичного виявлення нових стратегій та розуміння тенденцій у грі. Однак, мінусом є необхідність великого обсягу даних для навчання моделей та складність їх інтерпретації.

– Статистичний аналіз даних матчів [7]: статистичний аналіз та використання даних з матчів Dota 2 надають цінну інформацію про гру. Цей метод дозволяє вивчати ефективність героїв, стратегії та предмети, а також робити прогнози результатів матчів. Переваги полягають у конкретних числових даних, які можна використовувати для прийняття рішень. Проте, недоліком є можливість впливу факторів, таких як патчі на баланс гри та результати аналізу.

– Візуалізація даних: використання графічного представлення даних дозволяє зрозуміти залежності та шаблони у грі Dota 2. Візуалізація може надати інтуїтивний спосіб аналізувати дані, такі як рух гравців на карті, розподіл героїв та ефективність стратегій. Це сприяє легшому розумінню та виявленню закономірностей. Однак, використання візуалізації може бути обмеженим в аналізі більш складних аспектів гри.

– Аналіз стрімів та коментарів: стрімери та коментатори Dota 2 можуть надати важливу інформацію про гру та стратегії. Аналіз коментарів та обговорень допомагає розуміти підходи гравців, виявляти нові стратегії та отримувати враження про настрій гравців та спільноти. Переваги цього методу включають доступність реальних часових даних та експертну інформацію.

Проте, недоліком може бути суб'єктивність та обмежена універсальність аналізу.

– Соціальний аналіз [8]: вивчення соціальних мереж, форумів та спільноти гравців може надати інформацію про ставлення до гри, балансу героїв та оновлень гри. Це може бути важливим джерелом відгуків та сприйняття гравцями змін у грі. Однак, соціальний аналіз може бути суб'єктивним та необ'єктивним, залежно від спільноти гравців, яку ви досліджуєте.

У таблиці 1.1 наведено порівняльну характеристику сучасних методів аналітики комп'ютерної гри Dota 2:

Таблиця 1.1 – Порівняльна характеристика сучасних методів аналітики комп'ютерної гри Dota 2.

Засіб аналітики	Переваги	Недоліки
Машинне навчання та штучний інтелект	- Здатність до автоматичного виявлення нових стратегій та розуміння тенденцій у грі	- Необхідність великого обсягу даних для навчання моделей та складність їх інтерпретації
Статистичний аналіз даних матчів	- Конкретні числові дані, які можна використовувати для прийняття рішень	- Вплив факторів, таких як патчі на баланс гри та результати аналізу
Візуалізація даних	- Інтуїтивний спосіб аналізувати дані	- Візуалізація може бути обмеженою в аналізі більш складних аспектів гри
Аналіз стрімів та коментарів	- Доступність реальних часових даних та експертна інформація	- Суб'єктивність та обмежена універсальність аналізу

Продовження таблиці 1.1

Соціальний аналіз	- Може бути важливим джерелом відгуків та сприйняття гравцями змін у грі	- Може бути суб'єктивним та необ'єктивним, залежно від спільноти гравців, яку ви досліджуєте
----------------------	--	--

Отже, враховуючи всі описані методи, варто вибрати статистичний аналіз. Використання цього методу дозволяє отримати об'єктивні числові дані, які можна використовувати для прийняття рішень у грі. Через аналіз даних з матчів, можна вивчити ефективність героїв, стратегії та предмети, а також робити прогнози результатів матчів. Плюсами використання статистичного аналізу є об'єктивність даних, можливість здійснювати кількісні порівняння та приймати обґрунтовані рішення. Зважаючи на це, статистичний аналіз є цінним інструментом для аналізу комп'ютерної гри Dota 2 і може сприяти поліпшенню результатів гравців.

1.2 Аналіз сучасних засобів аналітики комп'ютерної гри Dota 2

Серед сучасних засобів аналітики комп'ютерної гри Dota 2 заслуговують уваги такі:

– OpenDota [9] – це ще один сервіс, спрямований на аналіз статистики гравців у грі Dota 2. Він надає користувачам широкий спектр інструментів та функцій для вивчення своєї гри та аналізу даних.

Переваги OpenDota:

1. Глибока аналітика. OpenDota надає велику кількість аналітичних засобів, включаючи графіки, таблиці та графіки, щоб детально розглядати різні аспекти гри, такі як використання героїв, ефективність гравця та інше.

2. Порівняння гравців. Користувачі можуть порівнювати свої досягнення з іншими гравцями, що дозволяє визначити сильні та слабкі сторони своєї гри.
3. Історія матчів. OpenDota веде детальну історію всіх матчів гравця, дозволяючи вивчати свої стратегії та розвиватися на їхній основі.
4. Безкоштовний доступ до даних. Багато базових функцій OpenDota доступні безкоштовно, що робить його доступним для широкого кола гравців.

Недоліки OpenDota:

1. Складність інтерфейсу. Інтерфейс OpenDota може здатися складним для новачків або тих, хто не знайомий з аналітичними інструментами.
2. Залежність від API і оновлень гри. Подібно до Dotabuff, OpenDota використовує API від розробника гри, тому може виникнути ситуація, коли оновлення гри або зміни у API впливають на доступність деяких функцій.
3. Не повний доступ до даних про приватні матчі. Деякі дані, особливо щодо приватних матчів, можуть бути недоступними або обмеженими через обмеження API.
4. Можливість втрати конфіденційності. Також, як і в інших аналітичних сервісах, деякі гравці можуть бути обурені можливістю публічного відображення своєї статистики та даних.

В цілому, OpenDota є потужним інструментом для аналізу гри в Dota 2, зокрема для тих, хто зацікавлений у глибокому розумінні стратегій та покращенні своїх ігрових навичок.

Дане забезпечення зображено на рисунку 1.1.

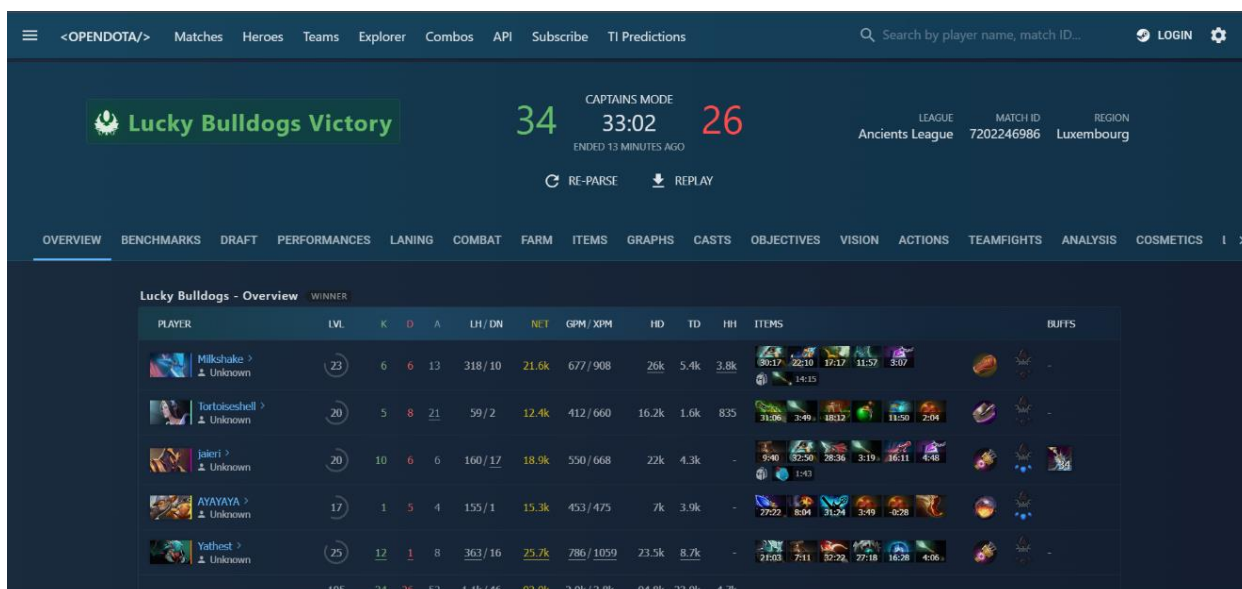


Рисунок 1.1 – Програмне забезпечення «OpenDota»

– Dota Plus [10] – це платний підписний сервіс в грі Dota 2, який пропонує ряд додаткових функцій та вигод для гравців. Ось кілька переваг та можливостей Dota Plus:

Переваги Dota Plus:

1. Hero Progression. Гравці можуть відстежувати свій прогрес з кожним героєм, отримуючи винагороди за досягнуті цілі та виклики.
2. Hero Relics. Dota Plus дозволяє гравцям відображати свої досягнення з конкретним героєм через реліки, що може бути видно іншим гравцям у грі.
3. Assistant. Надає інформацію та рекомендації щодо вибору героїв, предметів та навичок, а також надає аналітику під час гри.
4. Бойові пропозиції та виклики. Гравці отримують доступ до бойових пропозицій та викликів, що можуть додатково стимулювати їхню гру та пропонувати винагороди.
5. Косметичні предмети. Підписники отримують доступ до ексклюзивних косметичних предметів та стилів для героїв.

6. Сучасні та допоміжні статистичні дані. Dota Plus надає розширені статистичні дані, включаючи графіки використання героїв, предметів та інше.

Недоліки Dota Plus:

1. Платний сервіс. Dota Plus є платним, що може бути неприйнятним для гравців, які не бажають витратити додаткові кошти на гру.
2. Залежність від періодичних оновлень. Як і будь-який сервіс, Dota Plus може зазнавати змін та оновлень, які можуть впливати на його функціональність та привабливість.
3. Не для всіх гравців. Деякі гравці можуть вважати, що функції Dota Plus не є обов'язковими або важливими для їхньої гри, тому вони можуть не бачити в ньому значущої вартості.
4. Можливий дисбаланс. Деякі гравці можуть вважати, що функції Assistant, надані Dota Plus, можуть призводити до дисбалансу або неправильних рекомендацій у грі.

Дота Плюс створений для тих, хто шукає додаткові можливості та зручності у грі Dota 2, і його прийняття залежить від особистих вподобань і відношення до платних сервісів.

Програмне забезпечення зображено на рисунку 1.2.

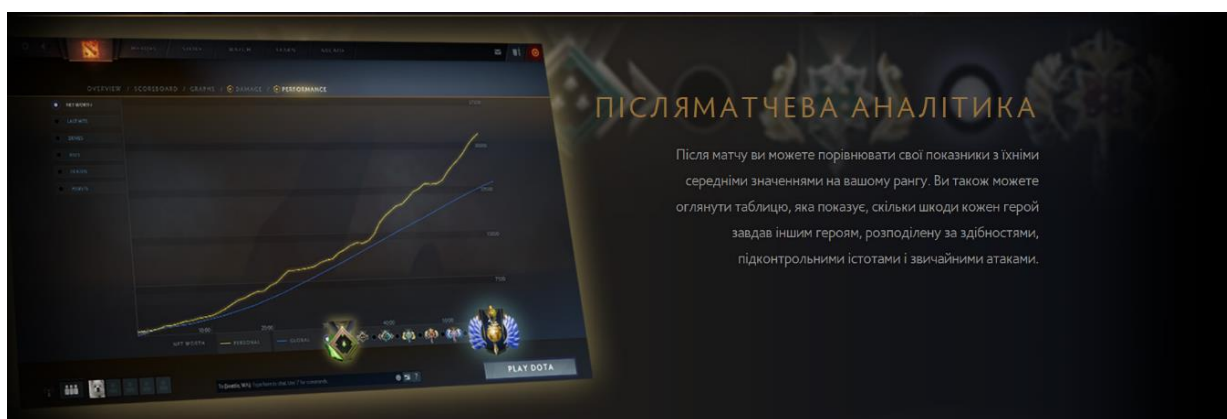


Рисунок 1.2 – Програмне забезпечення «Dota Plus»

– Strafe [11] – це мобільний додаток, призначений для відстеження інформації про електронні спортивні події та результати матчів у світі кіберспорту.

Переваги Strafe:

1. Відстеження інформації. Додаток дозволяє користувачам отримувати свіжу інформацію про різні кіберспортивні змагання, команди та гравці.
2. Результати та статистика. Strafe надає результати матчів, статистику гравців, інформацію про склади команд та інші важливі дані.
3. Наступні матчі та розклади. Користувачі можуть переглядати розклади та дізнаватися про наступні матчі у світі кіберспорту.
4. Повідомлення про події. Ви отримуєте повідомлення про важливі події, оновлення та результати на основі ваших інтересів.
5. Спільнота і коментарі. Strafe може включати функції спільноти, де користувачі можуть обговорювати матчі та події, ділитися своїми враженнями.

Недоліки Strafe:

1. Обмежена географічна покриття. Деякі події можуть мати обмежене покриття в залежності від регіону та доступності даних.
2. Залежність від актуалізацій. Точність та повнота інформації може залежати від регулярних оновлень та підтримки додатка.
3. Необхідність інтернет-з'єднання. Для коректної роботи додатка потрібне стабільне інтернет-з'єднання.
4. Можливі обмеження для деяких ігор. Деякі менш популярні чи екзотичні ігри можуть бути менше представленими або вимагати додаткових даних для відстеження.

Загалом, Strafe може бути цікавим інструментом для шанувальників кіберспорту, але, як і з будь-яким додатком, важливо розуміти його обмеження та переваги залежно від ваших потреб та очікувань.

Програмне забезпечення «Strafe» представлено на рисунку 1.3.

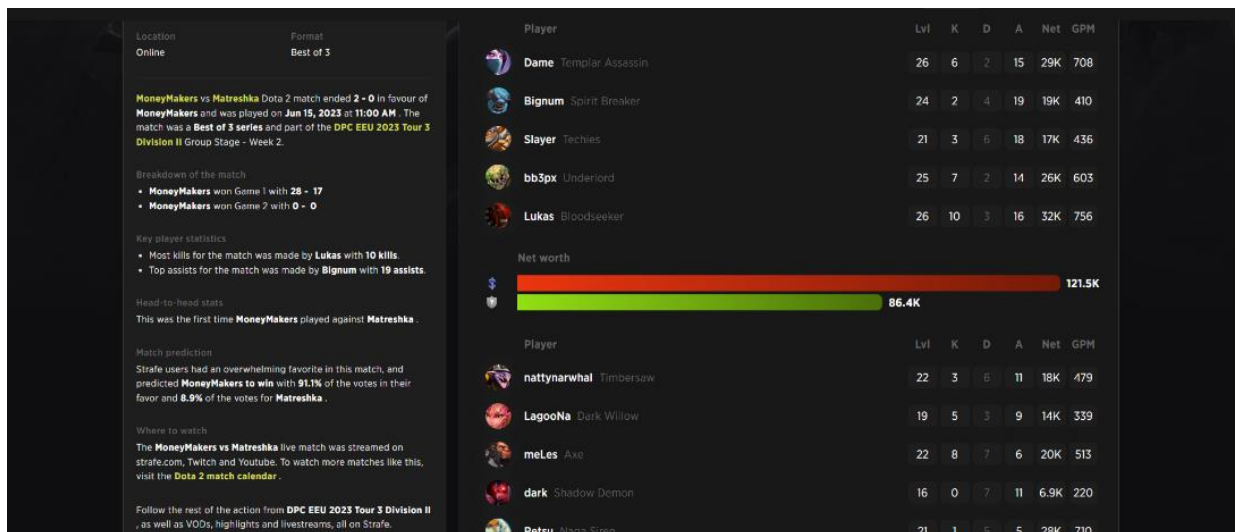


Рисунок 1.3 – Програмне забезпечення «Strafe»

– Dotabuff [12] – це онлайн-платформа, призначена для аналізу і відстеження статистики гравців у відомій грі Dota 2. Сервіс збирає дані про гравців, їхні героїв, матчі та інші параметри гри, надаючи користувачам можливість детально вивчати свої досягнення, а також порівнювати їх із іншими гравцями.

Переваги Dotabuff:

1. Статистика та аналіз. Гравці можуть переглядати різноманітну статистику щодо своєї гри, включаючи відсоток перемог, ефективність використання героїв, аналіз матчів та інше.
2. Профіль гравця. Кожен гравець має власний профіль з детальною інформацією про його досягнення, героїв та стилі гри.
3. Матчі та рейтинги. Сервіс відстежує всі матчі гравця, а також надає рейтингові системи для класифікації гравців за їхнім рівнем вмінь.
4. Докладні статистичні дані. Dotabuff надає докладні статистичні дані про героїв, предмети, карту гри і багато іншого, що допомагає гравцям розвиватися та вдосконалювати свою гру.

Недоліки Dotabuff:

1. Приватність даних. Деякі гравці можуть бути обурені тим, що їхня статистика є публічною, і сервіс може відображати їхні слабкі сторони та помилки.
2. Залежність від API. Dotabuff використовує API від розробника гри, тому може виникнути ситуація, коли зміни у грі або недоступність API можуть вплинути на функціональність сервісу.
3. Платна версія. Деякі розширені функції та можливості доступні лише за плату, що може не влаштовувати деяких користувачів.
4. Обмеженість на рівні даних. Оскільки Dotabuff залежить від даних, наданих розробниками гри, іноді може виникати обмеженість у зборі та відображенні деякої інформації.

Загалом, Dotabuff є важливим інструментом для гравців Dota 2, допомагаючи їм вивчати свою гру та розвиватися в процесі гри.

Програмне забезпечення представлено на рисунку 1.4.



Рисунок 1.4 – Програмне забезпечення «Dotabuff»

– Overwolf [13] – це платформа для розробки та використання ігрових додатків (overlay apps) для різних комп'ютерних ігор. Основна ідея полягає в тому, щоб дозволити розробникам створювати розширення та додатки, які можна використовувати під час гри, накладаючись на екран чи взаємодіючи з грою. Давайте розглянемо переваги та недоліки Overwolf:

Переваги Overwolf:

1. Розширення та додатки для ігор. Overwolf дозволяє розробникам створювати різні ігрові додатки, які можуть полегшити гравцям життя, забезпечуючи додаткові функції під час гри.
2. Спільнота та ресурси для розробників. Є спільнота розробників, яка підтримується Overwolf, і надає різноманітні ресурси, документацію та інструменти для того, щоб розробка була ефективною.
3. Інтерактивність з грою. Overwolf дозволяє створювати додатки, які можуть взаємодіяти з грою, накладаючись на екран або надаючи корисну інформацію під час гри.
4. Стрімінг та соціальна інтеграція. Overwolf має функції, які полегшують стрімінг гри та взаємодію зі спільнотою.
5. Підтримка для різних ігор. Overwolf підтримує широкий спектр ігор, що дозволяє використовувати додатки для багатьох відомих ігор.

