

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Пояснювальна записка

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему: Розробка методів та програмних засобів для підвищення ефективності проведення форумних рольових ігор

Виконала: студентка II курсу

групи 1ПІ-18 м

спеціальності

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Лапко М. С.

(прізвище та ініціали)

Керівник: к.т.н., доц. Романюк О. В.

(прізвище та ініціали)

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Освітньо-кваліфікаційний рівень – магістр
Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Романюк О. Н.
“ ____ ” _____ 2019 року

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ**

Лапко Маргариті Сергіївні

1. Тема роботи – розробка методів та програмних засобів для підвищення ефективності проведення форумних рольових ігор.

Керівник роботи: Романюк Оксана Володимирівна, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від “__”_____2019 року №____

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи: тип рольової гри – форумна (текстова), моніторинговий період – 30 днів, метод аналізу статистичних даних – матричний, максимальна кількість гравців форуму – 50 чоловік, метрики активності – якісні та кількісні, кількість метрик активності – 15, кількість експертів для визначення ваг метрик – 10.

4. Зміст розрахунково-пояснювальної записки: вступ; аналіз процесу ведення форумної рольової гри та порівняльна характеристика методів збору

ігрової статистики; розробка методу та засобів ведення форумної рольової гри; програмна реалізація аналітичного модуля за новим методом; тестування розробленого модуля; економічна частина; висновки; список використаних джерел; додатки.

5. Перелік графічного матеріалу актуальність дослідження; мета, об'єкт та предмет дослідження; порівняльна характеристика підходів до ведення рольової статистики; експертний аналіз та визначення ваг для вибору метрик; розробка алгоритму обчислення рейтингу гравців; формування таблиці рейтингу; остаточний вигляд вікна статистики; тестування системи; висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Романюк О. В., к.т.н., доцент кафедри ПЗ		
5	Бальзан М. В., к.е.н., доцент кафедри ЕПВМ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної галузі та порівняння аналогів	10.09.2019 – 20.09.2019	Виконано
2	Обґрунтування вибору методу збору статистики та принципи їх обчислення	21.09.2019 – 8.10.2019	Виконано
4	Розробка розширеної системи метрик активності гравців та удосконаленого методу обчислення рейтингу активності	09.10.2019 – 15.10.2019	Виконано
5	Розробка програмної реалізації та тестування	16.10.2019 – 20.11.2019	Виконано
6	Економічна частина	21.11.2019 – 04.12.2019	Виконано

Студент _____

(підпис)

Лапко М.С.
(прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи

(підпис)

Романюк О.В.

(прізвище та ініціали)

АНОТАЦІЯ

Магістерську кваліфікаційну роботу «Розробка методів та програмних засобів для підвищення ефективності проведення форумних рольових ігор» присвячено розробці нового методу обчислення рейтингу активності гравців для ведення звітності їх діяльності, його програмній реалізації та впровадженню у галузь рольової гри. У ході виконання проаналізовано сучасні методи збору статистичних методів та способи ведення рейтингу гравців у інших підвидах рольової гри. Обґрунтовано вибір метрик за допомогою соціологічного дослідження та експертної оцінки досвідчених гравців, на основі матричного методу був створений новий метод для зведення якісних та кількісних показників статистики.

Програмна реалізація виконана у вигляді незалежного аналітичного модуля на мові JavaScript, використовуючи бібліотеки для побудови графіків chart.js та D3, що дає можливість використати його на будь-якому рольовому форумі для аналізу.

ABSTRACT

Master's qualification work "Improving the effectiveness of forum role-playing game by development of methods and software realization" is devoted to the development of a new method of calculating the rating of players' activity for reporting their activities, its program realization and implementation in the field of role-playing. In the course of the implementation, modern methods of collecting statistical methods and ways of rating players in other sub-types of role-playing are analyzed. The choice of metrics is grounded by means of sociological research and expert evaluation by experienced players, and based on the matrix method, a new method was created for summarizing qualitative and quantitative indicators of statistics.

The software implementation is made in the form of an independent analytical module in JavaScript, using libraries Chart.js and D3, which makes it possible to use it in any role forum for analysis.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРОЦЕСУ ВЕДЕННЯ ФОРУМНОЇ РОЛЬОВОЇ ГРИ ТА ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА МЕТОДІВ ЗБОРУ ІГРОВОЇ СТАТИСТИКИ	13
1.1 Аналіз проблематики під час проведення форумної рольової гри	13
1.2 Варіантний аналіз підходів до ведення ігрової статистики.....	18
1.3 Аналіз та обґрунтування вибору одного з сучасних статистичних методів...	24
1.4 Висновок	27
2 РОЗРОБКА МЕТОДУ ТА ЗАСОБІВ ВЕДЕННЯ ФОРУМНОЇ РОЛЬОВОЇ ГРИ	28
2.1 Вибір та визначення ваг метрик активності гравців	28
2.2 Аналіз існуючих методів обрахунку рейтингу гравців.....	34
2.3 Розробка нового методу визначення активності гравця	37
2.4 Розробка алгоритму розрахунку активності гравця за запропонованим методом	44
2.5 Висновок	45
3 ПРОГРАМНА РЕАЛІЗАЦІЯ АНАЛІТИЧНОГО МОДУЛЯ ЗА НОВИМ МЕТОДОМ.....	47
3.1 Аналіз та обґрунтування вибору засобів реалізації методу.....	47
3.2 Розробка архітектури та алгоритму роботи аналітичного модуля	54
3.3 Розробка інтерфейсу аналітичного модуля, структура та опис основних елементів форуму для проведення гри	57
3.4 Розробка та опис основних графіків.....	62

	7
3.5 Висновок	69
4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО МОДУЛЯ.....	71
4.1 Обґрунтування вибору методики тестування системи	71
4.2 Проведення тестування	73
4.3 Висновки	77
5 ЕКОНОМІЧНА ЧАСТИНА.....	78
5.1 Оцінювання комерційного потенціалу розроблюваного модуля.....	78
5.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.	81
5.3 Прогнозування комерційних ефектів від реалізації результатів розробки	85
5.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.....	87
5.5 Висновки	91
ВИСНОВКИ.....	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	94
ДОДАТКИ	97
Додаток А. Технічне завдання.....	98
Додаток Б. Лістинг коду	102
Додаток В. Алгоритм роботи модуля	107
Додаток Г. Ілюстративний матеріал.....	108

ВСТУП

Обґрунтування вибору теми дослідження. Рольові ігри [1] складають один з основних жанрів ігор та існують на всіх цифрових платформах, а також поза ними, формуючи технологічний домен з великою кількістю форматів. Одним з таких форматів є текстова рольова гра. У порівнянні з іншими ігровими форматами, цей тип характеризується низкою деяких специфічних тільки для нього проблем, які потребують вирішення, особливо зі стрімким розвитком галузі.

Відсутність інструментів та чіткого теоретичного підґрунтя для проведення форумної гри – одна з таких проблем. Формат має найменший набір чітко визначених правил, що є як і причиною його популярності серед широкого вікового контингенту, так і недоліком, враховуючи те, що без правил та стандартів немає ні порядку, ні фіксованої для усіх системи, яка забезпечувала б компроміс у веденні гри між гравцями з різних спільнот.

На початку розвитку галузі, модератори слідкували за порядком та діяльністю гравців на форумах, оскільки у межах однієї спільноти грало не більше десяти гравців. Зі зростанням популярності рольової гри середня кількість гравців на один форум збільшилась до декількох сотень, що дещо позбавило модераторів мобільності та знизило ефективність вирішення малопомітних індивідуальних для окремих гравців проблем та конфліктів, приділяючи увагу лише масштабним аспектам. Слабкий індивідуальний підхід зі сторони адміністрації поступово зменшує рівень зацікавленості у спільноти гравцем, активність падає, а форум занепадає або накопичує порушників правил, заважаючи гравцям, які залишились, тому що модератори – теж люди, що не здатні бачити усе.

Щоб підвищити ефективність роботи модераторів форумів та рівень зацікавленості гравців форумних ігор, доцільно розробити універсальну систему метрик активності гравців, автоматизувати контроль над гравцями шляхом розробки нового методу розрахунку інтегрального показника їх активності.

Нерідко аналіз діяльності гравців завершується їх рейтинговою таблицею та графіками активності, що дозволяє підвищити інтерес до форуму завдяки спортивній конкуренції.

Оскільки жоден форум для проведення рольових ігор не забезпечує механізм побудови рейтингу та обрахунок показників активності (які навіть не є фіксованими для усіх спільнот), тоді як у свою чергу близько 70-80% усіх веб-сайтів мають аналітичну складову [2], галузь потребує змін, щоб залишатися конкурентоспроможною.

Кожен з розробників використовує свої критерії, згідно яких обраховується рейтинг (позитивні та негативні оцінки інших гравців до створеного контенту, кількість коментарів, порушень), свої формули та принципи. Варто зазначити, що велика кількість формул обрахунку даних є неточною з точки зору математичної статистики і логіки. Навіть на таких популярних веб-сайтах, як Amazon та Urban Dictionary [3] рейтинг користувачів обраховується некоректно, що псує враження про ресурс.

Найчастіше, при обрахунку рейтингу, враховуються лише позитивні та негативні оцінки відвідувачів, тобто лише якісні абсолютні показники. Ця стратегія достатньо обмежена та неточна, оскільки розробники використовують примітивні методи обчислення, не хвилюючись про те, що кінцевий результат може суперечити дійсності. Наприклад на сайті Urban dictionary рейтинг обраховується різницею позитивних та негативних оцінок, а на Amazon – часткою позитивних оцінок до усіх отриманих [4]. Обидва метода враховують лише абсолютні значення, і якщо проекспериментувати, виявиться, що обчислення розмістить користувача з меншим відсотком позитивних оцінок вище у рейтингу, ніж того користувача, у якого відсоток більший, що не є справедливим, так як метод не враховує жодних допоміжних метрик, щоб це виправдати математично.

Крім того, що безмежна кількість методів для обчислення рейтингу в інтернеті не відповідає очікуваним стандартам користувача, жоден з існуючих

методів не забезпечує використання потрібного переліку метрик для покриття проблем при проведенні форумної рольової гри, саме тому актуальною задачею є розробка нового методу, що використовуватиме як кількісні, так і якісні метрики (вибір яких теж буде запропоновано та обґрунтовано за допомогою експертної оцінки) під час ведення рольової гри та допоможе у вирішенні проблеми відсутності аналітичної складової у галузі та в цілому підвищити ефективність проведення форумної рольової гри.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась відповідно до плану науково-дослідних робіт кафедри програмного забезпечення.

Мета та завдання дослідження. Метою роботи є підвищення ефективності проведення форумної рольової гри за рахунок розробки нового методу автоматизованого розрахунку інтегрального показника активності гравців на основі розширеного діапазону як кількісних, так і якісних метрик активності.

Основними задачами дослідження є:

- визначити основні найбільш важливі проблеми під час проведення форумної рольової гри для їх вирішення;
- провести аналіз існуючих методів обчислення рейтингу гравця для визначення напрямку підвищення точності його розрахунку;
- розробити розширену систему метрик активності гравців шляхом експертного відбору найважливіших параметрів;
- розробити новий метод обчислення рейтингу активності гравців;
- розробити програмні компоненти та систему розрахунку рейтингу гравців на основі запропонованого методу;
- обґрунтувати метод тестування та провести дослідження розроблених засобів.

Об'єкт дослідження – процес визначення активності гравців форумних рольових ігор.

Предмет дослідження – методи та засоби визначення активності гравців форумних рольових ігор.

Методи дослідження. У процесі дослідження використовувались: теорія чисел, математичної статистики та аналізу, методи збору статистичних даних для обчислення рейтингу, лінійна алгебра, теорія інформаційного дизайну, теорія системного аналізу, існуючі методи обчислення рейтингу користувачів на веб-сайтах, зокрема матричний метод статистичного аналізу, комп'ютерне моделювання для перевірки та аналізу теоретичних положень.

Наукова новизна отриманих результатів.

1. Подальшого розвитку отримав метод обчислення рейтингу активності гравця на основі матричного методу, який на відміну від існуючого використовує розширений діапазон як кількісних, так і якісних показників активності, що дозволило підвищити ефективність роботи модераторів та вилучити людський фактор при вирішенні характерних проблем при проведенні рольової гри.

2. Вперше запропоновано загальну систему показників для визначення рейтингу активності гравців форумних рольових ігор, особливість якої полягає у її універсальності та узгодженій нормативності для міжнародних рольових спільнот.

Практична цінність отриманих результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби для підвищення ефективності проведення форумних рольових ігор.

Особистий внесок здобувача. Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто. У друкованих працях, опублікованих у співавторстві, автору належать такі результати:

- проведення дослідження з обґрунтування та створення універсальної системи метрик активності гравців для форумних рольових ігор [5];

- розробка методу та алгоритму обчислення рейтингу активності гравців на основі запропонованої розширеної системи якісних та кількісних показників [6].

Апробація матеріалів магістерської кваліфікаційної роботи. Результати роботи доповідалися у межах XII Міжнародної науково-практичної конференції «Інформаційні технології і автоматизація – 2019» Одеської національної академії харчових технологій та Міжнародна науково-практична конференція «Електронні інформаційні ресурси в освіті і науці: створення, використання, доступ» (Вінниця, 2019).

Публікації. Основні результати досліджень опубліковано в 2 наукових працях, у тому числі 2 – у матеріалах конференцій.

Структура та обсяг роботи. Магістерська кваліфікаційна робота складається зі вступу, п'яти розділів, висновків, списку використаних джерел, що містить 30 найменувань, 4 додатків. Робота містить 33 ілюстрації, 12 таблиць.

1 АНАЛІЗ ПРОЦЕСУ ВЕДЕННЯ ФОРУМНОЇ РОЛЬОВОЇ ГРИ ТА ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА МЕТОДІВ ЗБОРУ ІГРОВОЇ СТАТИСТИКИ

1.1 Аналіз проблематики під час проведення форумної рольової гри

Текстова (форумна) рольова гра онлайн (анг.: text role-playing або play-by-post role-playing game [7], – гра постами; далі – TRPG) – це рольова гра, в якій гравці взаємодіють один з одним в заздалегідь визначеному середовищі за допомогою тексту. Це підмножина онлайн-рольової спільноти, яка обслуговує як гравців, так і творчих письменників. TRPG можуть базуватися на інших рольових іграх, неоригінальних сюжетах, включаючи книги, телебачення та фільми, та на власностворених світах. Ця діяльність тісно пов'язана як з інтерактивною художньою літературою, так і зі спільним написанням твору.

TRPG вивчаються освітніми установами, як інструмент, що сприяє покращенню навчального процесу. Як результат, наприклад, мова дітей, що займаються літературною творчістю так само більш лаконічна, краще структурована і логічна, ніж в тих, хто нею не займається. Їм простіше будувати діалоги, адже частково все це вже було програно в думках і на папері, а пишуться зазвичай саме актуальні ситуації. Стейнкуелер [8] стверджував, що TRPG – це:

- вивчення будь-якого середовища природнім чином;
- вирішення спільних проблем, що впливає на розвиток цифрової та обчислювальної грамотності;
- пришвидшення розвитку когнітивних систем;
- вдосконалення мовних процесів та навичок взаємодії.

Не дивлячись на те, що як суцільне явище TRPG має більше переваг, ніж недоліків, стан речей у окремих спільнотах набагато гірше, ніж здається на перший погляд. Існує безліч підвидів рольових ігор, і кожному характерна своя проблематика, та тільки у TRPG вони накопичуються прямопропорційно стрімкому росту галузі, але не вирішуються ініціаторами через те, що форумна гра

– це достатньо неформальний вид літературної діяльності, головним аспектом якої є, першочергово, розвага.

На рисунку 1.1 зображено схемувидів проблематики, що виникає під час проведення форумної рольової гри.

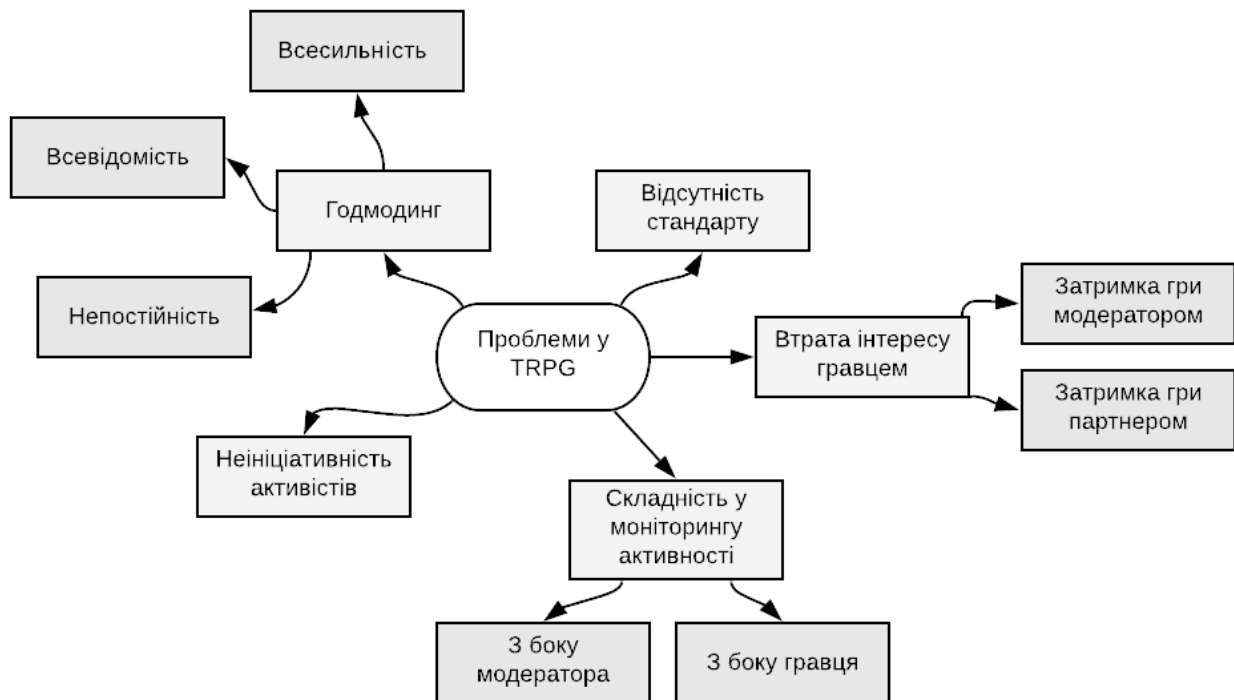


Рисунок 1.1 – Запропонована схема видів проблем під час проведення форумної рольової гри

За результатом огляду сучасного стану галузі, головними існуючими проблемами, що потребують негайного вирішення у процесі TRPG, є:

- годмодинг;
- відсутність стандарту;
- втрата інтересу гравцем;
- складність у моніторингу активності;
- не ініціативність активістів.

Годмод або годмодер (англ. «god mod» – режим бога) – це людина, представник драгівливого стилю поведінки гравців у TRPG. Існує ще один термін – Мері(Марті) Сью – тип персонажів, наділених гіпертрофованими перевагами, везінням і здібностями. Адміністрація рольових форумів зазвичай слідкує за їх появою, так як їх існування вважається чітінгом (махлюванням) у грі, але модератори фізично не здатні услідкувати за усіма, особливо коли на форумі більше п'ятдесяти гравців.

Є декілька шляхів ідентифікації годмодерства штучним шляхом, без модератора, але вони не є стовідсотковою гарантією успіху. Наприклад, існує спостереження, що зазвичай представники годмодингу пишуть занадто мало, та це не завжди є критерієм.