Недоліки Overwolf:

1. Використання ресурсів. Overwolf може використовувати додаткові ресурси комп'ютера, що може призвести до збільшення навантаження на систему, особливо при використанні багатьох додатків одночасно.
2. Залежність від розробників. Деякі ігри можуть не підтримувати Overwolf, або розробники можуть змінити умови використання, що може вплинути на функціональність додатків.
3. Безпека та довіра. Для деяких користувачів може виникнути питання щодо безпеки та приватності при використанні сторонніх додатків, зокрема, якщо вони взаємодіють з ігровими процесами.

4. Можливі конфлікти з антивірусами чи античит-системами. Оскільки Overwolf взаємодіє з процесами гри, деякі антивірусні програми чи античит-системи можуть виявляти його як потенційну загрозу.

Overwolf є цікавим інструментом для гравців, які шукають додатковий функціонал та інтерактивність під час гри. Важливо зазначити, що функціональність та підтримка можуть змінюватися в залежності від ігор та розробників.

Програмне забезпечення зображено на рисунку 1.5.

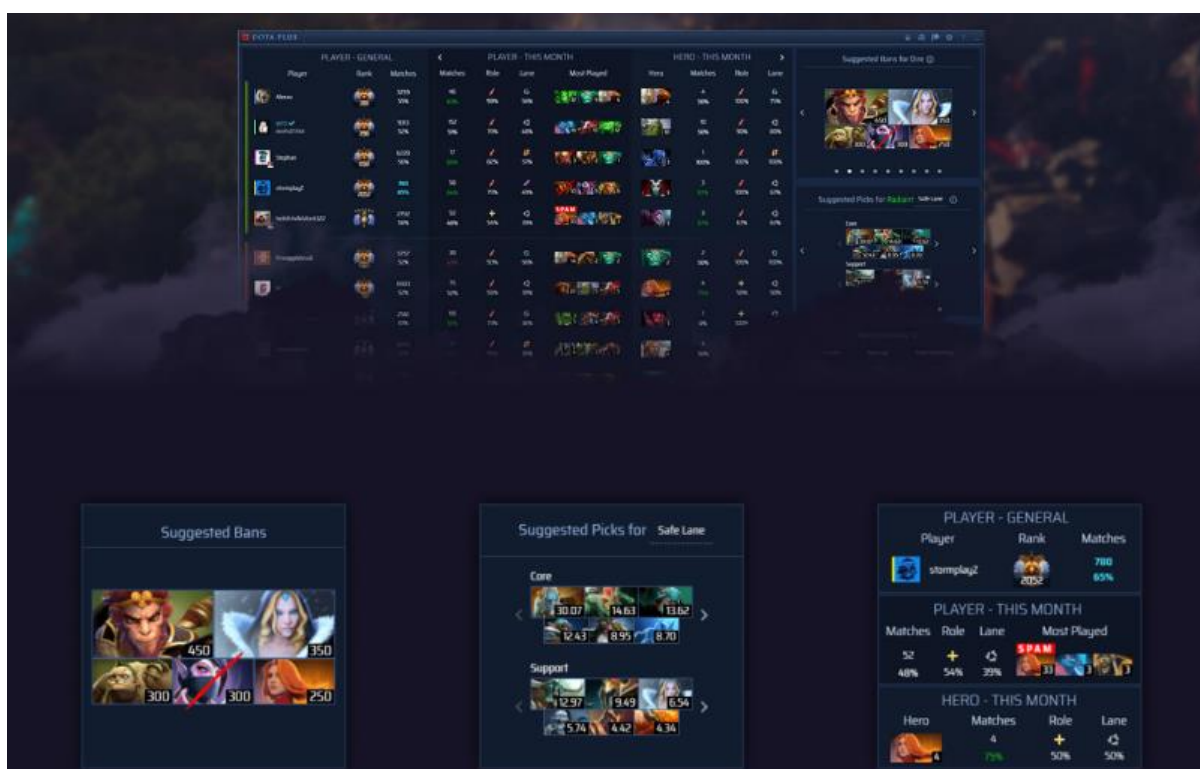


Рисунок 1.5 – Програмне забезпечення «Overwolf»

У таблиці 1.2 наведено порівняльну характеристику основних конкурентів розроблюваної інформаційної технології.

Таблиця 1.2 – Порівняльна характеристика основних програмних засобів аналітики комп'ютерної гри Dota 2.

Засіб аналітики	Переваги	Недоліки
OpenDota	- Великий обсяг доступної статистики	- Складність використання для новачків
Dota Plus	- Можливість аналізувати власні помилки та вдосконалювати гру - Детальна статистика гри - Рекомендації з вибору героїв та тренувальні програми	- Платна підписка
Strafe	- Візуальні ефекти та косметичні предмети - Слідкування за живими результатами матчів	- Відсутність детальних аналітичних функцій - Обмежені можливості порівняння даних
Dotabuff	- Аналітичні дані про гравця - Детальна статистика та аналіз власних матчів - Глобальні та локальні рейтинги	- Платна підписка - Функції можуть бути доступні лише на платформі
Overwolf	- Соціальні функції для спілкування та обміну досвідом - Зручний доступ до аналітики прямо в грі Dota - Різноманітні додатки та інструменти	- Вимагає додаткових ресурсів системи - Можливий вплив на продуктивність гри

На основі проведеної порівняльної характеристики існуючих засобів аналітики комп'ютерної гри Dota 2 можна зробити висновок, що є потреба в розробці інформаційної технології, яка міститиме усі необхідні функції для виконання поставлених завдань і компенсує зазначені недоліки функціоналу

існуючих засобів, а саме – обмежені можливості порівняння даних, відсутність деяких аналітичних функцій, платна підписка. Серед описаних інструментів для аналітики комп'ютерної гри Dota 2 найбільш повний функціонал надає платформа Dotabuff, тому вона обрана у якості аналога для розробки інформаційної технології.

1.3 Постановка задачі

Для розробки ефективної системи аналізу, яка буде враховувати ключові параметри гри, введемо такі позначення:

$X (x_1, x_2, x_3, x_4)$ – це множина параметрів гри, що аналізуються.

x_1 – частота обирання персонажа,

x_2 – частота обирання спорядження,

x_3 – частота виграшу персонажа,

x_4 – частота виграшу спорядження, з яким виграв персонаж

$Y (y_1, y_2, y_3, y_4)$ – вектор вагових коефіцієнтів параметрів гри, що приймають значення від 0 до 1, $\sum_1^4 Y_i = 1$.

y_1 – ваговий коефіцієнт частоти обирання персонажа,

y_2 – ваговий коефіцієнт частоти обирання спорядження,

y_3 – ваговий коефіцієнт частоти виграшу персонажа,

y_4 – ваговий коефіцієнт частоти виграшу спорядження, з яким виграв персонаж

Задача аналітики комп'ютерної гри Dota 2 полягає у визначенні значення

$$F = Y * X, \quad (1.1)$$

де F – це результат аналітики щодо обраного персонажа та його спорядження.

Таким чином, цей метод аналітики комп'ютерної гри Dota 2 за рахунок використання статистичного аналізу допоможе гравцям робити вибір в обранні певного персонажа чи спорядження.

1.4 Висновок до розділу 1

У даному розділі було досліджено та проаналізовано перелік причин, що доводять актуальність розробки інформаційної технології аналітики комп'ютерної гри Dota 2.

Був проведений огляд сучасних методів та засобів для вирішення задачі аналітики комп'ютерної гри Dota 2, визначено особливості даних засобів, які можуть бути корисними та їх недоліки, складено порівняльну характеристику.

В процесі огляду існуючих засобів аналітики комп'ютерної гри Dota 2 було з'ясовано, що багато з них мають недоліки, такі як обмежені можливості порівняння даних, відсутність деяких аналітичних функцій, платна підписка. На основі цього аналізу було сформульовано задачу дослідження – розробити інформаційну технологію, яка б дозволила розширити функціонал аналітики комп'ютерної гри Dota 2 шляхом створення системи, яка використовує статистичний аналіз для більш точного результату.

Сформульовані в цьому розділі висновки та задачі дослідження лягли в основу подальшого дослідження та розробки.

2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ, АЛГОРИТМУ ТА СТРУКТУРИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІТИКИ КОМП'ЮТЕРНОЇ ГРИ DOTA 2

2.1 Удосконалена математична модель вибору комп'ютерного персонажа

Класична математична модель вибору комп'ютерного персонажа може використовувати різні критерії для прийняття рішення. Одним із поширених підходів є використання функцій корисності, які оцінюють рівень задоволення гравця від кожного персонажа.

Нехай U_i представляє функцію корисності для персонажа i , де i може бути індексом конкретного персонажа. Ця функція може включати різні параметри, такі як сила, швидкість, здоров'я, навички і так далі. Функція вибору персонажа F_j , яка максимізує його загальну корисність може бути поданою як:

$$F_j = \arg \max_i U_i, \quad (2.1)$$

де $\arg \max_i$ визначає індекс персонажа, для якого функція корисності максимальна.

Використання статистичного аналізу при визначенні значення критеріїв дозволить більш точно врахувати складні взаємодії між різними факторами та забезпечити оптимальний вибір комп'ютерного персонажа.

Для удосконалення математичної моделі вибору комп'ютерного персонажа в грі Dota 2 сформуємо функцію M_{ij} вибору i -го персонажа і його j -го спорядження.

Тоді:

x_1 - частота обирання персонажа,

x_2 - частота обирання спорядження,

x_3 - частота виграшу персонажа,

x_4 - частота виграшу спорядження, з яким виграв персонаж
 y_1 – ваговий коефіцієнт частоти обирання персонажа,
 z_1 - ваговий коефіцієнт частоти обирання спорядження,
 y_2 - ваговий коефіцієнт частоти виграшу персонажа,
 z_2 - ваговий коефіцієнт частоти виграшу спорядження, з яким виграв персонаж

$$M_{ij} = \sum_{i=1}^4 (y_i \times C(x_i) + z_j \times G(x_j)), \quad (2.2)$$

де $C(x_i)$ – представляє оцінку i -го критерію для персонажа C , а $G(x_j)$ – оцінку j -го критерію для спорядження G .

Отже, введення правил при виборі персонажа забезпечить додаткові функціональні можливості при виборі персонажа та спорядження персонажа, що дозволить частіше вигравати у грі.

2.2 Удосконалений алгоритм аналітики комп'ютерної гри Dota 2

Розглянемо узагальнений алгоритм аналітики комп'ютерної гри Dota 2. Він полягає в особистому виборі персонажа гравцями на підставі їх власних уподобань і стратегії гри. Зазвичай цей процес включає в себе наступні етапи:

1. Визначення ролі в команді. гравці спочатку вирішують, яку роль вони будуть грати в команді. Це може бути роль "в carry" (гравець, який отримує багато фарму та наносить великий шкоду), "в support" (гравець, який надає підтримку та забезпечує контроль над картою), "в mid" (гравець, який грає на середній лінії і намагається домінувати у цій області) і т. д.

2. Вибір персонажа. Після визначення ролі гравці вибирають персонажів, які найкраще відповідають цій ролі та їхнім власним навичкам і вподобанням.

На рисунку 2.1 зображено UML-діаграму роботи зазначеного алгоритму.



Рисунок 2.1 – UML-діаграма узагальненого алгоритму аналітики комп'ютерної гри Dota 2

Створення удосконаленого алгоритму для аналітики комп'ютерної гри Dota 2 є доцільним та актуальним з кількох причин. По-перше, гра має велику кількість даних, які можна проаналізувати. Збір і обробка цих даних може дати корисну інформацію для гравців, тренерів та команд для поліпшення їхньої гри. По-друге, удосконалений алгоритм для аналітики Dota 2 може допомогти виявити та розкрити нові стратегії та тактики гри. За допомогою аналізу даних, можна виявити успішні шаблони гри, виявити слабкі місця противників та зрозуміти, які характеристики гри впливають на результат. Це може сприяти вдосконаленню гравців та команд, дозволяючи їм розвиватися та просуватися в рейтингових турнірах. Крім того, удосконалений алгоритм для аналітики Dota 2 може бути корисним для глядачів та експертів гри. Зі зростанням популярності електронного спорту, багато людей зацікавлені в спостереженні та аналізі професійних матчів. Завдяки інформаційній технології, глядачі можуть отримати статистику, графіки, гравців та команди, що допоможе їм краще розуміти гру та насолоджуватися переглядом матчів.

Отже, створення удосконаленого алгоритму для аналітики комп'ютерної гри Dota 2 є доцільним і актуальним, оскільки це допоможе гравцям покращити

свої навички за рахунок інформації, що допоможе змінити стиль гри або уникнути неочевидних помилок з вибором персонажа.

При реалізації удосконаленого алгоритму для аналітики комп'ютерної гри Dota 2 доцільно використати статистичний аналіз, як такий, що дозволяє знизити суб'єктивність впливу людського фактору на результати аналізу. Джерелом даних буде офіційний сайт Dota 2, який надає доступ до історичних даних про матчі, гравців, команди, героїв тощо. Очікується, що результати цього дослідження будуть корисними для гравців, розробників та дослідників у сфері ігрової аналітики.

Алгоритм складатиметься з таких кроків:

Крок 1. Збір даних:

– Знайомство з основними джерелами даних, які можуть бути використані для аналізу Dota 2 (веб-сайти, API, бази даних тощо).

– Завантаження або отримання даних про матчі, гравців, героїв, рейтинги, склади команд, тощо.

– Збереження зібраних даних у форматі, який зручний для подальшого аналізу.

Крок 2. Попередній аналіз даних:

– Обробка зібраних даних (дані про матчі, гравців, героїв, склади команд, тощо).

– Виявлення та видалення аномальних або неповних записів.

– Виявлення потенційних проблем чи викликів для подальшого аналізу.

Крок 3. Визначення цілей аналізу:

– Встановлення цілей та обмежень аналізу.

– Встановлення конкретних метрик або показників, які будуть використовуватися для оцінки та вимірювання.

Крок 4. Реалізація та виконання аналізу:

– Реалізація програми аналітики, включаючи необхідні функції та алгоритми.

– Застосування аналізу до зібраних даних.

– Отримання результатів аналізу та їх інтерпретація.

Крок 5. Валідація та оцінка результатів:

- Перевірка достовірності та точності результатів аналізу.
- Оцінка ефективності використаних методів та моделей.
- Порівняння результатів зі встановленими цілями та метриками.

Крок 6. Висновки та подання результатів:

- Зведення результатів аналізу.
- Формулювання висновків, які відповідають на поставлені питання.

На рисунку 2.2 зображено UML-діаграму роботи зазначеного алгоритму.

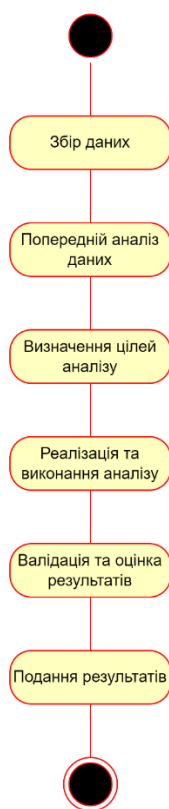


Рисунок 2.2 – UML-діаграма удосконаленого алгоритму аналітики комп'ютерної гри Dota 2

Отже, запропонований удосконалений алгоритм аналітики комп'ютерної гри Dota 2, дозволить розширити функціонал засобу, а саме підбір персонажа та спорядження за рахунок використання статистичного аналізу.

2.3 Розробка структури інформаційної технології аналітики комп'ютерної гри Dota 2

Розробка структури інформаційної технології аналітики комп'ютерної гри Dota 2 є критичним етапом, спрямованим на створення ефективної та добре організованої системи, яка сприятиме успішній аналітиці гри. Правильно розроблена структура забезпечує:

1. Оптимізацію процесу розробки. Використання ретельно розробленої структури дозволяє зменшити час розробки та впровадження нового програмного продукту.
2. Спрощення процесу підтримки та доповнення продукту новими функціями.
3. Створення безпечного та надійного програмного продукту.
4. Легку масштабованість продукту для роботи з більшим обсягом даних чи користувачів.
5. Високу продуктивність та швидкість роботи, навіть при збільшенні навантаження.
6. Високий ступінь покриття тестами для виявлення та виправлення помилок.

Запропонований алгоритм аналітики комп'ютерної гри Dota 2 можна реалізувати за допомогою відповідного програмного забезпечення, що включає модулі:

1. Модуль збору даних.
2. Модуль обробки даних.
3. Модуль статистичного аналізу.
4. Модуль виведення даних.

Модуль збору даних отримує інформацію про матчі Dota 2, використовуючи API. Далі ці дані передаються в модуль обробки даних, який очищає їх від непотрібної інформації, виконує перетворення та структурування. Наприклад, він може видаляти повторювані записи, корегувати помилки або перетворювати дані в певному форматі. Після обробки дані передаються до

модуля статистичного аналізу, де вони піддаються використанню статистичних методів та алгоритмів. Далі результати аналізу передаються в модуль виведення даних, що представляє їх в зручному для користувача вигляді.

Структуру інформаційної технології та взаємодію її основних модулів зображено на рисунку 2.3.

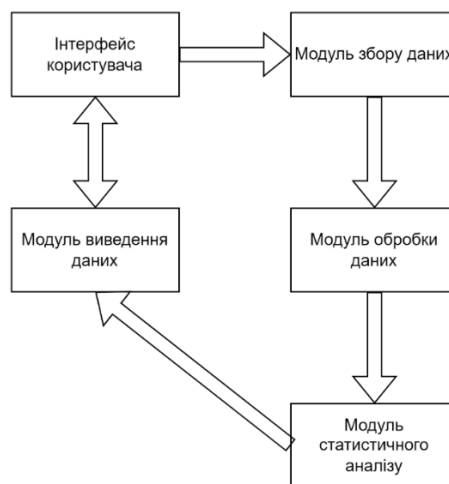


Рисунок 2.3 – Структура інформаційної технології аналітики комп'ютерної гри Dota 2

Отже, запропонована структура інформаційної технології аналітики комп'ютерної гри Dota 2 дозволить розширити функціонал засобу, а саме підбір персонажа та спорядження за рахунок використання статистичного аналізу.

2.4 Висновок до розділу 2

У даному розділі було удосконалено математичну модель, яка на відміну від існуючих відрізняється введенням додаткових критеріїв аналітики, що дозволило розширити функціонал інформаційної технології. Розроблено удосконалений алгоритм аналітики комп'ютерної гри Dota 2 та структуру інформаційної технології.