Годмодерство поділяється на:

- Всесильність (анг. «powerplay» – гра силою) – заподіяння шкоди іншому персонажу без дозволу гравця цього персонажа.
- Всевідомість (анг. «metagaming» – метагра) – персонаж гравця якимось чином знає ті деталі сюжету, які знати не міг.
- Непостійність (анг. «retconning» – перепланування) – ігнорування фактів у минулому, що впливають на теперішній та майбутній стан персонажа.

Відсутність стандарту.

У TRPG фігурує багато кількісних та якісних показників, на які опираються як гравці, так і модератори, щоб оцінити рівень своєї гри, і у 70% форумних ігор вони різні. Неформальність процесу є великим недоліком, оскільки зміна рівню навантаження впливає на зацікавленість – гравець, який писав більше 1000 слів за раз, відчує паузу у розвитку на форумі, де більше 500 слів за раз писати заборонено.

Ще одним аспектом відсутності стандартів є те, що якщо гравець потрапляє у нетипову ігрову ситуацію і йому невідомо, як діяти далі, відсутність стандартних

алгоритмів дій змушує його звернутися до модератора, що збільшує його кількість обов'язків.

Втрата інтересу гравцями відбувається, коли:

- гравець затримує гру своєю відсутністю з будь-якої причини (недієздатність, лень, зайнятість);
- модератор покладає менше зусиль у розвитку сюжетної лінії рольової гри, не з'являється на форумі, не йде на контакт з гравцями.

Складність у моніторингу під час проведення TRPG також може поділятися на проблеми гравця та модератора.

Проблеми з боку гравців: труднощі у веденні звітності боргів; ручне обрахування кількості постів за моніторинговий період, щоб встановити власну або чужу динаміку активності для порівняння; повторювальне порушення правил, якщо в перший раз воно ніким не було зафіксовано та виправлено.

Проблеми моніторингу з боку модератора: складність у організації дедлайнів гравців; недостатній рівень спостереження за гравцями у одинарних тривіальних випадках порушення правил; відсутність статистики активності – відсутність оцінювання діяльності гравця для його справедливого заохочення або покарання.

На не ініціативність активістів, більшим чином, впливає переконання про те, що для неформальної творчої літературної діяльності не потребує ані автоматизації процесу моніторингу, ані звітності, ні фіксованих для усієї галузі стандартів та норм.

Гравці старої школи вважають, що відсутність суворих правил є перевагою TRPG, своєрідною відзнакою, порівняно з іншими типами рольової гри. Вони розуміють, що поліпшення моніторингу, ведення статистики та обчислення рейтингу покращило б якість проведення гри та підвищило б вмотивованість гравців, але здатні це втілити розробники схильються до способу ведення рольової

гри, яким він був на початку виникнення у 2000-х, вважаючи, що проектування такої інновації не варта їх часу, якщо гравців і так наче все влаштовує.

На базі аналізу проблем створимо таблицю (табл. 1.1) для обґрунтування стану проблематики під час гри, дізнатися які проблеми вдасться вирішити шляхом розробки нового методу, а що зняти з обов'язків модератора.

Таблиця 1.1

Результат аналізу проблематики під час проведення TRPG

№	Проблема	Можливість автоматизації	Шлях вирішення
1	Годмодинг	0.5	Спроба знаходження годмодів за критерієм недостатньої кількості слів в пості. У інших випадках потрібне втручання модератора
1.1	Всесильність	0	
1.2	Всевідомість	0	
1.3	Непостійність	0	
2	Відсутність стандарту	1	Визначення універсальної системи метрик для оцінювання активності гравців
3	Втрата інтересу гравців: - затримка гри модератором;	0.5	Розробка системи аналізу та сповіщень для уникнення затримок
	- затримка гри партнером.	0.5	
4	Складність моніторингу - з боку гравця;	1	Програмна автоматизація моніторингу діяльності гравців і модераторів з публічним доступом
	- з боку модератора.	1	
5	Не ініціативність активістів	1	Демонстрація на прикладі, що автоматизація ведення TRPG є важливим фактором
		Усього:	5,5/10

Згідно таблиці 1.1 аналітично доведено, що якщо оцінити кожен проблему значенням 1 чи 0 (коли 1 – можливість вирішення проблеми шляхом впровадження автоматизації, створення та обрахунку метрик штучною системою, 0 – обов'язкове втручання людського фактору у особі модератора, 0.5 – часткове втручання модератора), 55% усіх проблем можливо вирішити технологічно та зняти з модераторів робоче навантаження.

Це надає їм можливість слідкувати за гомодерством, створювати контент для форуму та нагороджувати активних гравців згідно рейтингу, що збільшить зацікавленість у грі, а отже підвищить ефективність ведення TRPG.

1.2 Варіантний аналіз підходів до ведення ігрової статистики

Статистика в рольових іграх – це дані, які представляють певний аспект вигаданого персонажа. Ця частина даних зазвичай є цілим числом (без одиниць вимірювання). Розглянемо декілька прикладів із різних типів рольової гри, щоб об'єднати їх у рішення для TRPG.

Існуючі види елементів статистики у комп'ютерній рольовій грі зображена на рисунку 1.2.

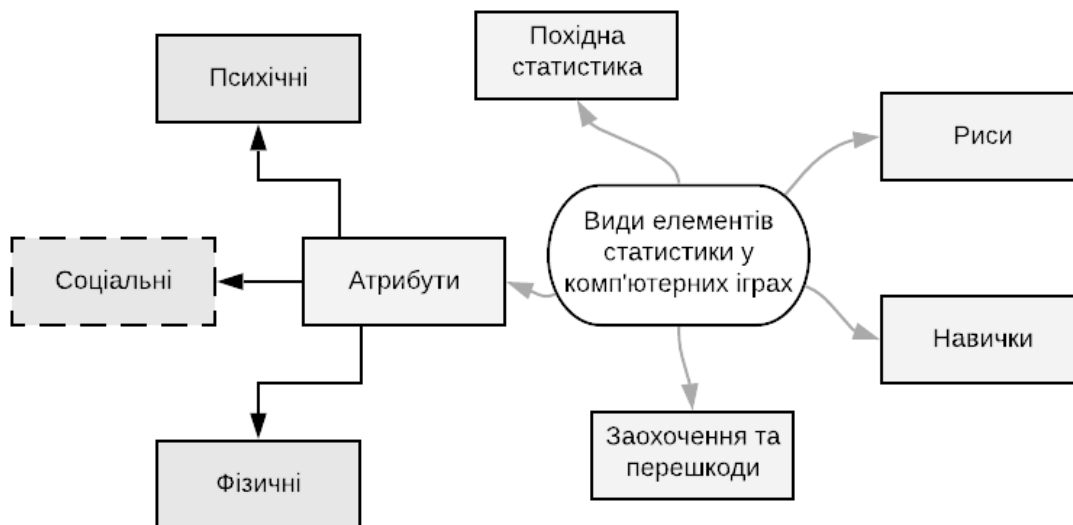


Рисунок 1.2 – Види елементів статистики персонажа у комп'ютерній рольовій грі

Перший підхід ведення статистики – сторінка персонажа із комп'ютерної гри Vampire: The Masquerade – Bloodlines [9]. Статистика персонажа у грі зображена на рисунку 1.3.



Рисунок 1.3 – Вигляд статистики у комп'ютерній рольовій грі

Більшість ігор поділяють свою статистику на кілька категорій. Набір категорій, які фактично використовуються в ігровій системі, а також точна статистика в межах кожної категорії сильно різняться. До найбільш часто використовуваних елементів статистики у комп'ютерній грі належать:

1. Атрибути.

Вони описують, якою мірою персонаж володіє природними, природженими характеристиками, загальними для всіх персонажів. Атрибути називають також характеристиками або здібностями. Більшість RPG використовують атрибути для опису фізичних та психічних характеристик персонажів, наприклад, їх силу чи мудрість.

Під час створення персонажів значення атрибутів зазвичай визначаються або рандомом, або шляхом розподілу фіксованої кількості балів по бажаним атрибутам.

2. Заохочення та перешкоди.

Заохочення – фізичне, соціальне, інтелектуальне чи інше індивідуальне вдосконалення персонажа. На противагу цьому існує перешкода – показник, що заважає персонажу виконати якусь дію протягом гри через його нестачу.

3. Навички.

Навичка представляє собою засвоєні знання та здібності персонажа протягом гри. Навички відомі різними іменами, включаючи вміння, здібності, сили, таланти та вміння. Під час створення персонажів навички персонажа гравця зазвичай вибираються з довгого списку. Персонаж може мати фіксовану кількість стартових навичок, або вони можуть платити за використання символічних точок.

4. Риси.

Риса являє собою широкий спектр знань персонажа. Риси виводяться із задалегідь визначеного списку.

5. Похідна статистика.

У багатьох іграх використовується похідна статистика: статистика, значення якої визначаються лише значеннями інших, "базових" статистичних даних.

Недоліками представлення статистики у рольовій грі є:

- використання лише кількісних значень без зазначення величини;
- надлишковість.

Зазвичай, усі характеристики персонажа, які гравець хоче вказати, він вказує при заповненні анкети («обличчя» персонажа у TRPG) на цього персонажа, фіксуючи у ній найдрібніші деталі. Для розробки методу обрахунку рейтингу гравця нам не потрібні характеристичні метрики його персонажів.

Наступним аналогом є найбільший на цей день веб-сайт для проведення TRPG – RPNation [10]. Він також містить елементи деякої статистики, але вона не

надається «third party» сервісами. Програмісти, що його створювали, використовували бібліотеки мов програмування, які забезпечують аналітичну обробку статистичної інформації. На рисунку 1.4 проілюстровано, як саме виглядає результат цієї обробки.

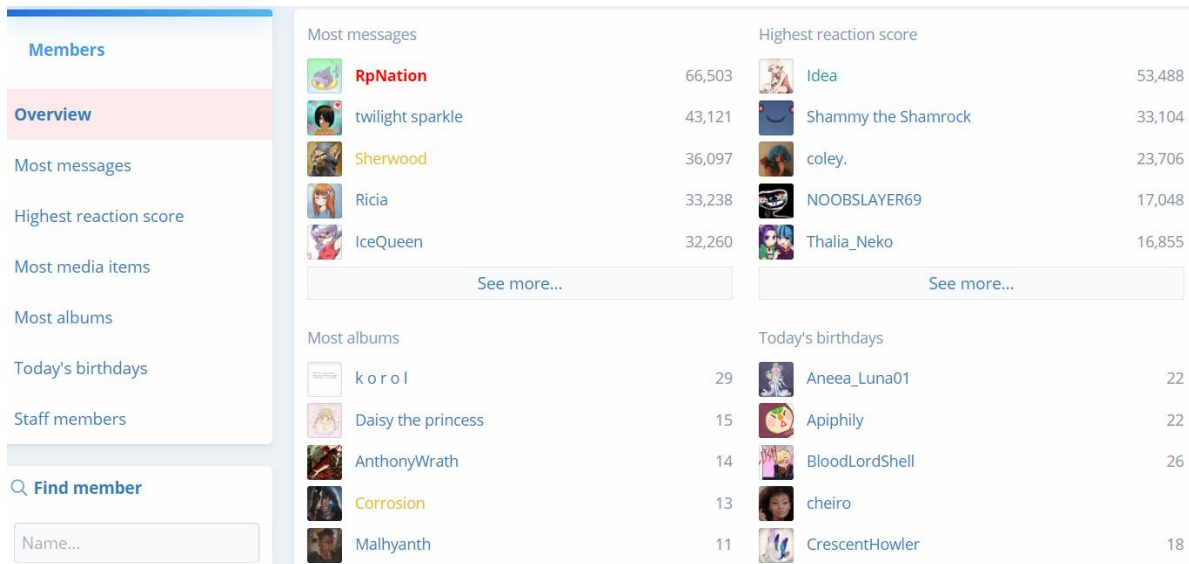


Рисунок 1.4 – Представлення статистики на веб-сайті RPNation

Як видно з рисунка, рольовий веб-сайт має декілька вкладок. Зосередимось на вкладці загального огляду – Overview. Вона містить рейтинг гравців за наступними характеристиками, деякі з яких будуть запозичені для розробки нового методу:

- Найбільша кількість повідомлень.
- Найбільша швидкість написання.
- Найбільша кількість альбомів.
- Найбільша кількість медіа, та інш.

Недоліками є: невелика кількість метрик для глибокого аналізу та відсутність будь-якої реалізації у вигляді наглядних графіків, схем та таблиць.

Третім аналогом є мультитематична платформа Amino (рис.1.5). Це соціальна мережа для творчих спільнот за різними напрямками, яка частково адаптована під рольову гру та теж містить деякі елементи статистики.

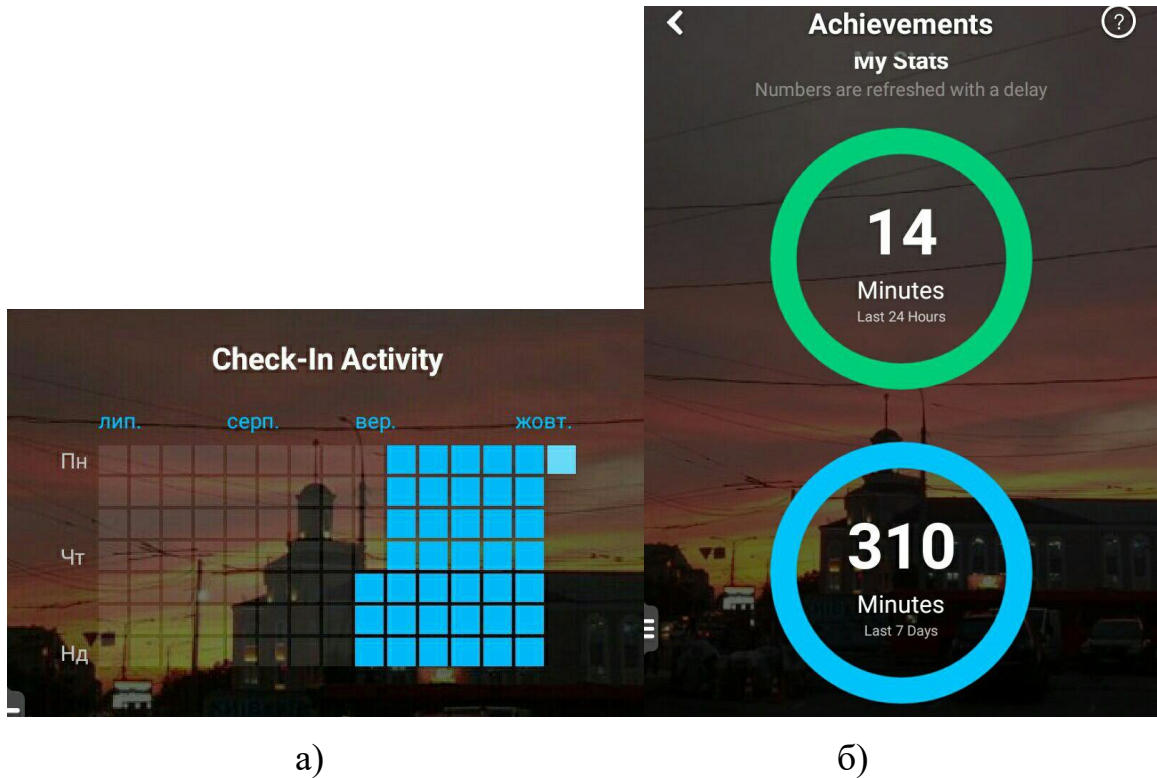


Рисунок 1.5 – Представлення статистики у мережі Amino

На платформі впроваджено систему Check in (з англ.: реєстрація), її вигляд показано на рисунку 1.5(а). Користувач повинен повідомити про свою присутність на Amino один раз за добу та накопичувати бонуси, що збільшуються у арифметичній прогресії, з кожним днем.

На сторінці «Achievements» (з англ.: досягнення), що на рисунку 1.5(б), користувач бачить, скільки Check in'ів зробив, скільки хвилин провів на сайті за 24 години або за тиждень. Нижче є дані про те, скільки користувач створив записів блогу та скільки послідовників отримав за весь час.

Також, що є цікавою системою, на платформі впроваджено рейтинг користувачів. Усього є 15 рівнів (рис 1.6) і кожен з них має свою оригінальну назву, враховуючи те, Аміно якої тематики користується гравець. Чим більше бонусів накопичує користувач за допомогою Check in'ів та взагалі діяльності на ресурсі, тим вище його рівень.

З досягненням високого рівня юзер може претендувати на місце модератора та допомагати у моніторингу.

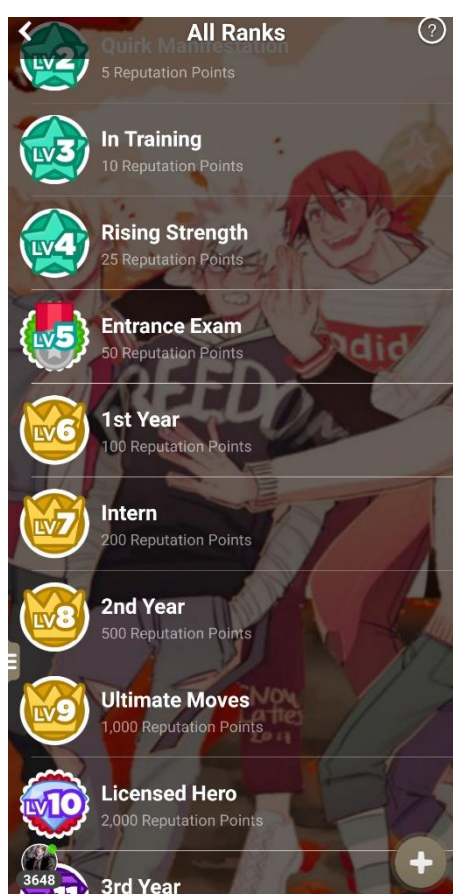


Рисунок 1.6 – Ранги рейтингу користувачів на прикладі Аміно по мотивам анімаційного серіалу

Недоліком платформи є неповноцінність даних. Цікава система, яку можна запозичити та вдосконалити, але Аміно не створений для рольової гри та не містить потрібних компонентів для її проведення.

Результати варіантного аналізу підходів до представлення статистики запропоновано у таблиці 1.2.

Таблиця 1.2

Результат варіантного аналізу методів представлення статистичних даних

Вид представлення статистики	Характерні ознаки представлення статистики у аналогів						
	Метрики персонажа	Метрики активності гравця	Рейтинг активності	Заохочення активності	Інформатив на кількість метрик	Графічне представл.	Сума
Комп'ютерна рольова гра	+						1
RPNation		+	+				2
Amino		+	+	+		+	4

Отже, у мережі є безліч ресурсів для проведення рольової гри (не тільки форуми, а й щоденники, соціальні мережі, сервіси, веб-сайти, адаптовані під проведення ігор) але мало хто з них використовує навіть примітивну аналітичну складову. У розділі наведені ті популярні середовища, які не нехтують аналітичним аспектом.

Згідно аналізу, проведеному у таблиці 1.2, було прийнято рішення при розробці нового методу та його програмної реалізації схилитися до вигляду представлення статистики сервісу Amino з суттєвим його покращенням у процесі розробки.

1.3 Аналіз та обґрунтування вибору одного з сучасних статистичних методів

Перш ніж перейти до розробки методу, потрібно визначити те, яким чином краще аналізувати дані. Для цього потрібно розглянути основні методи збору статистичних даних і вибрати один чи декілька для їх комбінації та модифікації.