3 РОЗРОБКА МОДУЛІВ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІТИКИ КОМП'ЮТЕРНОЇ ГРИ DOTA 2

3.1 Обґрунтування вибору мов програмування

Вибір мови програмування для розробки інформаційної технології аналітики комп'ютерної гри Dota 2 має велике значення для успішності проекту. Правильно підібрана мова дозволяє оптимізувати роботу розробників, полегшує розширення та підтримку програми, а також забезпечує високий рівень безпеки та продуктивності.

Мови програмування, які використовуються для розробки подібних інформаційних технологій організації заходів, включають, але не обмежуються такими мовами як C#, Python, JavaScript, PHP, Java та C++. Кожна з цих мов має свої особливості та переваги, і вибір зазвичай залежить від конкретних вимог проекту, потреб користувачів та навичок розробників.

JavaScript [14] (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Особливості JavaScript:

- JavaScript виконується в браузері користувача, що дозволяє створювати взаємодію та змінювати вміст веб-сайтів без перезавантаження сторінки.
- JavaScript підтримує об'єктно-орієнтоване програмування, що дозволяє створювати код в об'єктно-орієнтованому стилі.
- Сучасні версії JavaScript (ES6 і вище) пропонують багато нових можливостей, включаючи стрілкові функції, деструктуризацію, класи та інше.
- JavaScript можна використовувати на різних платформах, не тільки в браузерах, але і на сервері, за допомогою платформи Node.js.
- JavaScript дозволяє легко взаємодіяти з DOM (Document Object Model), що дозволяє динамічно змінювати вміст та структуру веб-сторінки.
- Підтримка асинхронного програмування дозволяє взаємодіяти з сервером та виконувати інші завдання, не блокуючи виконання коду.

Переваги мови JavaScript:

- Тип даних визначається, коли змінній або константі присвоюється значення.
- В JavaScript функції можна як виконувати, так і повертати, передавати їх як параметри іншим функціями і привласнювати як значення змінних.
- Методологія об'єктно-орієнтованого програмування дає змогу представити програму у вигляді сукупності об'єктів.
- Дозволяє частково перенести бізнес-логіку із сервера на сторону користувача, тобто виконувати код в браузері, що своєю чергою зменшує навантаження на сервери.
- JavaScript має розвинену інфраструктуру та активну спільноту. Так, веб-розробники можуть працювати з великою кількістю бібліотек і фреймворків як React, Angular і Vue, декількома пакувальниками, як Webpack, Gulp, та допоміжними бібліотеками як Lodash, axios, та іншими.

Недоліки мови JavaScript:

- проблема роботи із типами даних, проявляється у тому, що усі помилки можна побачити лише після компіляції та запуску програми.
- можливості мови досить обмежені, для їх розширення необхідно підключати або створювати сторонні бібліотеки та фреймворки.
- Виконання в браузері обмежує доступ до системних ресурсів, що може бути проблемою для деяких завдань.
- Безпека коду JavaScript обмежена браузерним середовищем, тому розробники повинні враховувати потенційні загрози безпеки.
- JavaScript може бути менш ефективним для важких обчислень порівняно з іншими мовами.
- Управління асинхронним кодом може призводити до складних сценаріїв та збільшувати ймовірність помилок.

C# [15] – об'єктно-орієнтована мова програмування, за допомогою якої можна розробити різного роду високопродуктивні та безпечні, в плані захищеності коду від помилок, додатки. Мова C# реалізована на платформі Microsoft.NET [16]. На мові C# можна створювати різноманітні Windows додатки, клієнт-серверні додатки, XML веб-служби, ігри, тощо.

Зручний редактор коду, з автоматичним пошуком лексичних та синтаксичних помилок, простий редактор для створення віконних додатків Windows, також вбудований відладчик і ще безліч інших засобів, які спрощують розробку програмного продукту.

Головною особливістю мови C#, яку вона запозичила у мови Java, є технологія віртуальної машини. За допомогою даної технології програма компілюється не в машинний, а в проміжний, так званий байт-код в мові Java та IL код в мові C#. Перевагою подібного виконання програм є повна незалежність від операційної системи, обладнання та зумовлений цим високий рівень безпеки. Така технологія дозволяє виконувати додатки на будь-якому пристрої з підтримкою віртуальної машини. Віртуальна машина мови Java отримала назву JVM [17] (Java Virtual Machine), а мови C# – .NET Framework [18].

Особливості C#:

- C# підтримує об'єктно-орієнтоване програмування, що дозволяє розробникам створювати модульний та організований код.
- Мова призначена для використання на платформі .NET, що відкриває доступ до широкої екосистеми бібліотек та інструментів.
- C# має підтримку асинхронного програмування, що полегшує роботу з асинхронними операціями та запитами.
- C# має строгую систему типізації, що сприяє виявленню помилок на етапі компіляції, а не під час виконання. Він також надає засоби для роботи з пам'яттю та обробки винятків.
- Мова має вбудовану підтримку Language Integrated Query (LINQ), яка дозволяє виконувати складні запити до даних.
- C# може використовуватися для розробки різних видів додатків, включаючи веб-додатки, десктопні програми, мобільні додатки та ігри.
- Широкий спектр можливостей мови включає делегати, події, атрибути, патерни та інші розширені концепції.

Переваги мови C#:

- Автоматична збірка сміття.
- Асинхронна модель виконання коду.
- Строга типізація, отже легше опанувати початківцям.
- Інтеграція з платформою .NET надає доступ до потужних бібліотек та інфраструктури для розробки додатків.
- C# дозволяє розробникам створювати розширені та потужні додатки, використовуючи сучасні підходи до програмування.
- З кожним оновленням мови вводяться нові функції та покращення синтаксису, що сприяє читабельності та розширенню мови.

Недоліки мови C#:

- Високий рівень входу для новачків.

- Велика кількість технологій і підходів до розробки десктопних програм і без попереднього аналізу важко сказати який з фреймворків краще підходить до вирішення тієї чи іншої задачі.
- Періодична «смерть» технологій в рамках платформи.
- Хоча є платформа .NET Core для кросплатформенної розробки, деякі функції можуть бути обмежені порівняно зі стандартним .NET.
- Використання продуктів та інфраструктури Microsoft може вимагати великої вартості, особливо для комерційних проектів.
- Спільнота C# менша порівняно з іншими популярними мовами, такими як JavaScript чи Python.

Java [19] – об'єктно-орієнтована мова програмування, випущена 1995 року компанією Sun Microsystems.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформи-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

Головним мотивом створення Java була потреба в мові програмування, яка б не залежала від платформи (тобто від архітектури) і яку можна було б використовувати для створення програмного забезпечення, що вбудовується в різноманітні побутові електронні прилади, такі як мобільні засоби зв'язку, пристрої дистанційного керування тощо.

У створенні мови програмування Java було чотири початкові цілі:

- Синтаксис мови повинен бути «простим, об'єктно-орієнтованим та звичним».

- Реалізація має бути «безвідмовною та безпечною».
- Повинна зберегтися «незалежність від архітектури та переносність».
- Висока продуктивність виконання
- Мова має бути «інтерпретованою, багатонитевою, із динамічним зв'язуванням модулів».

Під «незалежністю від архітектури» мається на увазі те, що програма, написана на мові Java, працюватиме на будь-якій підтримуваній апаратній чи системній платформі без змін у початковому коді та перекомпіляції.

Цього можна досягти, компілюючи початковий Java код у байт-код, який є спрощеними машинними командами. Потім програму можна виконати на будь-якій платформі, що має встановлену віртуальну машину Java, яка інтерпретує байткод у код, пристосований до специфіки конкретної операційної системи і процесора. Зараз віртуальні машини Java існують для більшості процесорів і операційних систем.

Основна перевага використання байт-коду – це портативність. Тим не менш, додаткові витрати на інтерпретацію означають, що інтерпретовані програми будуть майже завжди працювати повільніше, ніж скомпільовані у машинний код, і саме тому Java одержала репутацію «повільної» мови. Проте, цей розрив суттєво скоротився після введення декількох методів оптимізації у сучасних реалізаціях JVM.

Програми на Java утворені з визначень класів та інтерфейсів. Класи містять змінні та константи, які утримують дані, методи, які виконують дії, та конструктори, які створюють екземпляри класів – об'єкти. Дані можуть мати простий тип (наприклад байт, ціле число, символ) або бути посиланням на об'єкт.

Java є суворо типізованою мовою, кожна змінна та вираз має тип, відомий на етапі компіляції.

Java стала логічним продовженням розвитку мови C++ та значно розширила можливості написання як складних, так і простих додатків.

Особливості Java:

- Java використовує віртуальну машину Java (JVM), що дозволяє виконувати код на різних платформах без необхідності перекомпіляції.
- Мова повністю об'єктно-орієнтована, що дозволяє розробникам створювати модульний та організований код.
- Java має вбудовані механізми безпеки, такі як обмежені права доступу та автоматичне управління пам'яттю.
- Java поставляється з великою стандартною бібліотекою, яка включає багато готових компонентів для розв'язання різних завдань.
- Мова підтримує многозадачність та можливості паралельного програмування.
- Java використовує автоматичне управління пам'яттю за допомогою системи збору сміття, що полегшує роботу розробників.
- Java використовується для розробки веб-додатків, мобільних додатків (Android), корпоративних систем, вбудованих систем та багатьох інших видів програм.

Переваги мови Java:

- Java легко вивчається новачками.
- У Java прийняті концепції хорошого програмування.
- Рідко змінюється.
- Доступ до великої колекції бібліотек з відкритим кодом.

Недоліки мови Java:

- Низька швидкість.
- Багатослівність (verbosity).
- Платна для комерційного користування (з 2019).

Python [20] – інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню

коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, аспектно-орієнтована та функціональна.

Особливості Python:

- Python відомий своєю читабельністю та простотою синтаксису, що полегшує написання та розуміння коду.

- Python є інтерпретованою мовою, що дозволяє виконувати код без необхідності компіляції.

- Підтримка об'єктно-орієнтованого програмування (ООП) дозволяє розробникам використовувати концепції класів та об'єктів для створення коду.

- Python поставляється з великою стандартною бібліотекою, що включає різноманітні інструменти та модулі для вирішення різних завдань.

- Мова використовує динамічну типізацію, що дозволяє зручно взаємодіяти з об'єктами та змінними без явного вказування типів.

- З великою кількістю модулів та пакетів сторонніх розробників Python дозволяє легко розширювати функціональність та використовувати готові рішення.

- Підтримка не тільки об'єктно-орієнтованого програмування, але і інших парадигм, таких як функціональне та імперативне програмування.

Переваги Python:

- Python має велику та активну спільноту розробників, а також розгалужену екосистему з багатьма корисними бібліотеками та фреймворками.

- Зручний синтаксис та велика кількість інструментів дозволяють розробникам бути дуже продуктивними.

- Python є кросплатформним, що дозволяє використовувати код на різних операційних системах без змін.

- Вбудована стандартна бібліотека містить багато корисних інструментів для вирішення різних завдань.

– Python використовується в різних областях, включаючи веб-розробку, науку про дані, штучний інтелект, автоматизацію та інше.

Недоліки Python:

– В порівнянні з деякими компільованими мовами, Python може бути менш ефективним у виконанні деяких завдань.

– GIL обмежує одночасне виконання потоків, що може вплинути на продуктивність для деяких типів програм.

– У порівнянні з деякими іншими мовами, підтримка мобільних платформ може бути менш розвиненою.

– Завдяки інтерпретованому характеру, деякі завдання можуть виконуватися повільніше, особливо для обчислювально-інтенсивних операцій.

PHP [21] – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні вебсервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веброзробок (разом із Java, .NET, JavaScript, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP — проєкт відкритого програмного забезпечення.

PHP інтерпретується вебсервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, тому що браузер отримує готовий html-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

Особливості PHP:

– PHP легко вбудовується в HTML-код, що полегшує створення динамічних веб-сторінок.

– PHP широко використовується в розробці веб-додатків і підтримується більшістю хостинг-постачальників.

– Мова має вбудовану підтримку для багатьох систем управління базами даних, таких як MySQL, PostgreSQL, SQLite, і багато інших.

– PHP використовує динамічну типізацію, що дозволяє гнучко взаємодіяти з різними типами даних.

– Мова має зручний синтаксис для обробки даних, що надходять з HTML-форм.

– PHP має велику та активну спільноту розробників, а також розширену екосистему з численними додатковими бібліотеками та фреймворками.

– PHP легко встановлюється і використовується, що робить його доступним для широкого кола розробників.

Переваги PHP:

– PHP ідеально підходить для створення динамічних веб-сайтів, оскільки він може генерувати HTML на льоту.

– Завдяки великій спільноті, завдяки якій легко знайти підтримку та рішення для будь-яких проблем.

– Широкий вибір функцій для роботи з різними базами даних полегшує взаємодію з даними.

– PHP дозволяє швидко розгортати веб-проекти, що робить його зручним для невеликих та середніх проектів.

– Багато веб-сайтів та веб-додатків використовують PHP, що свідчить про його популярність.

Недоліки PHP:

– Використання глобальних змінних може призвести до проблем безпеки та неочікуваної поведінки.

– Іноді робота з PHP викликає складнощі в інтеграції з веб-дизайном порівняно з сучасними фронтенд-фреймворками.

– Деякі розробники вважають, що код на PHP може бути менш організованим та чистим порівняно з іншими мовами.

– У порівнянні з іншими мовами, PHP може бути менш ефективним щодо швидкодії, особливо для важких обчислень.

C++ [22] – мова програмування загального призначення з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої,

процедурної та ін. Б'ярн Страуструп почав створювати C++ в AT&T Bell Laboratories у 1979 році. На етапі зародження мова мала назву «Сі з класами». Згодом Страуструп перейменував мову на C++ у 1984 р.

У 1990-х роках C++ стала однією з найуживаніших мов програмування загального призначення. Мову використовують для системного програмування, розробки прикладного програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм, наприклад, відеоігор. C++ суттєво вплинула на інші популярні сьогодні мови програмування: C# та Java.

Особливості C++:

- C++ підтримує об'єктно-орієнтоване програмування, що дозволяє використовувати концепції класів та об'єктів для створення коду.

- Мова дозволяє розробникам працювати на низькорівневому рівні, що корисно для системного програмування та оптимізації.

- C++ підтримує шаблони, що дозволяє створювати загальні алгоритми та структури даних.

- Мова пропонує близьку до мови С ефективність та швидкодію, що робить її популярною для високопродуктивних додатків.

- C++ підтримує не тільки ООП, але і імперативне та функціональне програмування.

- C++ має багато функцій у своїй стандартній бібліотеці, що включає в себе алгоритми, контейнери, введення/виведення та інші.

Переваги C++:

- C++ відома своєю високою швидкістю та ефективністю, що робить її відмінним вибором для вимогливих до ресурсів застосунків, таких як ігри та системне програмування.

- Мова дозволяє розробникам працювати на низькорівневому рівні, контролюючи безпосередньо пам'ять та ресурси.

- Використання шаблонів дозволяє створювати загальні та перевикористовувані компоненти коду.

– Підтримка ООП дозволяє розробникам створювати чистий та організований код.

– Розробникам надається більше контролю над управлінням пам'яттю, що дозволяє оптимізувати ефективність програм.

Недоліки C++:

– Вивчення та використання C++ може бути складним для новачків через його розширену функціональність та складний синтаксис.

– З більшим рівнем контролю приходить більше можливостей для помилок, особливо пов'язаних з управлінням пам'яттю.

– Відсутність автоматичного сміттєзбірника може призводити до витоку пам'яті та інших проблем, пов'язаних із керуванням пам'яттю.

– Код, написаний на C++, може бути менш переносимим між різними платформами через особливості компіляторів та архітектури.

Ознайомившись з особливостями описаних мов програмування, у таблиці 3.1 виділено основні переваги та недоліки кожної з них.

Таблиця 3.1 – Порівняння мов програмування Java, JavaScript і C#

Характеристика / Мова програмування	C#	Python	JavaScript	Java	C++	PHP
Відкритий код	-	+	+	-	-	+
Кросплатформеність	+	+	+	+	-	+
Об'єктно-орієнтованість	+	+	+	+	+	+
Інтеграція зі сторонніми мовами	+	+	+	+	-	+
Велика спільнота розробників	+	+	+	+	+/-	+/-
Широка бібліотека	+	+	+	+	+	+
Простота вивчення	+	+	+	+/-	-	+/-
Швидкість опрацювання даних	+	+	+	+/-	-	+

Отже, проаналізувавши характеристики мов програмування C#, Python, JavaScript, PHP, Java та C++, було обрано мову програмування JavaScript. Вибір JavaScript для розробки інформаційної технології аналітики комп'ютерної гри Dota 2 обґрунтований через її інтеграційні можливості, швидкість, а також можливості розширення. Ця мова програмування найкраще задовольняє потреби розробки зазначеної інформаційної технології, забезпечуючи кросплатформеність, зручність розробки, доступність ресурсів та продуктивність.

3.2 Вибір середовища розробки

Для розробки програмного забезпечення існує ряд різних середовищ розробки. Вибір середовища напряму залежить від типу програмного забезпечення яке розроблюється. Так для створення прикладних програм є доцільним використання середовищ таких як Visual Studio, для розробки мобільних додатків таких як Android Studio.

Є кілька можливих середовищ розробки для мови JavaScript, таких як Visual Studio, Visual Studio Code, WebStorm. Розглянемо їх особливості, переваги та недоліки.

1) Visual Studio [23] – це інтегроване середовище розробки (IDE) від Microsoft, призначене для розробки різних типів програм, включаючи веб-додатки.

Особливості Visual Studio:

- Visual Studio підтримує різноманітні мови програмування, такі як C#, Visual Basic, C++, F#, JavaScript, TypeScript та інші.

- Має сильну інтеграцію з платформою .NET, що полегшує розробку десктопних та веб-додатків на цій платформі.

- Visual Studio підходить для розробки різноманітних типів додатків, включаючи веб-додатки, десктопні застосунки, мобільні додатки, ігри тощо.

- Інтеграція з хмарною платформою Azure, що полегшує розгортання та управління хмарними рішеннями.

- Вбудовані інструменти для автоматизації тестування, включаючи юніт-тести, навігаційні тести, тести динамічної перевірки тощо.

- Зручний інтерфейс для роботи з системами контролю версій, зокрема Git.

- Широкий вибір розширень та додатків, які полегшують роботу з різними технологіями та фреймворками.