Існують такі етапи збору статистичних даних:

- статистичне спостереження;
- зведення і групування матеріалів статистичного спостереження;
- абсолютні і відносні статистичні величини;
- варіаційні ряди;
- вибірка;
- виведення рядів динаміки [11].

Методи збору та аналізу статистичних даних відрізняються за цілями та масштабами та завершується розрахунком і аналізом різних за видом і формою вираження статистичних показників. Класифікація видів статистичних спостережень зображено на рисунку 1.7.

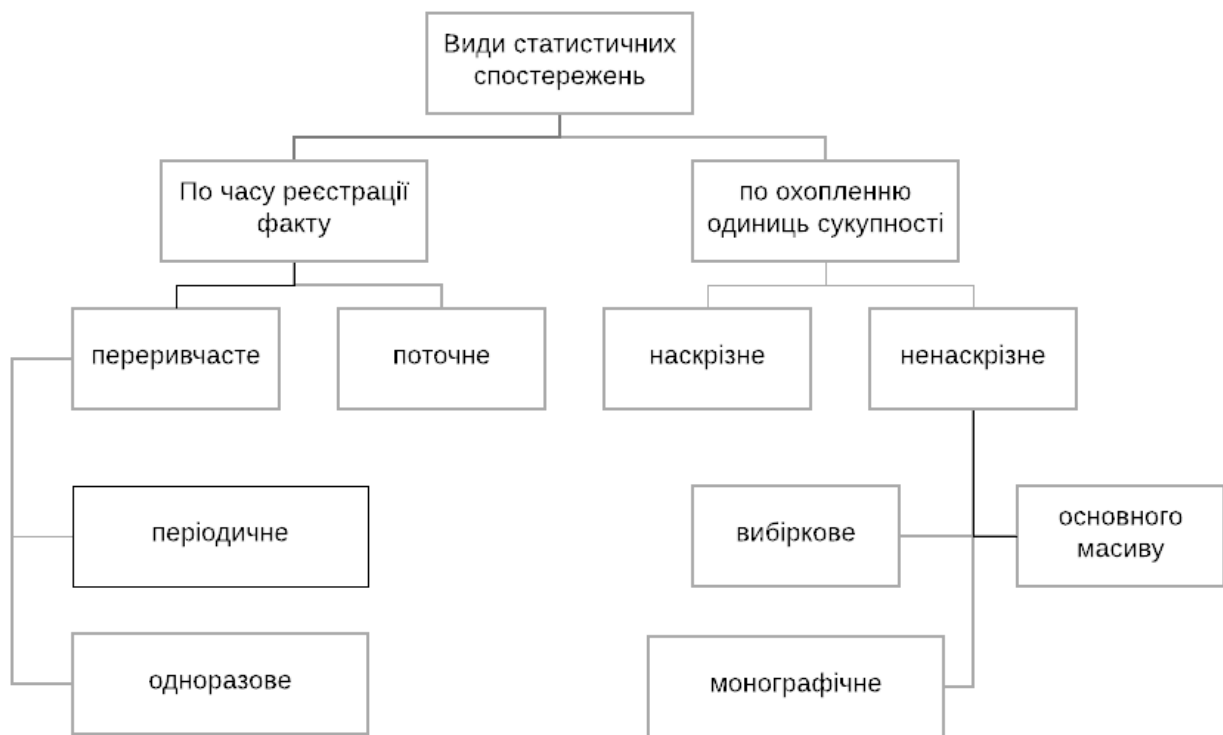


Рисунок 1.7 – Класифікація видів статистичного спостереження

На схемі рисунку 1.7 вже присутній шлях кінцевого вибору виду спостереження, враховуючи охоплення висвітлених проблем під час проведення

рольової гри, що потребують рішення. Обґрунтування такого вибору наведено у таблиці 1.3.

Таблиця 1.3

Обґрунтування вибору виду спостереження

№	Вид	Характеристика	Висновок
По часу реєстрації			
1	Поточне	реєстрація змін досліджуваних явищ здійснюється в міру їх настання	- рейтинг активності створюється раз на період моніторингу
2	<u>Періодичне</u>	<u>вид переривчастого спостереження, що повторюється через певні проміжки часу</u>	+
3	Одноразове	збір відомостей про кількісні характеристики досліджуваного явища здійснюється безпосередньо в процесі його дослідження	- Збір та обчислення даних для рейтингу активності повинен мати динаміку регулярності
По охопленню одиниць сукупності			
1	Наскрізне	обстеженню піддаються всі одиниці досліджуваної сукупності	- високі витрати трудових і фінансових ресурсів на отримання і обробку всього масиву інформації
3	<u>Ненаскрізне</u>	<u>обстеженню піддається лише частина одиниць досліджуваної сукупності</u>	+
4	Вибіркове	випадковий відбір тих одиниць досліджуваної сукупності, які повинні бути піддані статистичному спостереженню	- Для обчислення рейтингу активності потрібні чіткий заздалегідь визначений перелік показників
5	<u>Основного масиву</u>	<u>обстеженню піддаються ті одиниці сукупності, які мають найбільшу вагу досліджуваного ознаки</u>	+
6	Монографічне	метою монографічного обстеження є характеристика вже наявних істотних тенденцій в розвитку досліджуваного явища	- Стандарти та тенденції у метриках визначення активності гравця TRPG ще невідомі

Отже, у таблиці 1.3 запропоновано пояснення причин такого вибору виду спостереження, як першого етапу збору статистичних даних.

По часу реєстрації факту спостереження повинно бути переривчастим – періодичним, оскільки, як правило, рейтинг складається один раз за моніторинговий період для винагороди лідерів. Навіть враховуючи той факт, що серед метрик активності будуть ті, що вимагатимуть оновлення кожен день, це все ще періодичне спостереження, а не спостереження у реальному часі.

По охопленню одиниць сукупності спостереження повинно бути ненаскрізним, оскільки не всі одиниці сукупності стануть об'єктами спостереження. Деякі з них існуватимуть, як константи, що не змінюються в часі і не потребуватимуть регулярного моніторингу. Вид основного масиву був обраний завдяки поняттю ваги одиниць сукупності. Так як сама сукупність (перелік метрик для визначення активності) ще не запропоновано, для її логічного створення буде використано матричний метод, а отже експертну оцінку – тобто, поняття ваги фігуруватиме.

1.4 Висновок

У першому розділі магістерської дипломної роботи було проаналізовано проблематику під час проведення форумної рольової гри та визначено, що основними проблемами є: годмодинг, не ініціативність активістів, складність у моніторингу активності, втрата інтересу, відсутність фіксованої системи.

На даному етапі роботи було проведено варіантний аналіз деяких існуючих підходів ведення статистики. Обґрунтовано вибір виду статистичного спостереження.

2 РОЗРОБКА МЕТОДУ ТА ЗАСОБІВ ВЕДЕННЯ ФОРУМНОЇ РОЛЬОВОЇ ГРИ

2.1 Вибір та визначення ваг метрик активності гравців

При розробці системи розрахунку активності гравця форумної рольової гри однією з найважливіших завдань є оцінка та порівняння результатів їх діяльності. Потрібно врахувати те, що метрики, які складають оцінку діяльності, мають свою розмірність і вагу. Використання матричного методу дозволяє точно провести таку оцінку активності, крім того забезпечить мотивацію людей, націливши їх на досягнення поставлених завдань.

Один з найбільш вдалих способів вирішення описаних вище завдань – використання матричного методу вимірювання результативності (ММР), який відомий також під назвою «багатокритерійний метод вимірювання результативності».

ММР дозволяє звести у воєдино, виконати агрегацію та інтегрувати, показники так, що результатом роботи буде одне число. Це дає модераторам форумів можливість логічно і об'єктивно перерозподіляти заохочення і покарання між гравцями на основі виміряного результату.

ММР досить легко піддається автоматизації, що дозволяє застосовувати його у розроблюваному методі, для покращення проведення гри.

Матриця результативності покладена у основу методу, згідно його назви. Така матриця включає:

- оціночну шкалу;
- оцінку;
- оцінюючі показники з одиницями виміру і формату;
- значимість кожної метрики;
- результати, як окремо для кожної метрики, так і загальні.

Як вже зазначалося, на активність гравця впливає цілий ряд факторів, які можна отримувати за допомогою моніторингу діяльності гравця на форумі.

Діяльність характеризується декількома десятками критеріїв, та щоб визначити, які з них мають більший пріоритет у обрахунку рейтингу, було проведено опитування групи експертів з десяти чоловік. Кожному була запропонована анкета зі списком метрик активності гравця, які впливають на рейтинг. Експертами, балами від 1 до 10, було відмічено рівень впливу фактору.

Отримані результати наведені в таблиці 2.1.

Таблиця 2.1

Результати опитування експертів

Метрики активності	Думка експертів									
	1	2	3	4	5	6	7	8	9	10
1. Кількість персонажів	8	9	8	9	9	10	7	10	9	9
2. Кількість епізодів	10	10	10	9	10	8	10	9	10	10
3. Загальна кількість постів	9	10	9	10	10	10	8	10	10	10
4. Кількість слів у 1 пості	10	9	9	8	9	9	8	10	9	10
5. Залученість	8	8	7	6	7	8	8	9	8	8
6. Мінімальна кількість постів	6	7	7	8	8	9	10	8	9	9
7. Частота написання	9	9	10	10	9	8	7	8	10	9
8. Дедлайн	10	9	10	10	10	8	10	9	10	9
9. Відповідність форматування	5	6	6	6	7	5	6	7	5	7
10. Мінімальна кількість слів у пості	7	10	10	9	9	7	8	10	9	10
11. Максимальна кількість слів у пості	4	5	6	5	5	6	7	6	5	6
12. Кількість повторювальних слів	2	5	4	3	5	4	6	3	5	4
13. Середній інтервал часу між двома постами	9	10	9	9	8	9	10	10	9	8
14. Швидкість написання	8	7	7	7	8	7	8	9	6	7
15. Час, проведений на форумі	6	7	6	5	7	6	6	5	8	6

Тепер нам потрібно визначити показник ступеня узгодженості між експертами, тобто наскільки схожі їх думки. На практиці для визначення тісноти зв'язку двох ознак часто застосовується коефіцієнт рангової кореляції Спірмена[12], який демонструє рівень узгодженості експертів (формула 2.1).