Переваги Visual Studio:

- Visual Studio підтримує розробку додатків для різних операційних систем, включаючи Windows, macOS, і Linux.

- Інтеграція із сервісами Azure DevOps сприяє ефективній спільній роботі команд розробників.

- Можливості налагодження коду, включаючи підтримку віддаленого налагодження.

- Має велику та активну спільноту користувачів і широку базу знань.

Недоліки Visual Studio:

- Іноді великий обсяг функцій може призводити до великих вимог до ресурсів комп'ютера.

- Як інструмент від корпорації Microsoft, Visual Studio може бути вартісним, особливо для комерційних версій.

- Для деяких користувачів інтерфейс Visual Studio може вважатися завантаженим та складним.

Загалом, Visual Studio - це потужне інструментарій з багатьма можливостями, який підходить для широкого спектру розробок, включаючи веб-розробку.

2) WebStorm [24] - це інтегроване середовище розробки (IDE) від JetBrains, спроектоване для роботи з веб-технологіями, зокрема JavaScript, HTML і CSS.

Особливості WebStorm:

– Підтримка мов програмування. WebStorm підтримує широкий спектр мов програмування, таких як JavaScript, TypeScript, HTML, CSS, Sass, Less, CoffeeScript і багато інших.

– Автоматизація завдань. Інтеграція з інструментами автоматизації завдань, такими як Grunt, Gulp і npm, щоб полегшити процес розробки і підтримки проекту.

– Рефакторинг коду. WebStorm надає широкий набір інструментів для рефакторингу коду, що полегшує покращення якості та зручності коду.

– Вбудовані інструменти для фронтенд-розробки. Інтеграція з системами контролю версій, такими як Git, а також можливості роботи з фреймворками, такими як Angular, React, і Vue.js.

– Live Templates. Можливість визначити та використовувати власні шаблони для швидкого написання коду.

– Live Editing. Зміни в коді відображаються миттєво в браузері без перезавантаження сторінки (за умови, що використовується LiveEdit).

– Робота з Node.js. Вбудована підтримка для Node.js з можливістю виконання та налагодження серверних додатків.

Переваги WebStorm:

– Повна інтеграція з іншими продуктами JetBrains, що дозволяє зручно працювати в одному середовищі з різними технологіями.

– Швидка підтримка нових версій мов та технологій.

– Потужні інструменти для автоматизації процесу рефакторингу коду.

– WebStorm славиться своєю продуктивністю та швидкістю роботи.

Недоліки WebStorm:

– WebStorm є комерційним продуктом, і вартість ліцензії може бути високою для окремих розробників або невеликих команд.

– Завдяки багатофункціональності, WebStorm може вимагати відносно великих ресурсів комп'ютера, що може бути проблемою для менш потужних систем.

– Для новачків може знадобитися трошки часу для засвоєння всіх функцій та можливостей WebStorm.

В цілому, WebStorm є потужним інструментом для розробки веб-додатків з великою кількістю корисних функцій, які полегшують процес розробки та підтримки коду.

3) Visual Studio Code [25] - це легковагове та потужне інтегроване середовище розробки (IDE) від Microsoft.

Особливості Visual Studio Code:

– Visual Studio Code є легковаговим та швидким інструментом, що дозволяє швидко встановлювати та запускати.

– VS Code підтримує різноманітні мови програмування, включаючи JavaScript, TypeScript, Python, Java, і багато інших.

– Широкий вибір розширень для підтримки різних технологій та фреймворків.

– Інтегрована підтримка Git спрощує ведення систем контролю версій.

– Функціональність Live Share дозволяє спільно працювати над кодом з іншими розробниками в реальному часі.

– Функція автодоповнення коду, яка полегшує написання коду та зменшує кількість помилок.

– Має вбудовані інструменти для налагодження коду.

Переваги Visual Studio Code:

– VS Code є безкоштовним та відкритим за кодом, що робить його доступним для широкого кола користувачів.

– Інтерфейс користувача VS Code інтуїтивно зрозумілий та простий у використанні.

- Велика спільнота розробників створює багато розширень, що полегшує адаптацію IDE до конкретних потреб.

- Можливість ведення спільної роботи над проектами завдяки функціональності Live Share.

- Висока швидкість запуску та відкриття проектів.

Недоліки Visual Studio Code:

- Для дуже великих проектів може бути менше зручним порівняно з іншими більш розгалуженими IDE.

- У порівнянні з повноцінним Visual Studio, VS Code має менше функціональності, яка може бути важливою для певних типів розробок.

Загалом, Visual Studio Code є високоефективним та легким використанням інструментом для розробників з широким спектром функціональності та підтримкою різних технологій.

Отже, враховуючи характеристики описаних середовищ розробки та ресурси комп'ютера, прийнято рішення використовувати Visual Studio Code для розробки програмного забезпечення для інформаційної технології аналітики комп'ютерної гри Dota 2.

3.3 Розробка інтерфейсу інформаційної технології аналітики комп'ютерної гри Dota 2

Графічний інтерфейс користувача – тип інтерфейсу, який дає змогу користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки, на відміну від текстових інтерфейсів, заснованих на використанні тексту, текстовому наборі команд та текстовій навігації.

Виконання дій у ГІК – це безпосередня маніпуляція з графічними елементами. Окрім комп'ютерів, ГІК використовують у мобільних пристроях, таких, як мобільні телефони, планшети, електронні книги, портативні медіапрогравачі тощо.

Для розробки графічного інтерфейсу був використаний програмний засіб Figma [26]. Figma — це векторний онлайн-сервіс розробки інтерфейсів. Працює як у браузері так і як десктопний додаток.

Вибір Figma як основного інструменту для дизайну та розробки інтерфейсу був обґрунтований кількома факторами. Перш за все, це хмарна платформа, що дозволяє працювати під одним акаунтом онлайн з різних девайсів, незалежно від місцезнаходження. За потреби, є можливість відкривати доступ до макетів для інших користувачів та робити збереження версій для відстеження змін. Також Figma підтримує сторонні плагіни, які можна встановлювати для розширення функціоналу та полегшення деяких завдань. Крім того, Figma надає можливість створювати прототипи та деталізовані макети, що допомагає візуалізувати концепції та дизайн інтерфейсу перед його реалізацією. Для цього платформа надає користувачеві можливість використати вбудовані симулятори реальних девайсів: смартфонів, планшетів чи ноутбуків. Важливою перевагою Figma є також можливість залишати коментарі в процесі роботи щодо можливих майбутніх змін чи покращень макету, що сприяє постійному вдосконаленню та адаптації дизайну під потреби користувачів і бізнес-цілі проекту.

На рисунках 3.1 – 3.5 Зображено основні скріншоти інтелектуального модуля оплати за навчання.

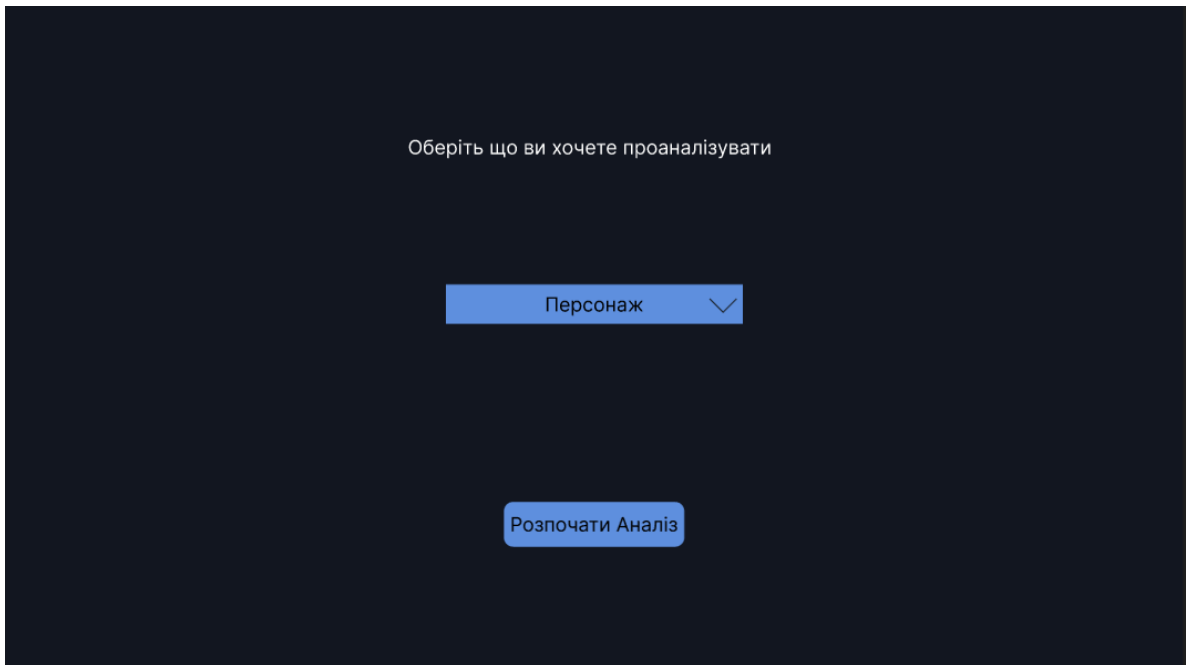


Рисунок 3.1 – Головна сторінка

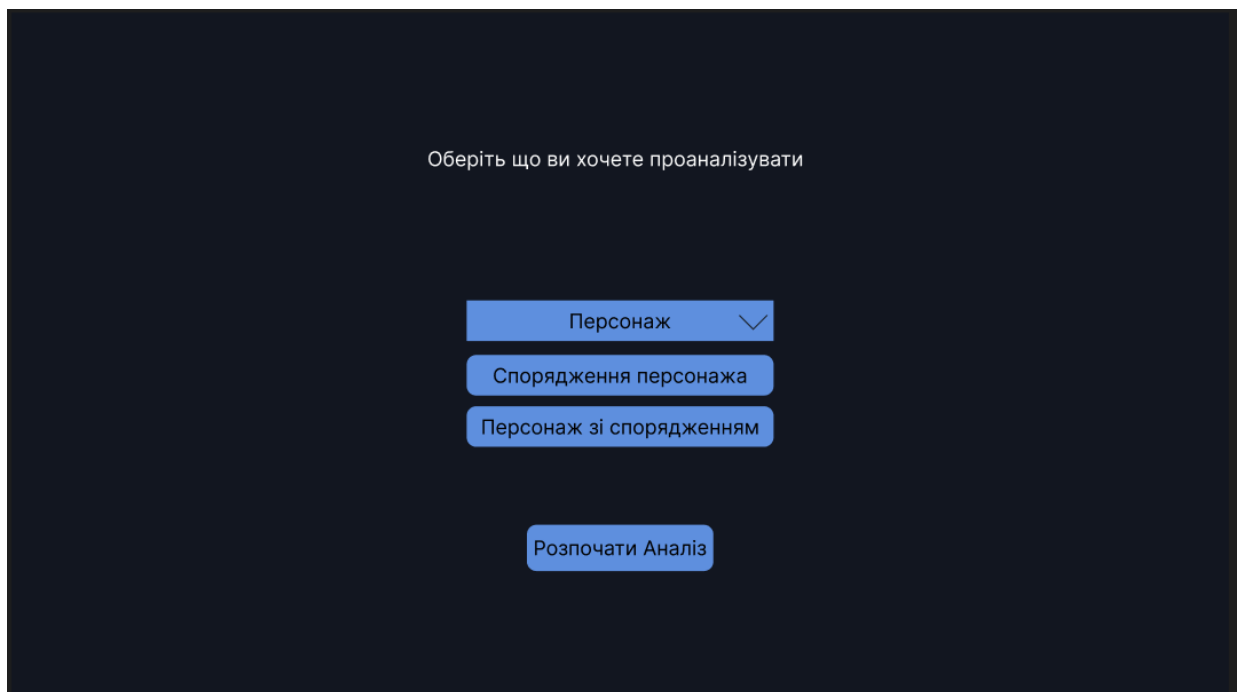


Рисунок 3.2 – Вибір об'єкта аналізу

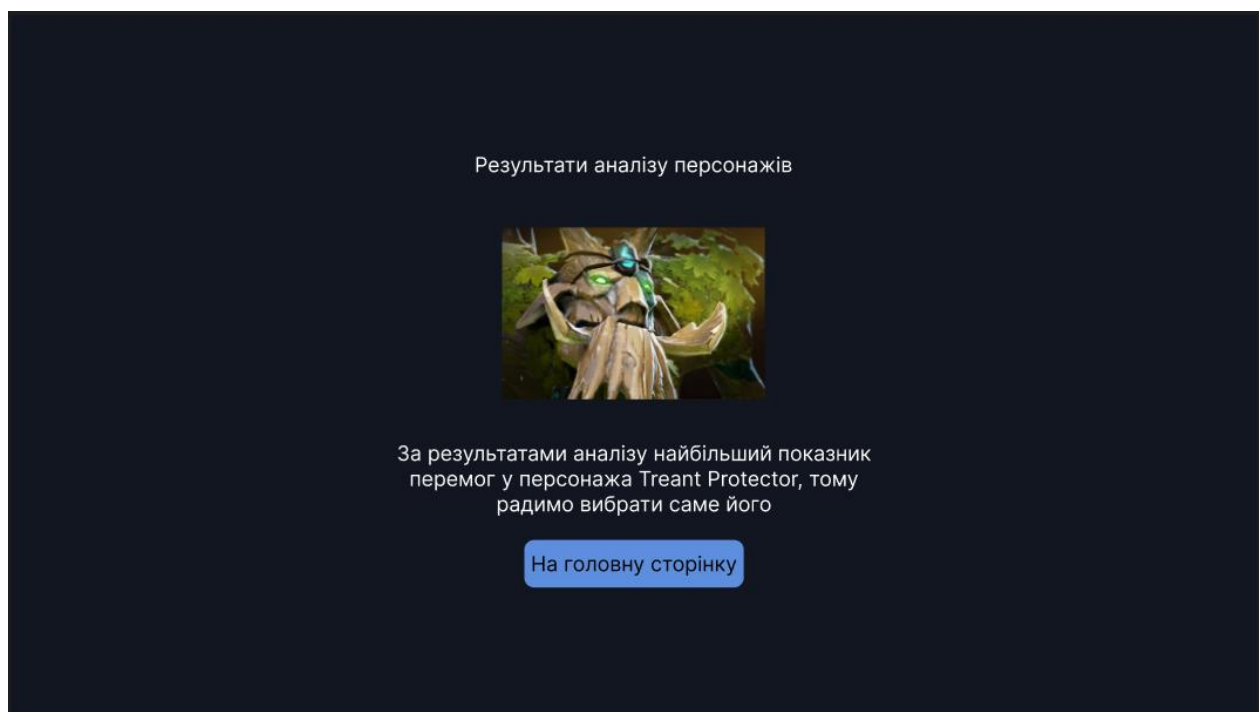


Рисунок 3.3 – Результат аналізу персонажів

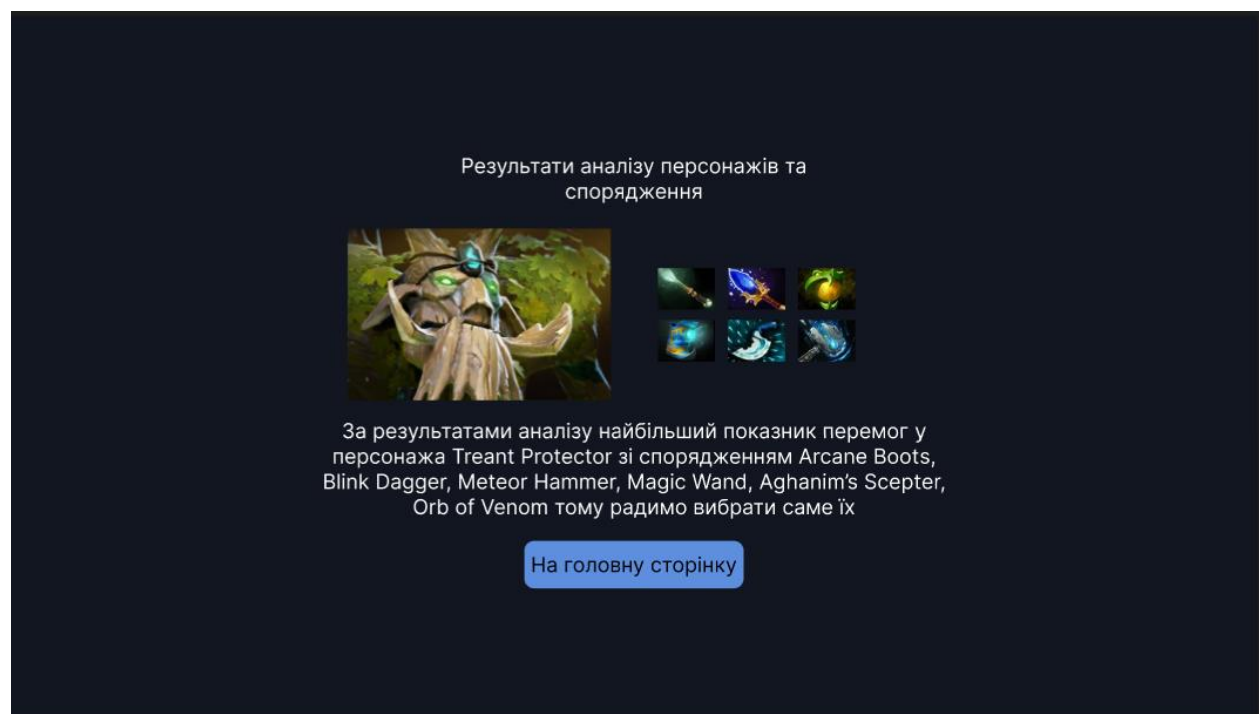


Рисунок 3.4 – Результат аналізу персонажів та спорядження

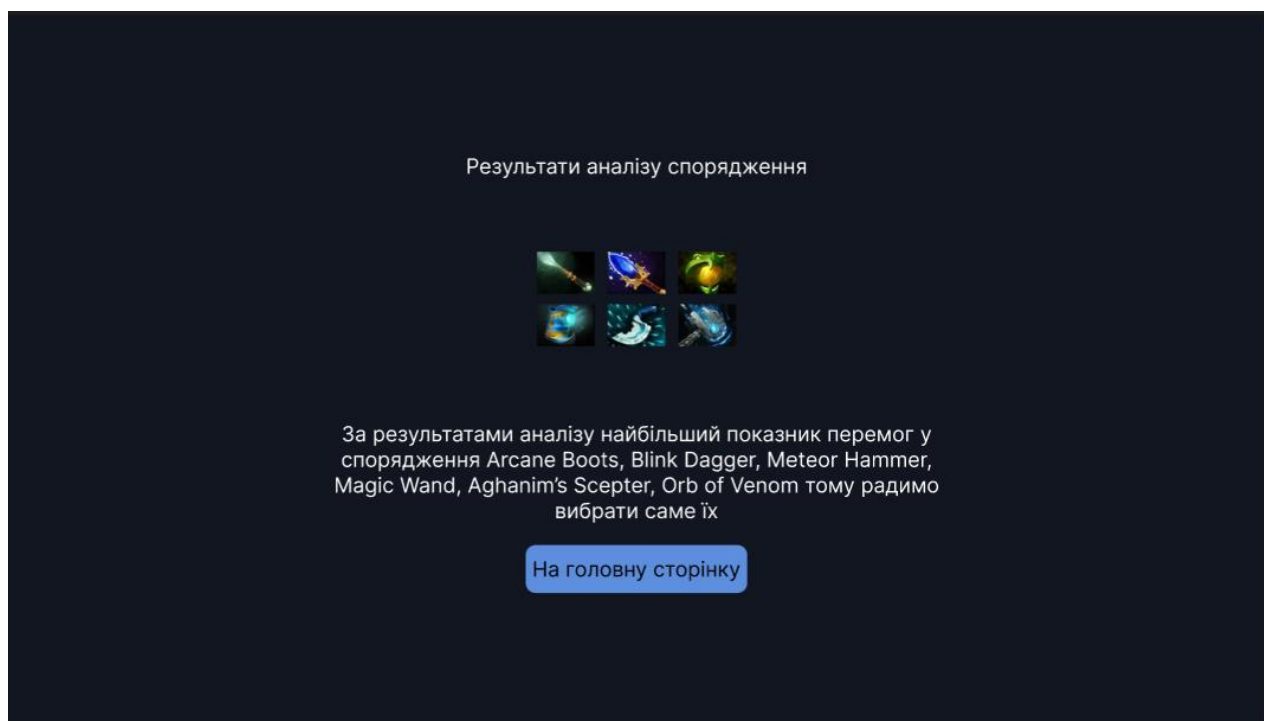


Рисунок 3.5 – Результат аналізу спорядження

Отже, у даному підрозділі було розроблено інтерфейс користувача інформаційної технології аналітики комп'ютерної гри Dota 2.

3.4 Програмна реалізація модуля збору даних

Модуль збору даних є важливою складовою інформаційної технології аналітики комп'ютерної гри Dota 2. Цей модуль відповідає за збір різних видів ігрової інформації під час гри в режимі реального часу або з історичних джерел. Алгоритм функціонування такого модуля складатиметься з таких кроків:

Крок 1. Встановлення підключення:

- Реалізація логіки для встановлення підключення до джерела даних, незалежно від того, чи це API, сторонні сервіси або історичні дані.

Крок 2. Збір інформації:

- Організація процесу збору даних, включаючи запити до API або обробку історичних даних.

- Збір інформації про гравців, команди, результати ігор, статистику героїв, інше.

Крок 3. Передача даних в модуль обробки даних.

UML-діаграма роботи модуля наведена на рисунку 3.6

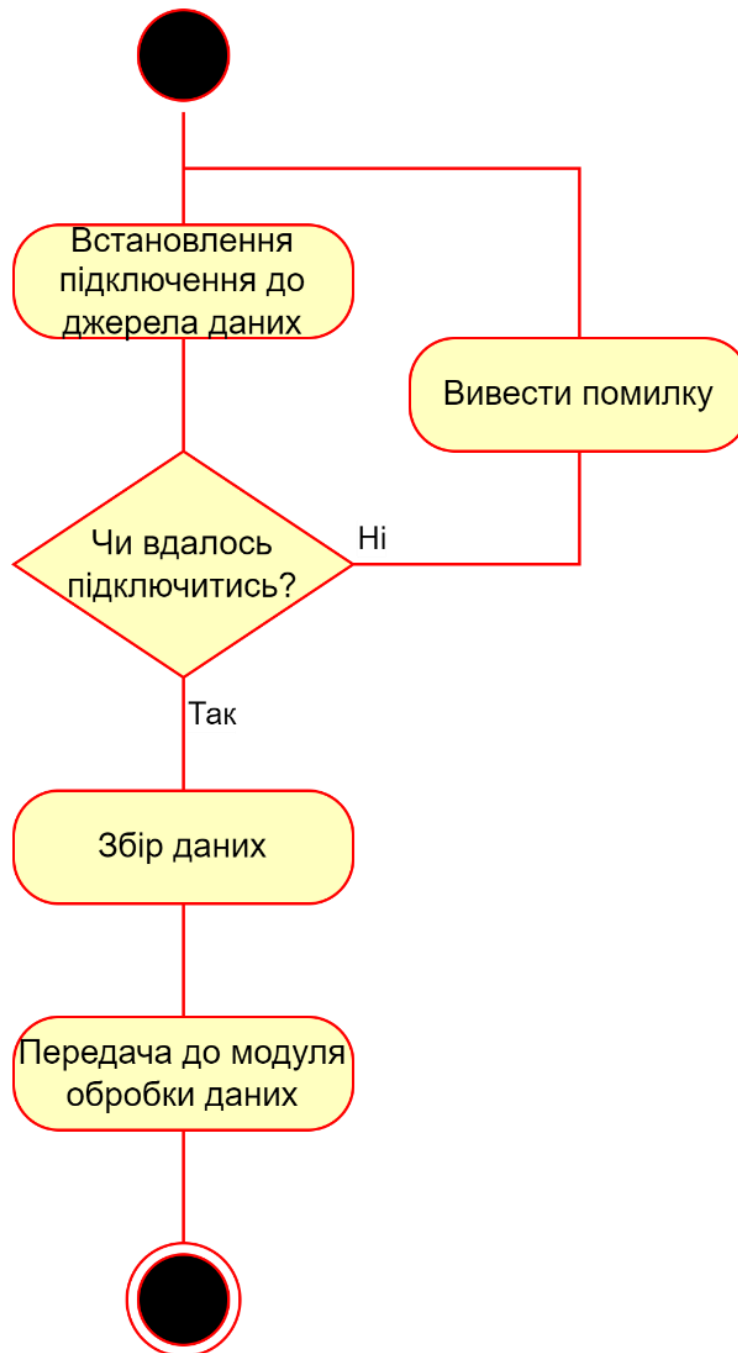


Рисунок 3.6 – UML-діаграма роботи модуля збору даних

Фрагмент лістингу коду, який виконує модуль збору даних:

```
const axios = require('axios');

// Ключ API для доступу до даних Dota 2
const apiKey = 'Ваш ключ API з https://www.opendota.com/api-keys';

async function getHeroes() {
  try {
    const response = await
axios.get('https://api.steampowered.com/IEconDOTA2_570/GetHeroes/v0001/', {
      params: {
        key: apiKey
      }
    });
    return response.data;
  } catch (error) {
    console.error('Помилка при отриманні даних про героїв', error);
    return null;
  }
}
```

Отже, у даному підрозділі було виконано програмну реалізацію модуля збору даних інформаційної технології аналітики комп'ютерної гри Dota 2.

3.5 Програмна реалізація модуля обробки даних

Модуль обробки даних відповідає за обробку та підготовку даних для подальшого аналізу. Алгоритм функціонування такого модуля складатиметься з таких кроків:

Крок 1. Прийом даних:

- Отримання даних з модуля збору даних.

Крок 2. Очищення та фільтрація даних:

- Видалення некоректних даних або аномальних значень.
- Виконання фільтрації даних, якщо потрібно вибрати лише певні записи або події.

Крок 3. Нормалізація даних:

- Приведення даних до єдиного формату і одиниць вимірювання.
- Здійснення нормалізації числових даних, щоб забезпечити їх порівнюваність.

Крок 4. Агрегація даних:

- Групування даних за потрібними параметрами (гравцями, командами, матчами, тощо) для проведення аналізу на рівні агрегованих даних.

Крок 5. Масштабування даних:

- Масштабування даних для забезпечення оптимальної продуктивності аналітичних алгоритмів, наприклад, шляхом стандартизації значень.

Крок 6. Передача даних в модуль статистичного аналізу.

UML-діаграма роботи модуля наведена на рисунку 3.7

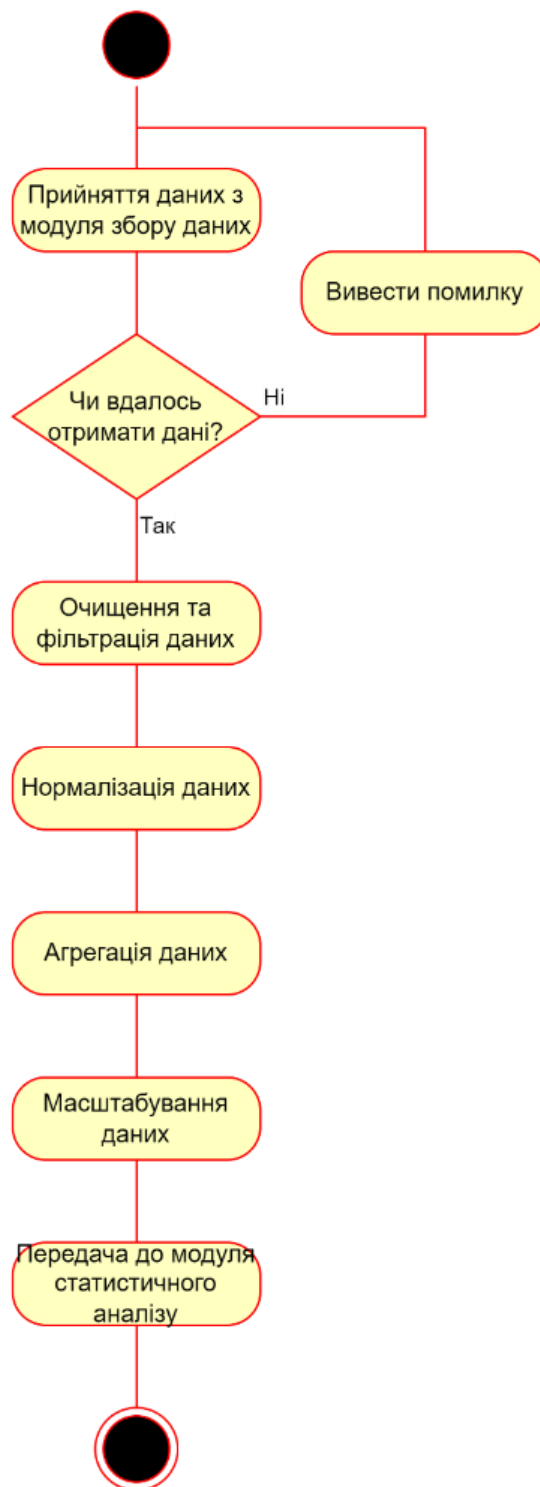


Рисунок 3.7 – UML-діаграма роботи модуля обробки даних

Фрагмент лістингу коду, який виконує модуль обробки даних:

```
const axios = require('axios');
```

```

const _ = require('lodash');
async function processData(apiUrl) {
  try {
    const rawData = await fetchDataFromAPI(apiUrl);
    const cleanedData = cleanData(rawData);
    console.log('Оброблені дані:', scaledData);
  } catch (error) {
    // Обробка помилок
  }
}
const apiUrl = 'https://example.com/api/data';
processData(apiUrl);

```

Отже, у даному підрозділі було виконано програмну реалізацію модуля обробки даних інформаційної технології аналітики комп'ютерної гри Dota 2.

3.6 Програмна реалізація модуля статистичного аналізу

Модуль статистичного аналізу є ключовим етапом в інформаційній технології аналізу комп'ютерної гри Dota 2. За допомогою статистичного аналізу він аналізує дані та приймає рішення щодо результатів аналізу. Ось алгоритм функціонування такого модуля:

Крок 1. Прийом даних для аналізу.

Крок 2. Підготовка даних для аналізу:

- Статистичні обчислення: Розрахунок основних статистичних параметрів, таких як середнє значення, медіана, дисперсія, стандартне відхилення тощо.

Крок 3. Визначення взаємозв'язків:

- Виявлення корисних взаємозв'язків, які можуть бути використані для покращення вибору персонажа чи спорядження.

Крок 4. Передача результатів у модуль виведення даних.

UML-діаграма роботи модуля представлена на рисунку 3.8

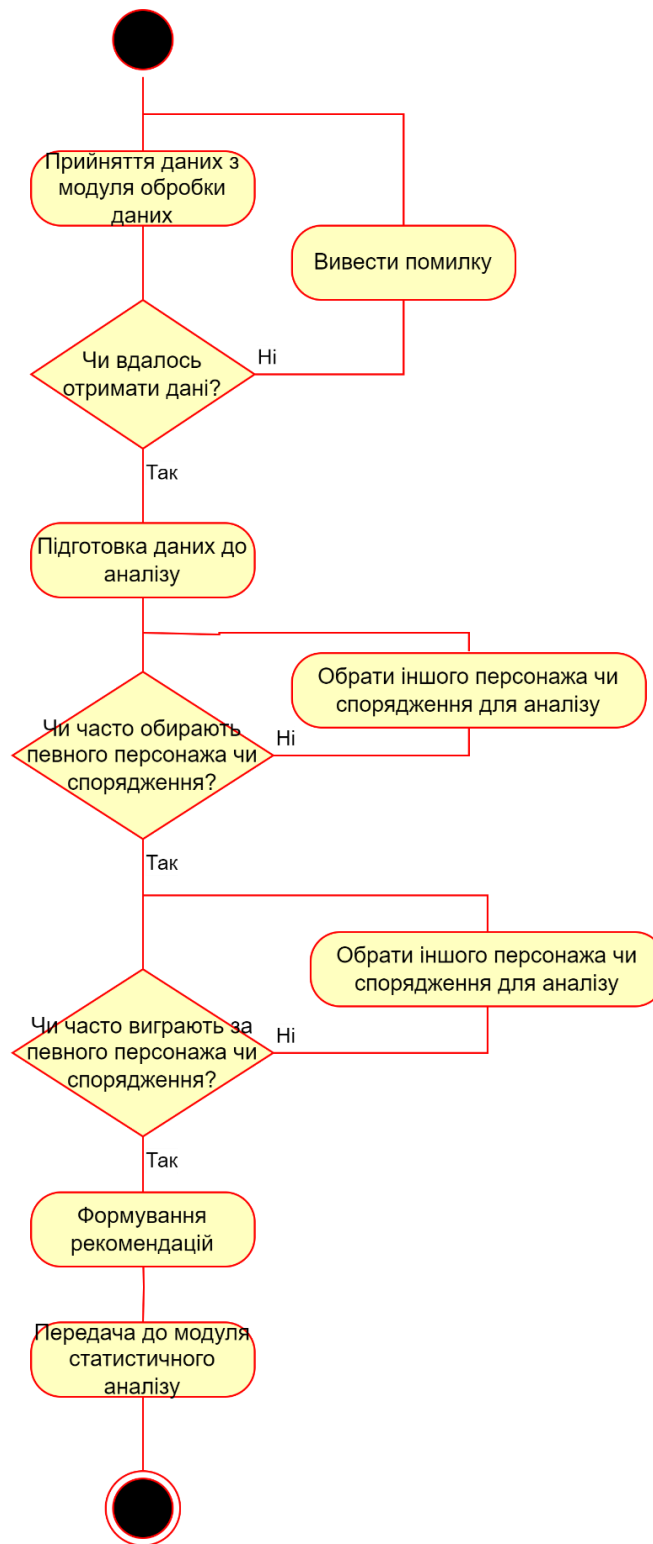


Рисунок 3.8 – UML-діаграма роботи модуля статистичного аналізу

Фрагмент лістингу коду, який виконує модуль статистичного аналізу:

```
async function main() {  
  const heroData = await getHeroes();  
  const itemsData = await getItems();  
  if (heroData && itemsData) {  
    const recommendation = recommendHeroAndItems(heroData, itemsData);  
    console.log('Рекомендація для героя:', recommendation.hero);  
    console.log('Рекомендація для спорядження:', recommendation.items);  
  }  
}
```

Отже, у даному підрозділі було виконано програмну реалізацію модуля статистичного аналізу інформаційної технології аналітики комп'ютерної гри Dota 2.

3.7 Програмна реалізація модуля виведення даних

Модуль виведення даних виводить дані на інтерфейс користувача. Алгоритм функціонування модуля складатиметься з таких кроків:

Крок 1. Прийом даних:

- Отримання даних з модуля статистичного аналізу.

Крок 2. Виведення результатів на інтерфейс користувача.

На рисунку 6 зображено схему алгоритму функціонування зазначеного модуля.

UML-діаграма роботи модуля представлена на рисунку 3.9

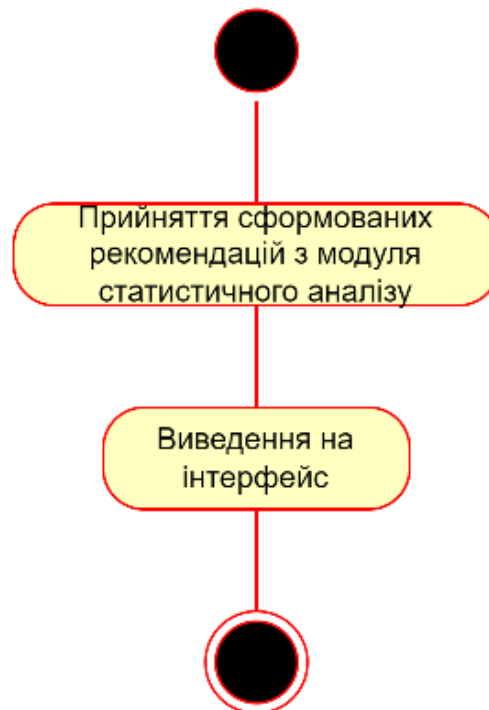


Рисунок 3.9 – UML-діаграма роботи модуля виведення даних

Фрагмент лістингу коду, який виконує модуль виведення даних:

```

function displayData(data) {
    const resultElement = document.getElementById('result');
    resultElement.innerHTML = `

Оброблені дані: ${data}</p>`;
}
const processedData = [42];
displayData(processedData);


```

Отже, у даному підрозділі було виконано програмну реалізацію модуля виведення даних інформаційної технології аналітики комп'ютерної гри Dota 2.