Він приймає значення від -1 до 1 та використовується для вимірювання ступеня тісноти й напрямку зв'язку між кількісними та якісними ознаками, що характеризують однорідні об'єкти і ранжируваних за одним принципом.

Тепер необхідно проранжувати показники.

Ранжування – це процедура впорядкування об'єктів вивчення, яка виконується на основі переваги. Ранг – це порядковий номер значень ознаки, розташованих в порядку зростання або зменшення їх величин. Якщо значення ознаки мають однакову кількісну оцінку, то ранг всіх цих значень приймається рівним середньої арифметичної з відповідних номерів місць, які вони визначають. Дані ранги називаються зв'язковим.

$$W = \frac{S}{\frac{1}{12} * m^2 * (n^3 - n) - m \sum_{i=1}^m T_i}, \quad (2.1)$$

$$d_j = S_j - \frac{\sum_{i=1}^n S_i}{n}, \quad (2.2)$$

$$S_i = \sum_{k=1}^L R_{kI}, \quad (2.3)$$

$$T_i = \frac{1}{12} \sum_{e=1}^L (t_e^3 - t_e), \quad (2.4)$$

де L – кількість груп зв'язаних рангів;

t_e – кількість зв'язаних рангів в кожній групі;

n – кількість рангів;

m – кількість експертів;

S_i – сума рангів;

d_j – відхилення суми.

Для нашого випадку, кількість експертів = 10, кількість метрик = 15. За результатами таблиці 2.2 побудуємо матрицю рангів на базі отриманих даних таблиці 2.1.

Таблиця 2.2

Матриця рангів

Метрики активності	Думка експертів									
	1	2	3	4	5	6	7	8	9	10
1. Кількість персонажів	8	9.5	8	10.5	10.5	14.5	5	13	9	9.5
2. Кількість епізодів	14	13.5	13.5	10.5	14	8.5	13.5	8.5	13.5	13.5
3. Загальна кількість постів	11	13.5	10	14	14	14.5	9	13	13.5	13.5
4. Кількість слів у 1 пості	14	9.5	10	7.5	10.5	12	9	13	9	13.5
5. Залученість	8	7	6	4.5	4	8.5	9	8.5	5.5	6.5
6. Мінімальна кількість постів	4.5	5	6	7.5	7	12	13.5	5.5	9	9.5
7. Частота написання	11	9.5	13.5	14	10.5	8.5	5	5.5	13.5	9.5
8. Дедлайн	14	9.5	13.5	14	14	8.5	13.5	8.5	13.5	9.5
9. Відповідність форматування	3	3	3	4.5	4	2	2	4	2	4.5
10. Мінімальна кількість слів у пості	6	13.5	13.5	10.5	10.5	5.5	9	13	9	13.5
11. Середня кількість слів у пості	2	1.5	3	2.5	1.5	3.5	5	3	2	2.5
12. Кількість повторювальних слів	1	1.5	1	1	1.5	1	2	1	2	1
13. Інтервал часу між двома постами	11	13.5	10	10.5	7	12	13.5	13	9	6.5
14. Швидкість написання	8	5	6	6	7	5.5	9	8.5	4	4.5
15. Час, проведений на форумі	4.5	5	3	2.5	4	3.5	2	2	5.5	2.5

За даними таблиці 2.2., визначимо середню суму рангів, квадратів відхилень від середньої суми (табл. 2.3).

Значення кожної ознаки ранжуються за ступенем зростання (від 1 до n), потім визначається різниця (d) між рангами, відповідними одному спостереженню.

Таблиця 2.3

Середня сума рангів та квадратів відхилень

Метрики	Експерти										Сума рангів, S_i	d_j	d_j^2
	1	2	3	4	5	6	7	8	9	10			
1	8	9.5	8	10.5	10.5	14.5	5	13	9	9.5	97.5	17.5	306.25
2	14	13.5	13.5	10.5	14	8.5	13.5	8.5	13.5	13.5	123	43	1849
3	11	13.5	10	14	14	14.5	9	13	13.5	13.5	126	46	2116
4	14	9.5	10	7.5	10.5	12	9	13	9	13.5	108	28	784
5	8	7	6	4.5	4	8.5	9	8.5	5.5	6.5	67.5	-12.5	156.25
6	4.5	5	6	7.5	7	12	13.5	5.5	9	9.5	79.5	-0.5	0.25
7	11	9.5	13.5	14	10.5	8.5	5	5.5	13.5	9.5	100.5	20.5	420.25
8	14	9.5	13.5	14	14	8.5	13.5	8.5	13.5	9.5	118.5	38.5	1482.25
9	3	3	3	4.5	4	2	2	4	2	4.5	32	-48	2304
10	6	13.5	13.5	10.5	10.5	5.5	9	13	9	13.5	104	24	576
11	2	1.5	3	2.5	1.5	3.5	5	3	2	2.5	26.5	-53.5	2862.25
12	1	1.5	1	1	1.5	1	2	1	2	1	13	-67	4489
13	11	13.5	10	10.5	7	12	13.5	13	9	6.5	106	26	676
14	8	5	6	6	7	5.5	9	8.5	4	4.5	63.5	-16.5	272.25
15	4.5	5	3	2.5	4	3.5	2	2	5.5	2.5	34.5	-45.5	2070.25
Разом в середньому	120	120	120	120	120	120	120	120	120	120	1200		20364

Згідно таблиці 2.3, за значимістю метрики розподілилися наступним чином:

1. Загальна кількість постів.
2. Кількість епізодів.
3. Дедлайн.
4. Кількість слів у 1 пості.
5. Середній інтервал часу між двома постами.
6. Мінімальна кількість слів у пості.
7. Частота написання.
8. Мінімальна кількість постів.
9. Залученість.
10. Кількість персонажів.

11. Швидкість написання.
12. Час, проведений на форумі.
13. Середня кількість слів у пості.
14. Відповідність форматування.
15. Кількість помилок.

Сума по стовпчиках матриці у таблиці 2.3 рівні між собою, значить, матриця складена правильно. Тепер проведемо оцінку середнього ступеня узгодженості думок всіх експертів. Це можливо, використавши формули 2.1 та 2.4, де $S = 20364$, $n = 15$, $m = 10$.

Отже:

$T_1 = 6.5$; $T_2 = 12.5$; $T_3 = 11$; $T_4 = 8.5$; $T_5 = 11.5$; $T_6 = 8.5$; $T_7 = 19$; $T_8 = 15.5$;
 $T_9 = 17.5$; $T_{10} = 11.5$.

Визначимо коефіцієнт конкордації виходячи з отриманих даних:

$$\sum_{i=1}^m T_i = 6.5 + 12.5 + 11 + 8.5 + 11.5 + 8.5 + 19 + 15.5 + 17.5 + 11.5 = 122$$

$$W = \frac{20364}{\frac{1}{12} * 10^2 (15^3 - 15) - 10 * 122} = 0.77$$

З отриманих результатів видно, що $W = 0.77$ – це наявність високого ступеня узгодженості думок експертів. Також в результаті опитування було складено список метрик за рівнем їх важливості під час обрахування рейтингу активності гравця.

Згідно отриманих результатів після проведення дослідження, за допомогою матричного методу були визначені п'ятнадцять найбільш значущих метрик для обчислення рейтингу активності гравця під час проведення форумної рольової гри.

2.2 Аналіз існуючих методів обрахунку рейтингу гравців

Як правило, завдання розробки алгоритму-формули розрахунку рейтингу активності є складнішим, ніж може здатися на перший погляд. Існує безліч підходів до питання обрахунку, і кожен ресурс розробляє власну формулу, але між ними все ж таки є дещо спільне.

Класифікації методів розрахунку рейтингу не існує як такої, але їх можна поділити на: прості (методи однієї дії) та складні.

Розглянемо декілька підходів до обрахування рейтингу за простими методами однієї дії і обґрунтуємо, у чому їх недолік.

Найчастіше, при обрахунку рейтингу, враховуються лише позитивні та негативні оцінки відвідувачів, тобто лише якісні абсолютні показники. Даний принцип, реалізовані формулами 2.5 та 2.6 відповідно використовують такі відомі сайти, як Amazon та Urban Dictionary [13].

$$R = positive_{marks} - negative_{marks} , \quad (2.5)$$

де $positive_{marks}$ – кількість позитивних оцінок;

$negative_{marks}$ – кількість негативних оцінок.

Припустимо, у першому випадку, у одного об'єкта є 600 позитивних оцінок і 400 негативних, тобто в результаті 60% позитивних. У другого об'єкта 5500 позитивних оцінок і 4500 негативних, тобто в результаті 55% позитивних. Даний алгоритм розмістить другий об'єкт (з рейтингом 1000, але всього з 55% позитивних оцінок) вище першого об'єкта (з рейтингом 200 і з 60% позитивних оцінок).

$$R = average = \frac{positive_{marks}}{all_{marks}} , \quad (2.6)$$

де all_{marks} – кількість усіх оцінок отриманих об'єктом.

У другому випадку, середня оцінка добре працює, якщо у об'єкта багато оцінок. Припустимо, що у одного об'єкта 2 позитивні оцінки і 0 негативних, а у другого об'єкта 100 позитивних оцінок і 1 негативна. Даний алгоритм розмістить другий об'єкт (з купою позитивних оцінок) нижче першого об'єкта (з дуже малим числом позитивних оцінок). Метод не є інформативним для складних ресурсів та не підвищить зацікавленість гравця.