3.8 Тестування інформаційної технології аналітики комп'ютерної гри Dota 2

Відкриємо веб-сторінку з інформаційною технологією аналітики комп'ютерної гри Dota 2 на рисунку 3.10.

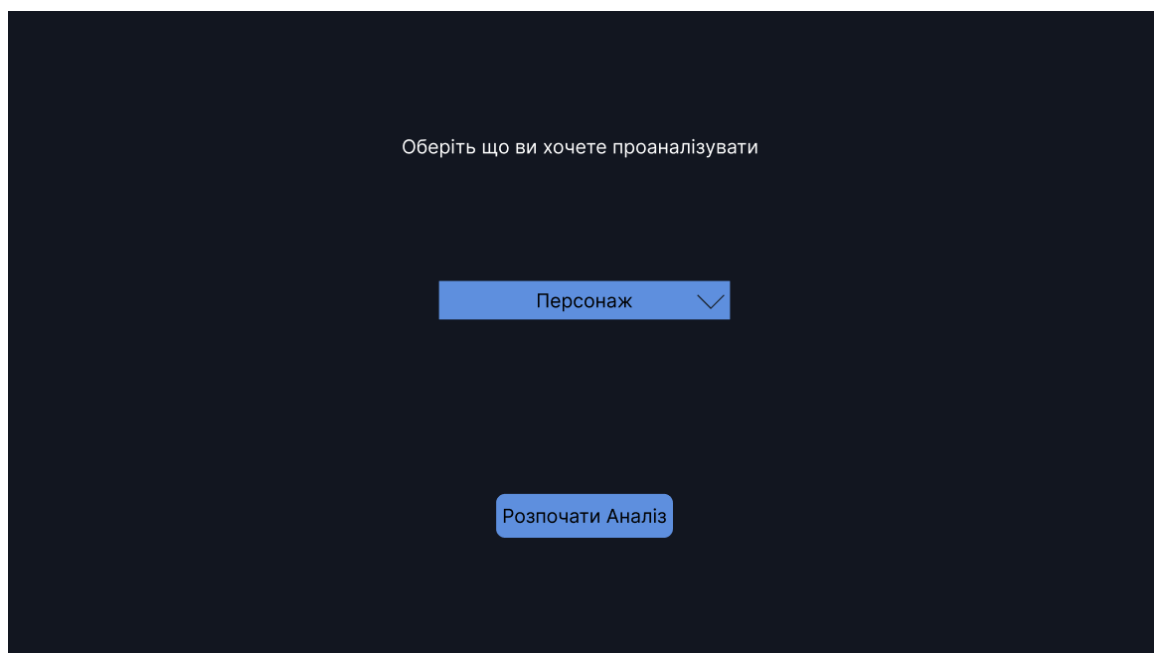


Рисунок 3.10 – Початкова сторінка інформаційної технології

Після входу, потрапляємо на головну сторінку, на якій відображено вибір що саме хоче проаналізувати гравець – якого персонажа краще обирати, яке спорядження чи і те і інше. Вибір зображено на рисунку 3.11.

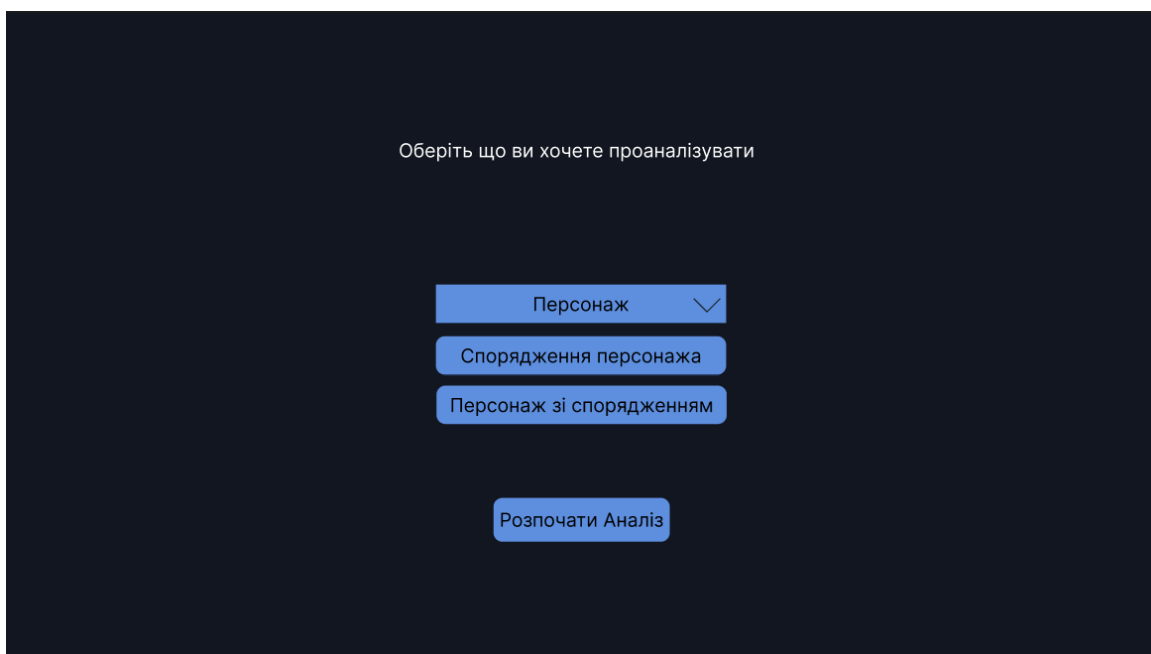


Рисунок 3.11 – Вибір об'єкта аналізу

Зробивши вибір і натиснувши кнопку «Розпочати Аналіз», гравцеві виведеться результат аналізу. В даному випадку виведеться рекомендація персонажа на рисунок 3.12.

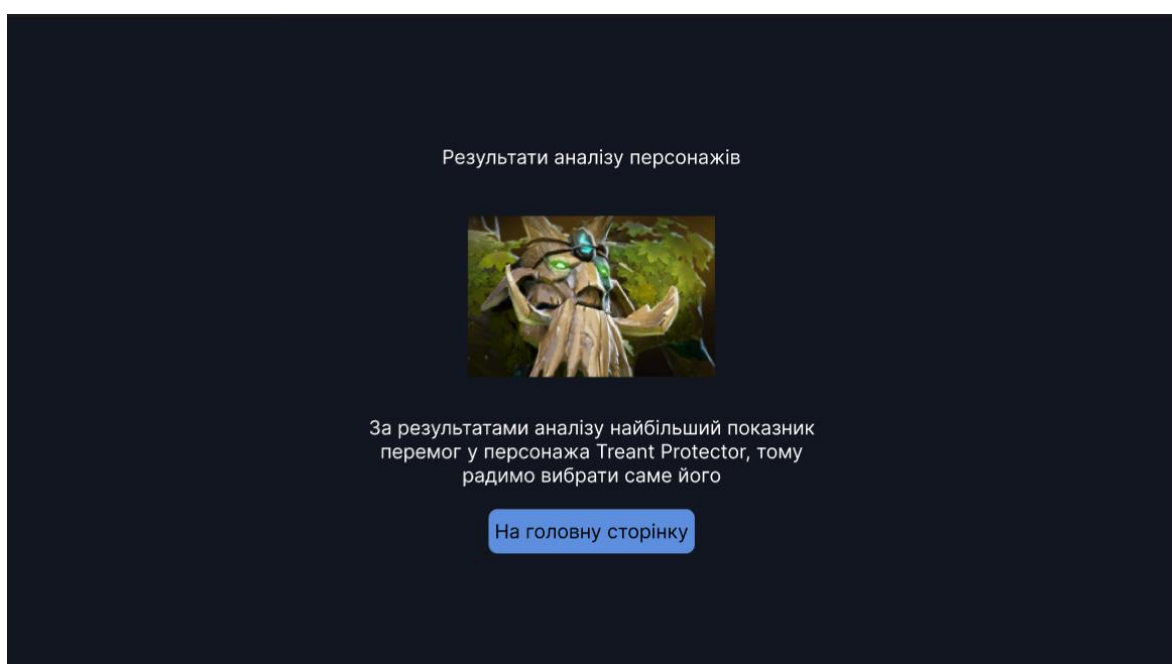


Рисунок 3.12 – Результат аналізу персонажа

Якщо ж обрати вибір спорядження, виведеться рекомендація спорядження на рисунку 3.13.

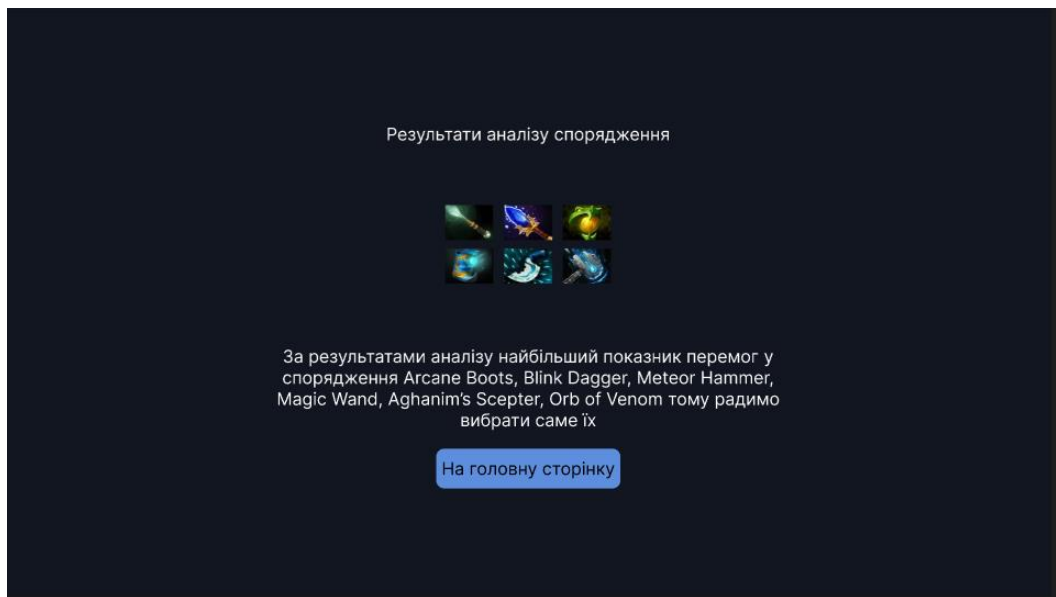


Рисунок 3.13 – Результат аналізу спорядження

Якщо ж обрати вибір персонажа зі спорядженням, виведеться відповідна рекомендація на рисунку 3.14

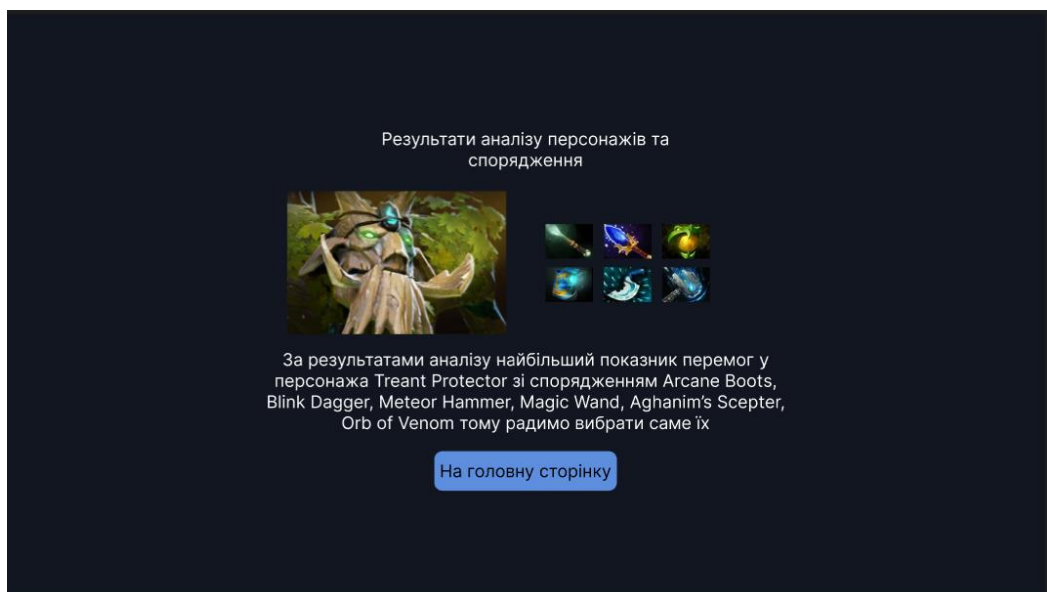


Рисунок 3.14 – Результат аналізу персонажів та спорядження

З метою тестування інформаційної технології було проведено понад 500 досліджень, а саме було проаналізовано та рекомендовано різних персонажів та спорядження відповідно до 4 вищенаведених правил аналітики.

Сутність кожного експерименту полягала у виборі з потужності множини персонажів, спорядження та персонажа зі спорядженням як предмет аналітики та проведенні процесу аналізу для кожного з них понад 100 разів.

Порівнюємо розроблену інформаційну технологію з аналогом Dotabuff за такими критеріями: швидкість роботи, наявність функціональної можливості вибору персонажа, наявність функціональної можливості вибору спорядження.

Порівняння наведено в таблиці 3.2.

Таблиця 3.2 – Порівняльна характеристика розробленої інформаційної технології з Dotabuff

Додаток	Швидкість роботи, с	Наявність функціональної можливості вибору персонажа	Наявність функціональної можливості вибору спорядження
Dotabuff	5	-	-
Інформаційна технологія аналітики комп'ютерної гри Dota 2	1	+	+

Таким чином, в результаті тестування програмного засобу, можна зробити висновок про те, що було розширено функціонал додатку, а саме функціональні можливості вибору комп'ютерного персонажа та його спорядження за скорочений час завдяки автоматизації цього процесу.

3.9 Висновок до розділу 3

Таким чином, у третьому розділі обґрунтовано вибір засобів для реалізації інформаційної технології аналітики комп'ютерної гри Dota 2.

Проаналізовано мови програмування, які найчастіше використовуються для реалізації подібних задач: C++, C#, Python, Java, JavaScript та PHP. Визначено їх особливості, переваги та недоліки, а також доцільність у використанні для поставленого завдання. В результаті аналізу для реалізації програмного модуля для інформаційної технології було обрано мову програмування JavaScript, адже вона об'єктно-орієнтована і безкоштовна, а також має весь необхідний функціонал.

Розглянуто особливості найпопулярніших середовищ для розробки додатків мовою JavaScript, а саме MS Visual Studio, MS Visual Studio Code та WebStorm. Врахувавши особливості кожного варіанту, обрано Microsoft Visual Studio Code в якості середовища для розробки.

В процесі розробки інтерфейсу для інформаційної технології аналітики комп'ютерної гри Dota 2 створено деталізовані макети окремих сторінок; розроблено прототип, який дозволив протестувати функціонал.

Результати тестування опрацьовано та порівняно з аналогом. Розширено функціонал додатку для вибору комп'ютерного персонажа та спорядження, що свідчить про те що мету дослідження було досягнуто.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту інформаційної технології аналітики комп'ютерної гри Dota 2.

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту залучаємо 3-х незалежних експертів, якими є провідні викладачі випускової або спорідненої кафедри.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу здійснюємо із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 4.1.

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу інформаційної технології аналітики комп'ютерної гри Dota 2

Критерії	Експерти		
	Яровий А. А.	Іванчук Я. В.	Савчук Т. О.
	Бали, виставлені експертами		
Технічна здійсненність концепції	1	2	2
Ринкові переваги (наявність аналогів)	2	3	2
Ринкові переваги (ціна продукту)	3	2	3
Ринкові переваги (технічні властивості)	2	1	2

Продовження таблиці 4.1

Ринкові переваги (експлуатаційні витрати)	3	2	2
Ринкові перспективи (розмір ринку)	1	3	3
Ринкові перспективи (конкуренція)	2	4	2
Практична здійсненність (наявність фахівців)	3	3	3
Практична здійсненність (наявність фінансів)	2	2	2
Практична здійсненність (необхідність нових матеріалів)	4	2	4
Практична здійсненність (термін реалізації)	2	2	3
Практична здійсненність (розробка документів)	2	3	2
Сума балів	27	29	30
Середньоарифметична сума балів, СБ	29		

За результатами розрахунків, наведених в таблиці 4.1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал інформаційної технології аналітики комп'ютерної гри Dota 2 – середній.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-

технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Витрати на основну заробітну плату дослідників (Z_o) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.1)$$

де k – кількість посад дослідників, залучених до процесу дослідження; M_{ni} – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.; T_p – число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні; t_i – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 4.2.

Таблиця 4.2 – Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	15000	714	50	35700
Розробник	12000	571	120	68520
Всього:				104220

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт розраховують за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.2)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год; t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_m \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.3)$$

де M_m – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), у 2023 році $M_m=6700$ грн; K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду; K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати; T_p – середня кількість робочих днів в місяці, приблизно $T_p = 21 \dots 23$ дні; $t_{зм}$ – тривалість зміни, год.

Зроблені розрахунки зводимо до таблиці 4.3.

Таблиця 4.3 – Витрати на заробітну плату робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Аналіз сучасних методів аналітики комп'ютерної гри Dota 2	80	4	50,6	1,27	4048
Аналіз сучасних засобів аналітики комп'ютерної гри Dota 2	80	4	50,6	1,27	4048

Продовження таблиці 4.3

Розробка удосконаленого алгоритму аналітики комп'ютерної гри Dota 2.	120	5	54,2	1,36	6504
Розробка алгоритмів функціонування складових інформаційної технології аналітики комп'ютерної гри Dota 2.	240	5	54,2	1,36	13008
Розробка структури інформаційної технології аналітики комп'ютерної гри Dota 2.	120	4	50,6	1,27	6072
Реалізація програмного забезпечення	660	6	57,8	1,45	38148
Всього					71828

Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0,1 \cdot (Z_o + Z_p) = 0,1 \cdot (104220 + 71828) = 17605 \text{ грн,} \quad (4.4)$$

Нарахування на заробітну плату H_{zn} розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$\begin{aligned}
 H_{зп} &= \beta \cdot (З_о + З_p + З_д) = \\
 &= 0,22 \cdot (104220 + 71828 + 17605) = 42604 \text{ грн},
 \end{aligned}
 \tag{4.5}$$

де $З_о$ – основна заробітна плата розробників, грн.; $З_p$ – основна заробітна плата робітників, грн.; $З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Витрати на матеріали M , що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$M = \sum_1^n H_i \cdot Ц_i \cdot K_i, \tag{4.6}$$

де H_i – кількість матеріалів i -го виду, шт.; $Ц_i$ – ціна матеріалів i -го виду, грн.; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; n – кількість видів матеріалів.

Зроблені розрахунки зводимо до таблиці 4.4.

Таблиця 4.4 – Матеріали, що використані на розробку

Найменування матеріалів	Ціна за одиницю, грн.	Витрачено	Вартість витрачених комплектуючих, грн.
Папір	165	1	650
Флешка	300	1	300
Всього, з врахуванням коефіцієнта транспортних витрат			1045

До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10...12% від вартості програмного забезпечення. Балансову вартість програмного забезпечення розраховують за формулою:

$$V_{\text{прг}} = \sum_1^k C_{\text{іпрг}} \cdot C_{\text{прг.і}} \cdot K_i, \quad (4.7)$$

де $C_{\text{іпрг}}$ – ціна придбання програмного забезпечення і-го виду, грн.; $C_{\text{прг.і}}$ – кількість одиниць програмного забезпечення відповідного виду, шт.; K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного забезпечення, $K_i = (1, 1 \dots 1, 12)$; k – кількість видів програмного забезпечення.

Зроблені розрахунки зводимо до таблиці 4.5.

Таблиця 4.5 – Витрати на придбання програмного забезпечення

Найменування програмного забезпечення	Ціна за одиницю, грн.	Витрачено	Вартість програмного забезпечення, грн.
Microsoft Office	2000	1	2000
Microsoft Windows	1540	1	1540
Всього, з врахуванням коефіцієнта інсталяції та налагодження			3894

Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою:

$$A = \frac{C_6}{T_b} \cdot \frac{t}{12}, \quad (4.8)$$

де C_6 – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; t – термін використання основного фонду, місяці; T_b – термін корисного використання основного фонду, роки.

Зроблені розрахунки зводимо до таблиці 4.6.

Таблиця 4.6 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук	20000	5	1	333
Принтер	6500	5	1	108
Роутер	1000	3	1	28
Всього	469			

Витрати на силову електроенергію V_e , якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$V_e = \sum \frac{W_i \cdot t_i \cdot C_e \cdot K_{впi}}{ККД} = \frac{0,045 \cdot 1300 \cdot 7,5 \cdot 0,85}{0,87} = 429 \text{ грн.} \quad (4.9)$$

де W_i – встановлена потужність обладнання, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год.; C_e – вартість 1 кВт електроенергії, грн.; $K_{впi}$ – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

Зроблені розрахунки зводимо до таблиці 4.7.

Таблиця 4.7 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук	0,045	1300

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_B = (Z_o + Z_p) \cdot \frac{H_{IB}}{100\%} = (104220 + 71828) \cdot \frac{70}{100} = 123234 \text{ грн}, \quad (4.10)$$

де H_{IB} – норма нарахування за статтею «Інші витрати».

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$V_{HЗВ} = (Z_o + Z_p) \cdot \frac{H_{HЗВ}}{100\%} = (104220 + 71828) \cdot \frac{120}{100} = 211258 \text{ грн}, \quad (4.11)$$

де $H_{HЗВ}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$\begin{aligned} V_{\text{заг}} &= Z_o + Z_p + Z_{\text{дод}} + Z_n + M_B + V_{\text{прг}} + A_{\text{обл}} + V_e + I_B + V_{HЗВ} = 104220 + \\ &71828 + 17605 + 42604 + 1045 + 3894 + 469 + 429 + 123234 + 211258 = \\ &576586 \text{ грн}, \end{aligned} \quad (4.12)$$

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{В_{заг}}{\eta} = \frac{576586}{0,9} = 640651 \text{ грн}, \quad (4.13)$$

де η – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії впровадження, то $\eta=0,9$.

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати: ΔN – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; $Ц_0$ – вартість послуги у році до впровадження інформаційної системи; $\pm\Delta Ц_0$ – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta\Pi$ для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta\Pi = (\pm\Delta Ц_0 \cdot N + Ц_0 \cdot \Delta N_i)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.14)$$

де $\pm\Delta\Pi$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm\Delta\Pi_0$ може мати як додатне, так і від’ємне значення (від’ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни); N – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки; Π_0 – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році; Π_6 – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів; ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки); λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$; ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho = 0,2 \dots 0,5$; θ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\theta = 18\%$.

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\Delta\Pi = ((100000 - 40000) \cdot 2500 - (5000 - 2500) \cdot 40000) \cdot 0,8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 10995600 \text{ грн}, \quad (4.15)$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^t} = \frac{10995600}{(1+0,1)^1} = 9996000 \text{ грн}, \quad (4.16)$$

де $\Delta\Pi$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.; T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо $T=1$ рік); τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$; t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{инв}} \cdot ЗВ = 5 \cdot 640651 = 3203255 \text{ грн}, \quad (4.17)$$

де $k_{\text{инв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{инв}}=2 \dots 5$, але може бути і більшим; $ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = ПП - PV = 9996000 - 3203255 = 6792745 \text{ грн}, \quad (4.18)$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.; PV – теперішня вартість початкових інвестицій, грн.

Оскільки $E_{abc} > 0$, то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_B або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій E_B , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_B = T_{ж} \sqrt{1 + \frac{E_{abc}}{PV}} = \sqrt[1]{1 + \frac{6792745}{3203255}} = 1,76, \quad (4.19)$$

де $T_{ж}$ – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування $\tau_{\text{мін}}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{\text{мін}}$ визначається за формулою:

$$\tau_{\text{мін}} = d + f = 0,9 + 0,5 = 1,4, \quad (4.20)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,9 \dots 0,12$; f – показник, що характеризує

ризикованість вкладення інвестицій; зазвичай величина $f = 0,05 \dots 0,5$, але може бути і значно вищою.

Оскільки $E_b = 1,76 > \tau_{\min} = 1,4$, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_o = \frac{1}{E_b} = \frac{1}{1,76} = 0,57 \text{ року.} \quad (4.21)$$

Оскільки $T_o = 0,57 < 1 \dots 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

4.4 Висновок до розділу 4

У розділі було виконано розрахунок витрат на розробку та виготовлення нового технічного рішення, сума яких складає 640651 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, знайдено термін окупності витрат для виробника та економічний ефект для споживача при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розробка у виробництві та використанні дешевша за аналог і є висококонкурентоспроможною. Період окупності складе близько 0,57 року.

ВИСНОВКИ

Аналітика комп'ютерних ігор, зокрема Dota 2, має велику популярність для гравців, команд, тренерів та фанатів цієї популярної МОБА-гри. Тому, доцільною є розробка аналітики для самої гри. Вона надає можливість розуміти як влаштована гра, ефективно використовувати героїв та стратегії, а також покращувати особисті геймплейні навички. Гравці можуть аналізувати свої матчі, виявляти помилки, вдосконалювати тактику і прийоми гри, що допомагає досягати вищих результатів.

У розділі 1 було проведено аналіз сучасних методів та засобів для вирішення задачі аналітики комп'ютерної гри Dota 2, визначено особливості даних засобів, які можуть бути корисними, та їх недоліки, складено їх порівняльну характеристику. Виділено головні переваги функціоналу платформ, а також опрацьовано недоліки, які варто усунути. Сформульовано постановку задачі. Крім того, досліджено такі сучасні методи як машинне навчання та штучний інтелект, статистичний аналіз даних, візуалізація даних, аналіз стрімів та коментарів та соціальний аналіз, що використовуються в процесі аналітики комп'ютерної гри, та визначено метод статистичного аналізу як такий, що дозволяє отримати об'єктивні числові дані, здійснювати кількісні порівняння та приймати обґрунтовані рішення.

У розділі 2 було удосконалено математичну модель вибору комп'ютерного персонажа, яка на відміну від існуючих відрізняється введенням додаткових критеріїв аналітики, що дозволило розширити функціонал інформаційної технології. Використання статистичного аналізу при визначенні критеріїв дозволить більш точно врахувати складні взаємодії між різними факторами та забезпечити оптимальний вибір комп'ютерного персонажа.

Крім того, було розроблено схему удосконаленого алгоритму, який передбачає аналітику комп'ютерної гри Dota 2, а також алгоритми роботи складових інформаційної технології.

З метою реалізації запропонованого алгоритму аналітики комп'ютерної гри було розроблено структуру інформаційної технології, до якої входять такі модулі: модуль збору даних, модуль обробки даних, модуль статистичного аналізу, модуль виведення даних.

У розділі 3 запропоновано програмну реалізацію інформаційної технології з обґрунтуванням вибору середовища і мови програмування. В результаті обрано мову програмування JavaScript, як таку, що забезпечує кросплатформеність, зручність розробки, доступність ресурсів та продуктивність. Описано середовище виконання та середовище проектування інтерфейсу користувача.

З метою тестування інформаційної технології було проведено понад 500 досліджень, а саме було проаналізовано та рекомендовано різних персонажів та спорядження відповідно до 4 вищенаведених правил аналітики.

Сутність кожного експерименту полягала у виборі з потужності множини персонажів, спорядження та персонажа зі спорядженням як предмет аналітики та проведенні процесу аналізу для кожного з них понад 100 разів.

Таким чином в результаті тестування програмного засобу, можна зробити висновок про те, що було розширено функціонал додатку, а саме функціональні можливості вибору комп'ютерного персонажа та його спорядження за скорочений час завдяки автоматизації цього процесу.

У розділі 4 було виконано розрахунок витрат на розробку та виготовлення нового технічного рішення, сума яких складає 640651 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, знайдено термін окупності витрат для виробника та економічний ефект для споживача при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розробка у виробництві та використанні дешевша за аналог і є висококонкурентоспроможною. Період окупності складе близько 0,57 року.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Савчук Т. О., Тишко В. А. Розробка удосконаленого алгоритму аналітики комп'ютерної гри Dota 2 / Всеукраїнська науково-практична інтернет-конференція «LII Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації», ВНТУ Вінниця, 2023.

2. Савчук Т. О., Тишко В. А. Структура інформаційної технології аналітики комп'ютерної гри Dota 2 / XXVIII Міжнародна науково-практична конференція «Science and development of methods for solving modern problems», Мельбурн, Австралія, 2023.

3. Савчук Т. О. Комп'ютерна програма ««Інформаційна технологія аналітики комп'ютерної гри Dota 2» / Савчук Т. О., Тишко В. А. / Свідоцтво про реєстрацію авторського права на твір (АП) с202308054, Вх-47007/2023 від 28.11.

4. Game Analytics – [Електронний ресурс]. – Режим доступу: <https://www.indicative.com/resource/gaming-analytics/>.

5. Машинне навчання – [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Машинне_навчання (дата звернення: 20.06.2023).

6. Dota 2 – [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Dota_2.

7. Статистичний аналіз – [Електронний ресурс]. – Режим доступу: <https://www.simplilearn.com/what-is-statistical-analysis-article>.

8. Соціальний аналіз – [Електронний ресурс]. – Режим доступу: <https://buklib.net/books/34093>.

9. OpenDota – [Електронний ресурс]. – Режим доступу: <https://www.opendota.com/>.

10. DotaPlus – [Електронний ресурс]. – Режим доступу: <https://www.dota2.com/plus/>.

11. Strafe – [Електронний ресурс]. – Режим доступу: <https://www.strafe.com/>.

12. Dotabuff – [Электронный ресурс]. – Режим доступа:
<https://www.dotabuff.com/>

13. Overwolf – [Электронный ресурс]. – Режим доступа:
<https://www.overwolf.com/>.

14. JavaScript – [Электронный ресурс]. – Режим доступа:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>.

15. C# – [Электронный ресурс]. – Режим доступа:
https://uk.wikipedia.org/wiki/C_Sharp.

16. Microsoft.NET – [Электронный ресурс]. – Режим доступа:
<https://dotnet.microsoft.com/en-us/>.

17. JVM – [Электронный ресурс]. – Режим доступа:
https://en.wikipedia.org/wiki/Java_virtual_machine.

18. .NET Framework – [Электронный ресурс]. – Режим доступа:
<https://dotnet.microsoft.com/en-us/download/dotnet-framework>.

19. Java – [Электронный ресурс]. – Режим доступа:
<https://uk.wikipedia.org/wiki/Java>.

20. Python – [Электронный ресурс]. – Режим доступа:
<https://docs.python.org/3/>.

21. PHP – [Электронный ресурс]. – Режим доступа:
<https://www.php.net/manual/en/index.php>.

22. C++ – [Электронный ресурс]. – Режим доступа:
<https://learn.microsoft.com/en-us/cpp/?view=msvc-170>.

23. Visual Studio – [Электронный ресурс]. – Режим доступа:
<https://visualstudio.microsoft.com/>.

24. Web Storm – [Электронный ресурс]. – Режим доступа:
<https://www.jetbrains.com/webstorm/>.

25. Visual Studio Code – [Электронный ресурс]. – Режим доступа:
<https://code.visualstudio.com/>.

26. Figma – [Электронный ресурс]. – Режим доступа:
<https://uk.wikipedia.org/wiki/Figma>.

ДОДАТКИ

Додаток А (обов'язковий)

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: Інформаційна технологія аналітики комп'ютерної гри Dota 2

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФШТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 83,8% Схожість 16,2%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.


Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи  Тишко В.А.

Керівник роботи  Савчук Т.О.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку  Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

Модуль статистичного аналізу

```
const axios = require('axios');

// Ключ API для доступу до даних Dota 2
const apiKey = 'Ваш ключ API з https://www.opendota.com/api-keys';

// Функція для отримання даних про героїв
async function getHeroes() {
  try {
    const response = await
axios.get('https://api.steampowered.com/IEconDOTA2_570/GetHeroes/v0001/', {
      params: {
        key: apiKey
      }
    });
    return response.data;
  } catch (error) {
    console.error('Помилка при отриманні даних про героїв', error);
    return null;
  }
}

// Функція для отримання даних про спорядження
async function getItems() {
  try {
    const response = await
axios.get('https://api.steampowered.com/IEconDOTA2_570/GetGameItems/V001/', {
```

```
        params: {
            key: apiKey
        }
    });
    return response.data;
} catch (error) {
    console.error('Помилка при отриманні даних про спорядження', error);
    return null;
}
}
```

// Основна функція

```
async function main() {
    const heroData = await getHeroes();
    const itemsData = await getItems();

    if (heroData && itemsData) {
        const recommendation = recommendHeroAndItems(heroData, itemsData);
        console.log('Рекомендація для героя:', recommendation.hero);
        console.log('Рекомендація для спорядження:', recommendation.items);
    }
}

function generateRecommendations(numRecommendations) {
    const recommendations = [];

    for (let i = 0; i < numRecommendations; i++) {
        const heroData = await getHeroes();
        const itemsData = await getItems();

        if (heroData && itemsData) {
```



```
const recommendation = recommendHeroAndItems(heroData,
itemsData);
  recommendations.push({
    hero: recommendation.hero,
    items: recommendation.items
  });
}
}
return recommendations;
}
```

```
const recommendations = generateRecommendations(5);
```

```
for (let i = 0; i < recommendations.length; i++) {
  console.log(`Рекомендація #${i + 1}`);
  console.log('Рекомендація для героя:', recommendations[i].hero);
  console.log('Рекомендація для спорядження:', recommendations[i].items);
  console.log('\n');
}
```

```
async function generateRecommendationPDF() {
  const doc = new PDFDocument();
  doc.pipe(fs.createWriteStream('recommendations.pdf'));
}
```

```
for (let page = 1; page <= 5; page++) {
  doc.addPage({
    size: 'A4',
    layout: 'portrait',
    margins: {
      top: 50,
      bottom: 50,
```

```

        left: 50,
        right: 50
    }
});

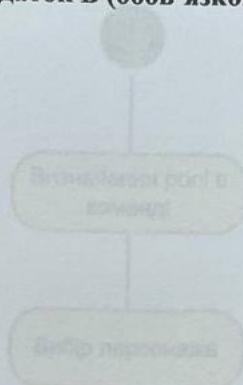
const heroData = await getHeroes();
const itemsData = await getItems();

if (heroData && itemsData) {
    const recommendation = recommendHeroAndItems(heroData,
itemsData);
    doc.fontSize(16).text(`Сторінка ${page}`, { align: 'center' });
    doc.moveDown();
    doc.fontSize(14).text('Рекомендація для героя:', { align: 'left' });
    doc.text(recommendation.hero, { align: 'left' });
    doc.moveDown();
    doc.fontSize(14).text('Рекомендація для спорядження:', { align: 'left'
});
    doc.text(recommendation.items, { align: 'left' });

    if (page < 5) {
        doc.addPage();
    }
}
}
doc.end();
}
main();

```


Додаток В (обов'язковий)




ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІТИКИ КОМП'ЮТЕРНОЇ
ГРИ DOTA 2Рисунок В.1 – UML-діаграма узагальненого алгоритму аналітики комп'ютерної
гри Dota 2

Виконав: студент 2-го курсу,
групи 1КН-22м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)

 Тишко В. А.
(прізвище та ініціали)

Керівник: PhD, проф. каф. КН

 Савчук Т. О.
(прізвище та ініціали)

« 07 » 12 2023 р.

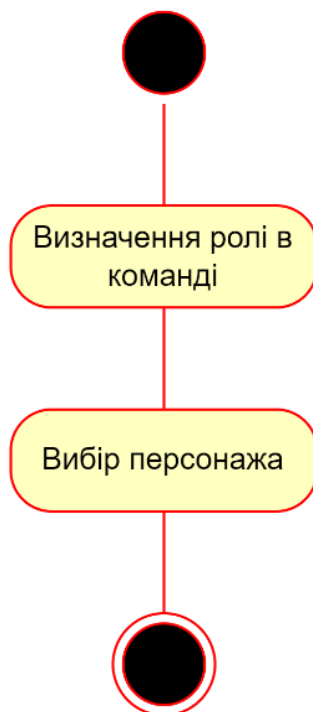


Рисунок В.1 – UML-діаграма узагальненого алгоритму аналітики комп'ютерної гри Dota 2



Рисунок В.2 – UML-діаграма удосконаленого алгоритму аналітики комп'ютерної гри Dota 2



Рисунок В.3 – Структура інформаційної технології аналітики комп'ютерної гри Dota 2.