Тепер розглянемо декілька складних методів. Тут ми також маємо лише якісні абсолютні показники, а саме бали ранжування від 1 до 5. Для зручності ранжування 1-5 балів перетворюють у ранжування від 0 до 1 балів.

Розрахунок рейтингу відбувається за формулою 2.7.

$$R = R_u - \frac{R_u - R_q}{(K + 1)^{\frac{K * 0,02}{R_u \pm 0,1}}}, \quad (2.7)$$

де R_u – це оцінка ведення гри гравця іншим гравцем;

R_q – оцінка гри гравця модератором;

K – кількість оцінок, що надали інші гравці.

Графік залежності оцінок R_u від R_q проілюстровано на рисунку 2.1.

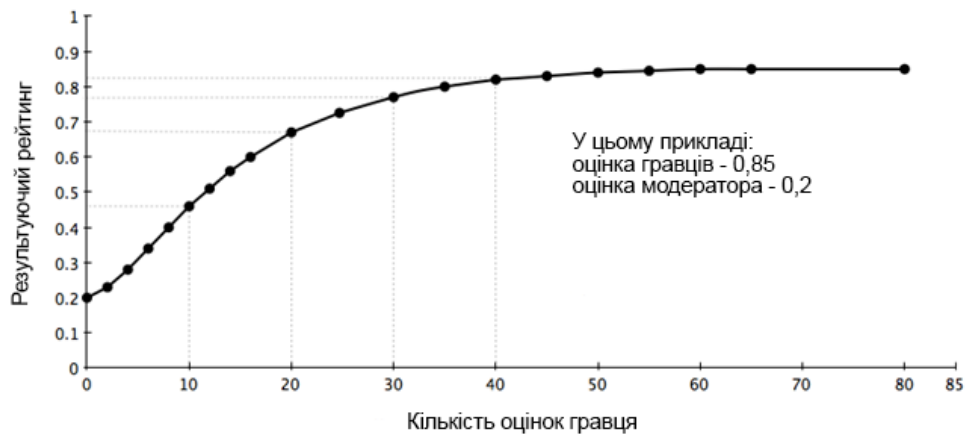


Рисунок 2.1 – Графік залежності оцінки гравців і оцінки модератора

Формула пов'язує ці три значення так, що рейтинг переходить від R_q до R_u , коли зростає кількість оцінок гравців. Якщо їх немає, то рейтинг оцінюється лише модератором. Якщо їх дуже багато, він дорівнює оцінкам, що дали гравці, при цьому думка модератора вже суттєво на рейтинг не впливає.

Щоб побачити недоліки, необхідно розглянути формулу обрахунку показника R_u (формула 2.8).

$$R_u = \left(\frac{\sum_{i=1}^K m * c}{\sum_{i=1}^K c} \right), \quad (2.8)$$

де c – коефіцієнт якості оцінки;

m – оцінка гравця, яка набуває значень від 0 до 1 в залежності від того, скільки балів (від 1 до 5) отримав гравець.

Недолік методу полягає у тому, що метод містить занадто багато критеріїв, на відміну від попереднього методу. У всіх параметрів максимальне значення 1, і якщо прирівняти до одиниці всі змінні, сформується сумнівний результат.

Існує ще один метод, який називається «нижня межа довірчого інтервалу Вільсона для параметра Бернуллі» [14], яка обраховується за формулою 2.9.

$$R = \left(p + \frac{z^2}{2n} \pm \frac{z_{\alpha/2}}{2} * \sqrt{\frac{\left[p(1-p) + \frac{z_{\alpha/2}^2}{4n} \right]}{n}} \right) / \left(1 + \frac{z_{\alpha/2}^2}{n} \right), \quad (2.9)$$

де p – частка позитивних оцінок;

$z_{\alpha/2}$ – квантиль стандартного нормального розподілу;

n – загальна кількість оцінок.

Метод використовується тоді, коли необхідно знайти баланс між частиною позитивних оцінок та невизначеністю малого числа спостережень. Формула Вільсона дає можливість відповісти на питання, чи можна з імовірністю 95% сказати, якою буде "реальна" частка позитивних оцінок, маючи даний набір оцінок. Метод також оперує позитивними та негативними оцінками, але вважається складним. При обчисленні використовується мінус там, де \pm , щоб обчислити нижню межу.

Довірчий інтервал Вільсона можна застосувати не тільки для такого обрахунку. Наприклад, він може використовуватися в наступних цілях:

- виявлення спаму або зловживань: скільки людей, які побачили новину на форумі, помітять його як спам;
- створення списку «найкращого»: скільки людей, які побачили повідомлення, помітять його як «найкраще»;

Тому, частково цей метод можна використати у нашому методі, наприклад, для того, щоб створити список учасників квесту: дізнатись, який відсоток гравців вирішить брати участь у квесті, щоб налаштувати під нього норми мінімумів для обрахунку рейтингу.

Отже, у підрозділі було розглянуто лише деякі, але найбільш використовувані методи обрахунку рейтингу, жоден з яких навіть приблизно не використовує метрики для обрахунку саме діяльності під час гри гравця, як персонажа, а не як користувача форуму, тому актуальності набуває створення нового методу обчислення рейтингу активності.

2.3 Розробка нового методу визначення активності гравця

Для того, щоб підвищити ефективність проведення TRPG, в першу чергу, потрібна універсальна система правил, якій слідуватиме переважна більшість гравців у світі. Для цього було проведено дослідження, який показник активності гравця у TRPG вважається найзмістовнішим, що обґрунтовано в підрозділі 2.1.

Було обрано 15 показників активності, що розташовані від найважливішого, до менш важливішого. Опишемо запропоновані метрики, методи їх обрахунку.

1. Загальна кількість постів за період моніторингу (n_{post}). Пост (вимірюється у словах) – одиниця тексту за один хід гравця. Якщо за цей період гравець не написав жодного поста за певного персонажа, рейтинг за цього персонажа дорівнюватиме 0. Якщо у гравця тільки 1 персонаж, і за нього відсутні пости, рейтинг гравця не обчислюється взагалі.

2. Загальна кількість епізодів (n_{ep}). Епізод (шт.) – одиниця сюжету, у межах якої взаємодіють гравці. Епізод може бути квестом – це епізод, який рухає загальний сюжет гри. Так як і у випадку загальної кількості постів, якщо кількість епізодів за конкретного персонажа дорівнюватиме 0, рейтинг для нього не обраховується. Якщо у гравця тільки один персонаж, рейтинг для гравця не обраховується взагалі. Принцип обрахунку/не обрахунку за двома першими найважливішими метриками проілюстровано блок-схемою на рисунку 2.2.

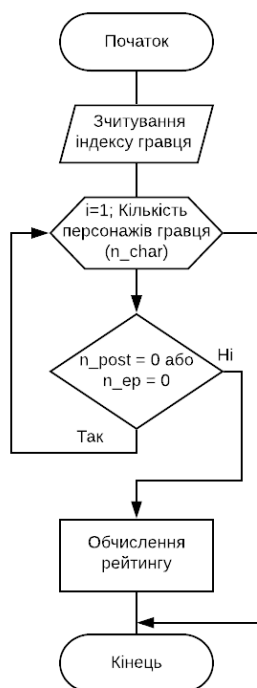


Рисунок 2.2 – Блок-схема алгоритму обрахунку рейтингу при відсутності у гравця епізодів чи постів

3. Дедлайн (t_{max}) – максимальний термін часу, за який гравець має відповісти партнеру. Він може змінюватися в залежності від вимог, але по замовчуванню $t_{max} = 2$ дні = 48 год. Він не залежить від кількості персонажів гравця та його завантаженості, тому гравцю слід стежити за терміном дедлайну та сповіщати відповідальних осіб при неможливості писати.

4. Кількість слів у одному пості (n_{words}). Обчислюється за формулою 2.10.

$$n_{words} = \frac{words}{n_{post}} \quad (2.10)$$

де $words$ – кількість слів у всіх написаних постах на період моніторингу.

5. Середній інтервал часу між двома постами (t).

Обраховується в годинах, щоб уникнути незручних значень, якщо обчислювати у секундах, за формулою 2.11.

$$t = \frac{t_1 + t_2 + t_3 + \dots + t_n}{(n_{post} - 1)} \quad (2.11)$$

де $t_1, t_2, t_3 \dots t_n$ – це інтервали часу написання між кожним постом та відповіддю на нього співгравцем.

6. Мінімальна кількість слів у 1 пості ($word_{min}$) – та норма, якої гравець має дотримуватися весь період моніторингу, тобто жоден пост не повинен бути меншим за $word_{min}$.

Встановлюється на початку цього періоду за статистичними результатами попереднього. Може корегуватися модераторами або бути обрахованим автоматично шляхом знаходження такого найменшого значення кількості слів за попередній період моніторингу, яке все ще є більшим за $word_{min}$.

Тобто, з кожним періодом моніторингу ця норма буде несуттєво

збільшуватись, провокуючи професіональній зріст гравця.

7. Частота написання (v) – величина, яка відображає те, як часто гравець відповідає партнеру по епізоду.

Обраховується за формулою 2.12.

$$v = \frac{n_post}{t} = \frac{n_post * (n_post - 1)}{t1 + t2 + t3 + \dots + tn} \quad (2.12)$$

Якщо перед постом гравця є пост його партнера, знаходиться різниця часу між ними (tn) – те, як швидко гравець відповів співгравцю. Гравець нотифікується, якщо $t > t_max$.

Блок-схема алгоритму обчислення метрик часу зображено на рисунку 2.3.