Рисунок В.4 – UML-діаграма роботи модуля збору даних



Рисунок В.5 – UML-діаграма роботи модуля обробки даних

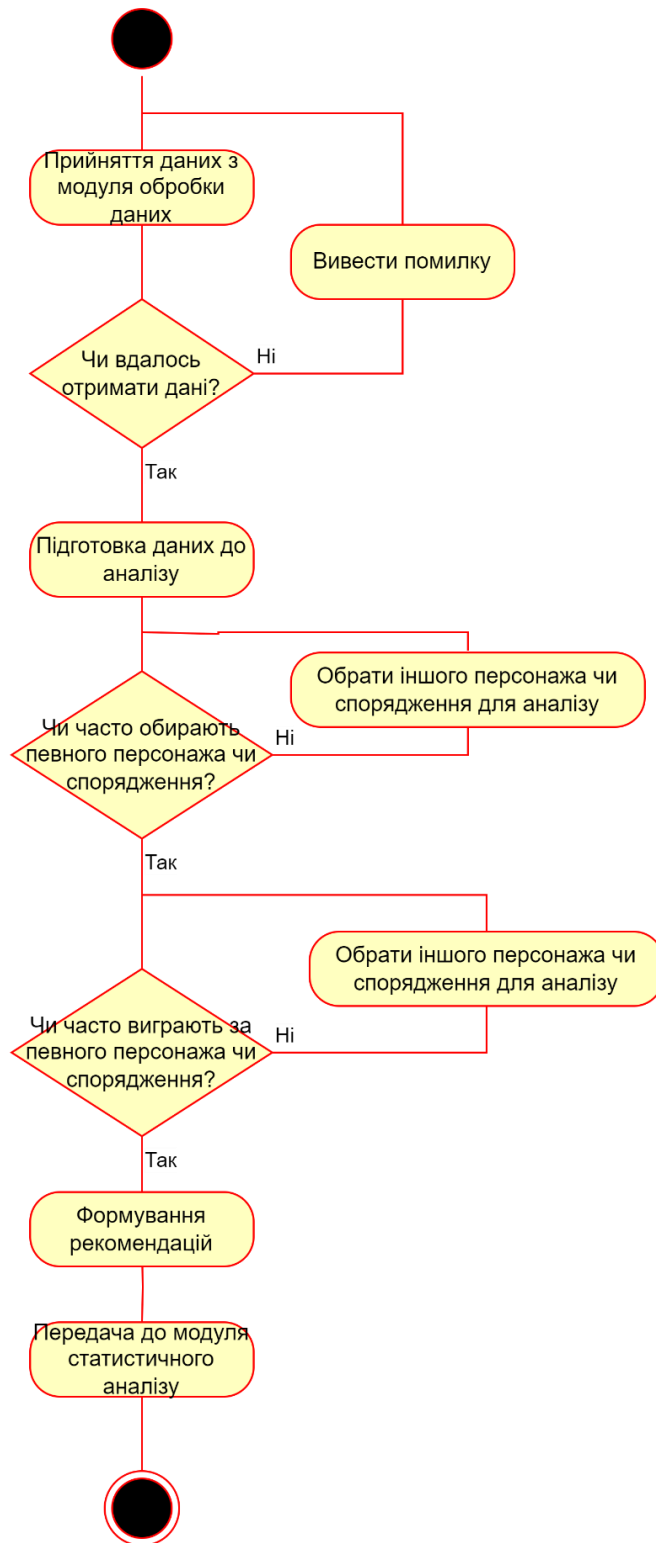


Рисунок В.6 – UML-діаграма роботи модуля статистичного аналізу

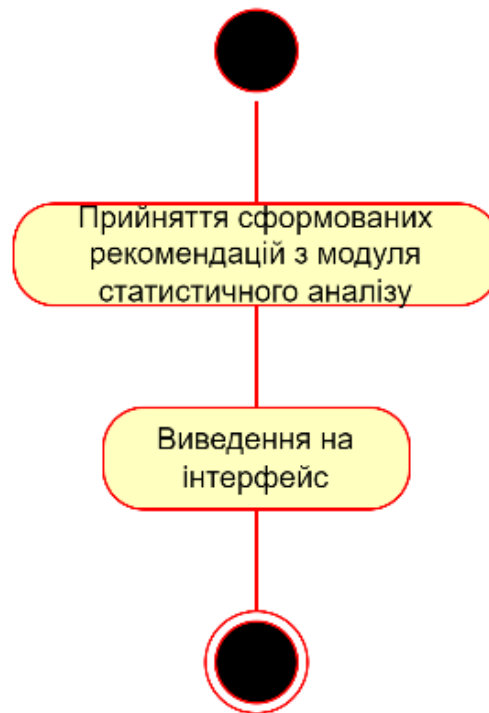


Рисунок В.7 – UML-діаграма роботи модуля виведення даних

Додаток Г (довідниковий)

Інструкція користувача

Крок 1. Відкрити інформаційну технологію аналітики комп'ютерної гри Dota 2 на рисунку Б.1.

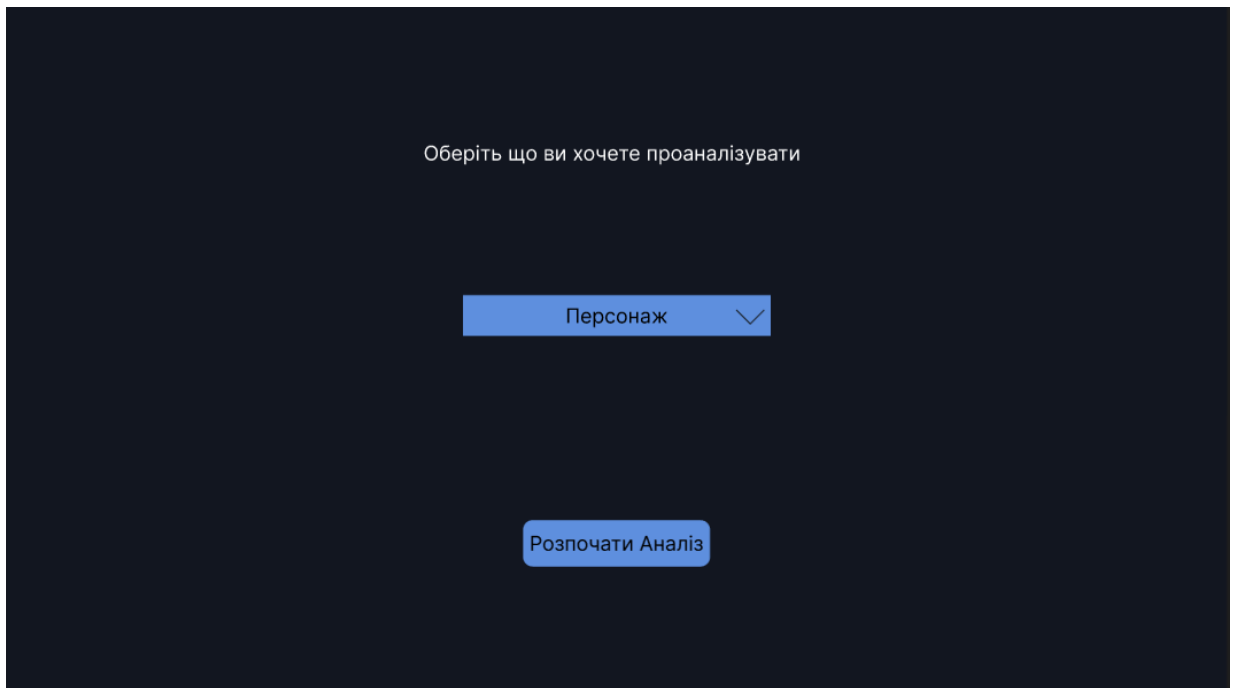


Рисунок Б.1 – Головна сторінка

Крок 2. На головній сторінці користувач може вибрати, що він хоче проаналізувати натиснувши на кнопку спадаючого меню на рисунку Б.2.

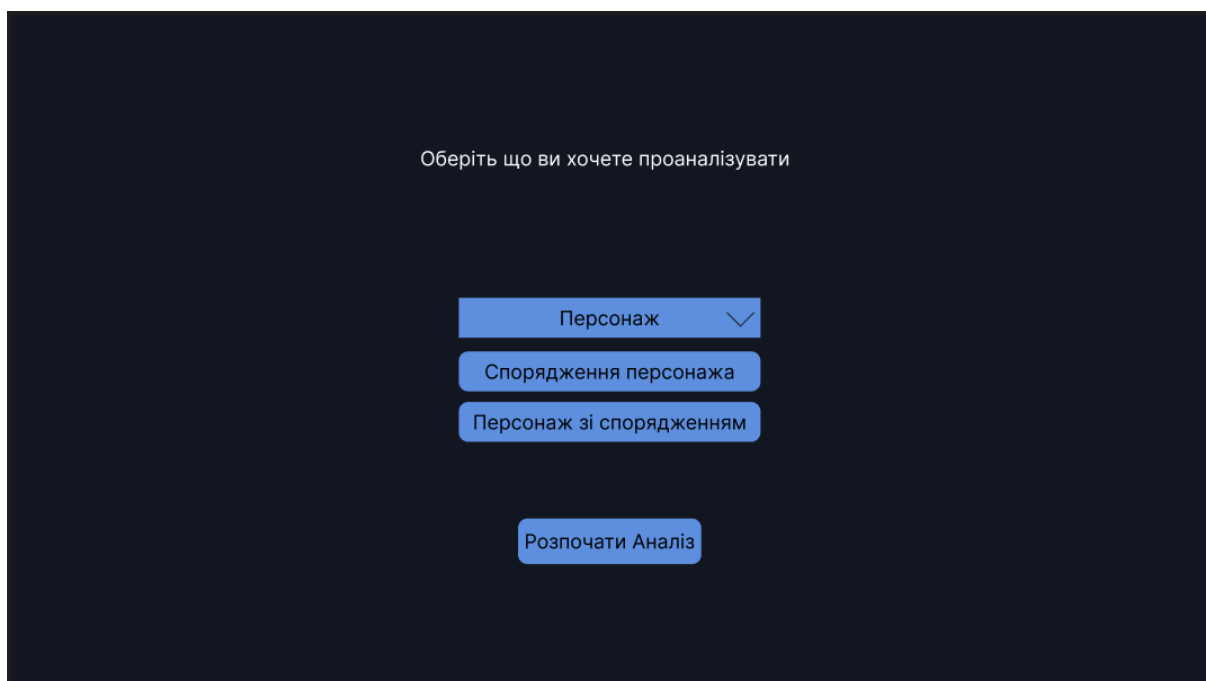


Рисунок Б.2 – Вибір об'єкта аналізу

Крок 3. Після вибору об'єкта аналізу, треба натиснути на кнопку «Розпочати Аналіз». Після цього користувачеві виведеться результат аналітики з рекомендаціями на рисунках Б.3-Б.5.

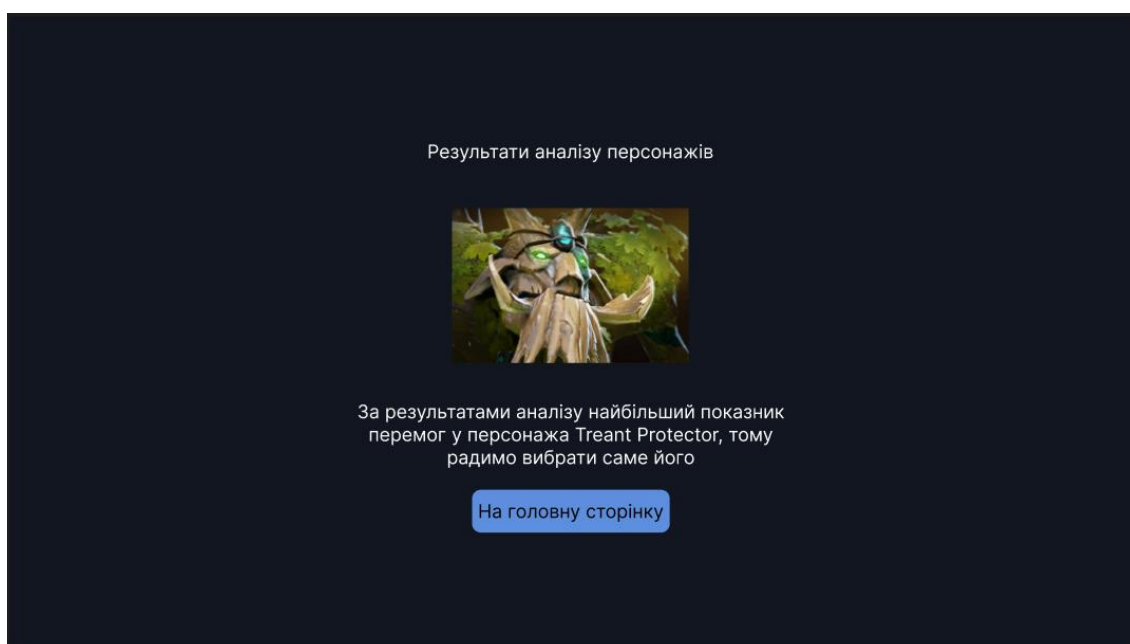


Рисунок Б.3 – Результат аналізу персонажів

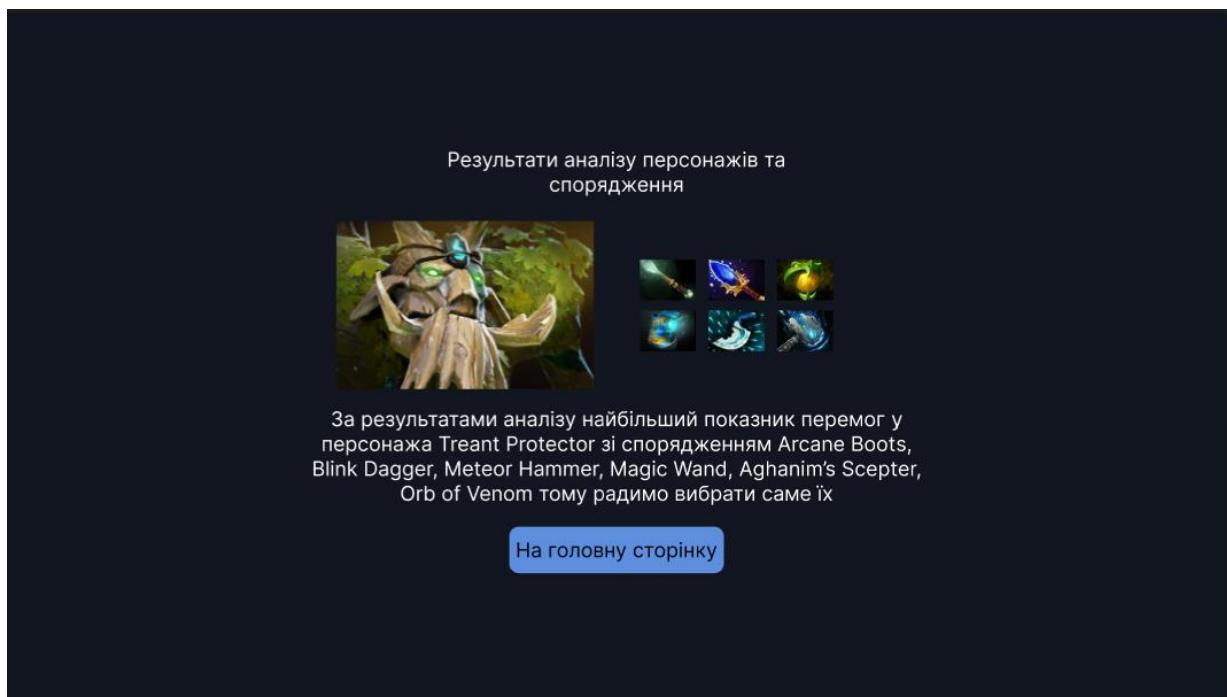


Рисунок Б.4 – Результат аналізу персонажів та спорядження

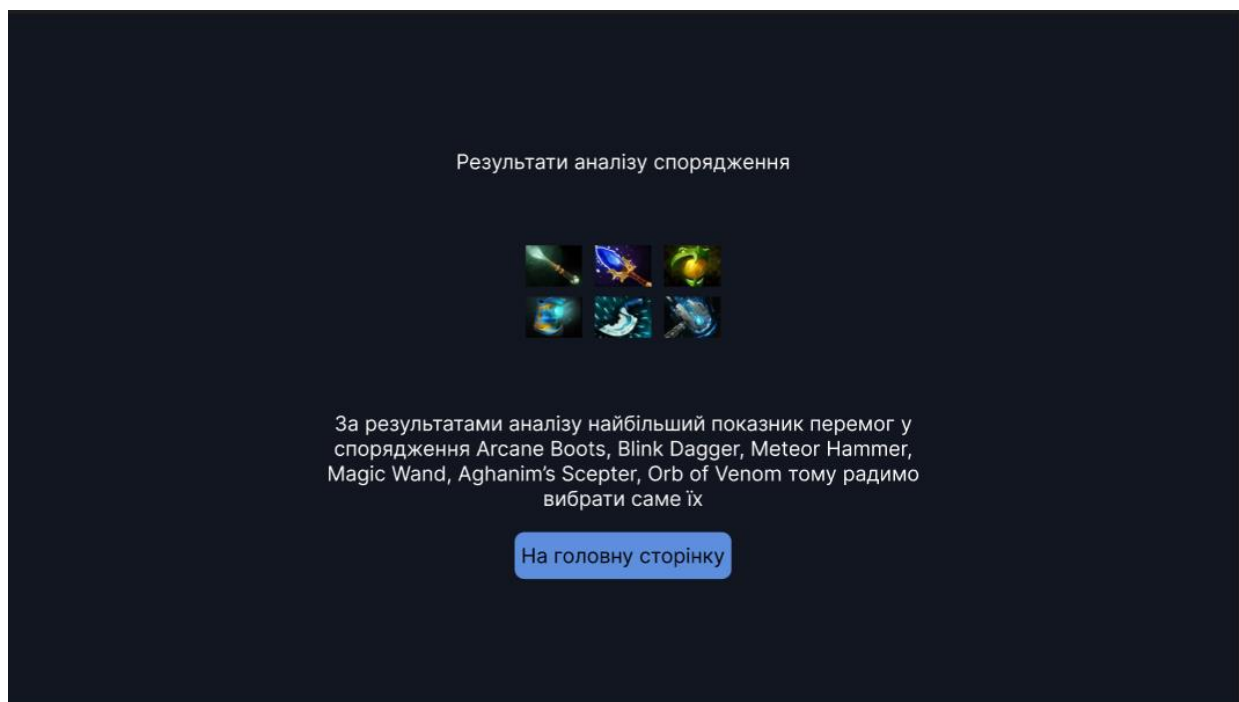


Рисунок Б.5 – Результат аналізу спорядження