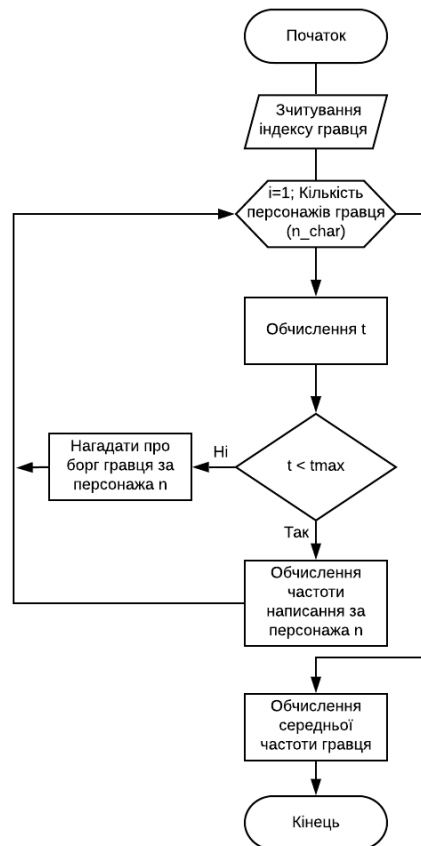


Рисунок 2.3 – Блок-схема алгоритму обчислення метрик часу

8. Мінімальна кількість постів ($post_min$) – показник, фіксоване число кількості постів за моніторинговий період, які зобов’язується написати гравець, Величина залежить від кількості його персонажів (рис 2.4). Мінімальна норма постів обраховується за таким самим принципом, як і мінімальна кількість слів.

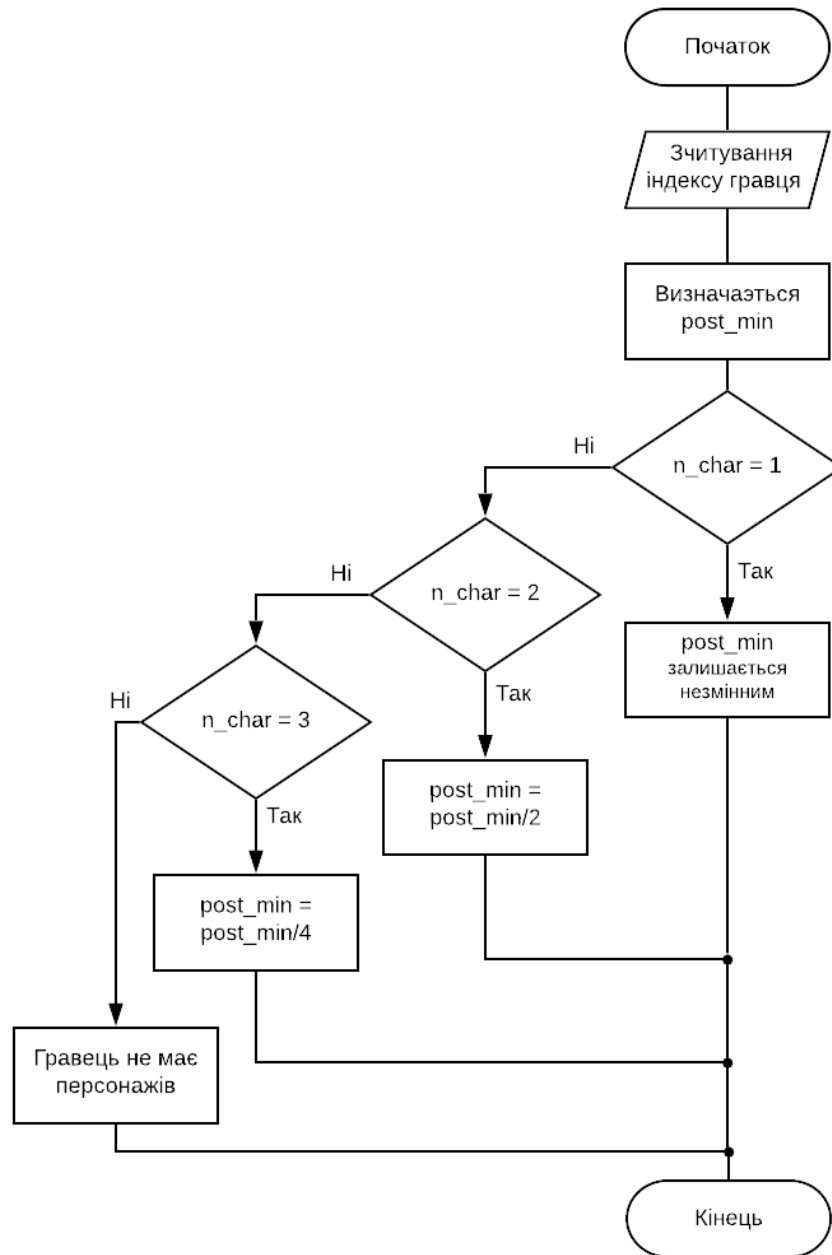


Рисунок 2.4 – Блок-схема алгоритму формування змінної $post_min$

9. Залученість [15] (Z) – середня кількість реакцій (лайків / відміток «Подобається», коментарів, але у нашому випадку це кількість постів) які робить один гравець по відношенні до одиниці контенту (у нашому випадку – Епізоду).

Часто, представники годмодингу пишуть мало, тому на гравців, у яких залученість менше норми варто звертати увагу. Залежить від кількості персонажів. Обраховується за формулою 2.13.

$$Z = \frac{n_post}{n_ep}. \quad (2.13)$$

Для того, щоб нарахувати гравцю залученість, необхідно, щоб кількість епізодів була > 0 . Змінна може набувати лише додатних значень. $Z < 1$ у випадку, коли епізодів більше, ніж постів – гравцю варто передивитися закинуті епізоди, в межах яких він не бажає грати, та закрити їх.

10. Кількість персонажів, яку має гравець (n_char) – інколи необмежена, інколи лімітована величина (часто дозволяється мати не більше 3-х). Від n_char залежать декілька інших метрик, наприклад, норма постів, так як кількість персонажів це безпосередня характеристика завантаженості гравця.

11. Швидкість написання (слово/с). Відлік часу починається, коли курсор встановлено у поле для введення тексту, та закінчується з надсиланням на сервер. Не середня величина. Вказується під кожним написаним постом та обчислюється відношенням кількості слів у поточному пості до часу. Не впливає на рейтинг, але може враховуватися, як додатковий фактор при змаганнях на швидкість, або для задоволення зацікавленості гравця. Окремий графік демонструватиме, як швидкість гравця збільшувалась чи зменшувалась з кожним періодом моніторингу.

12. Час проведений на форумі (T , год). Якщо значення досягне одного місяця або року, гравець заохочується адміністрацією. Також, якщо значення буде

меншим 1 год за тиждень – гравець нотифікується.

13. Середня кількість слів у одному пості обчислюється за формулою 2.14.

$$\text{word}_{\text{avg}} = \frac{\text{word}_{\text{max}} + \text{word}_{\text{min}}}{2} \quad (2.14)$$

де word_{max} та word_{min} – це максимум і мінімум слів у одному пості, які дозволяються модераторами спільноти.

Чим ближче значення n_{words} до word_{avg} , тим краще.

14. К-сть граматичних помилок (grammar_errors) – якісна метрика, яка характеризує якість гри гравця. Зазвичай, представники годмодингу пишуть примітивно та неуважно, тому метрика може допомогти у виявленні таких гравців. Обчислює кількість граматичних помилок у всіх написаних гравцем постах за період моніторингу.

15. Відповідність форматування – якісна метрика, що характеризує вигляд тексту. Текст у рольовій грі форматується для зручності читання. Слова персонажа (починається діалогом, з тире) виділяють напівжирним, думки (беруться в лапки) – курсивом, дії – звичайним шрифтом. Виявляється наявністю тегів форматування ([B] для жирного, [I] для курсиву).

Рейтинг обраховується за формулою 2.15.

$$\text{Rez} = 2 * \frac{v * n_{\text{post}}^2}{(n_{\text{post}}(\text{word}_{\text{max}} + \text{word}_{\text{min}}) - 2 * \text{words}) * n_{\text{ep}}} = \frac{v * Z}{|\text{word}_{\text{avg}} - n_{\text{words}}|} \quad (2.15)$$

Після операцій над перерахованими метриками, результатами є індивідуальні графіки і таблиці окремих метрик зі статистикою гравця і усіх його персонажів, рейтинги гравців за окремими параметрами, що заохочуватиме гравця бути першим якщо не у загальному рейтингу, тоді хоча б з окремого параметра, та головна рейтинг-таблиця активності користувачів.

Блок-схему обрахунку метрик за запропонованим методом визначення рейтингу гравців на основі отриманих показників наведено у додатку В.

Отже, у роботі розглянуто принцип роботи методу обчислення рейтингу гравця для покращення ефективності проведення TRPG вигляду play-by-post, вибір кількісних та якісних метрик та спосіб їх поєднанні і обробки у вигляді блок-схеми.

В кінці обчислень ми отримуємо значення, яке на етапі візуалізації результатів сформує рейтингову таблицю та графіки активності користувачів як загальні, так і по окремим метрикам.

2.4 Розробка алгоритму розрахунку активності гравця за запропонованим методом

Блок-схема алгоритму знаходиться у додатку В. Розглянемо алгоритм поетапно.

1 та 2: на перших етапах зчитується індекс гравця, щоб отримати повний звіт на нього.

3: так як у гравця може бути декілька персонажів, аналітичний модуль обраховує рейтинг для кожного з них всередині циклу. Вихід здійснюється, коли дані за усіх персонажів були обраховані.

4: обраховується кількість епізодів та кількість постів за n-ого персонажа.

5: при умові, якщо кількість епізодів або кількість постів = 0, аналітичний модуль завершує роботу.

6: обрахунок зацікавленості (Z) відношенням кількості постів до кількості епізодів. Зацікавленість (i взагалі рейтинг) не обраховуються взагалі, якщо кількість постів або кількість епізодів дорівнює нулю.

7: якщо зацікавленість менше або дорівнює 1, це означає, що у гравця або забагато відкритих епізодів, або замало написаних в них постів, тому система нотифікує гравця про це – 8.

9: обраховується середня кількість слів у одному пості.

11: обраховується константа середнього значення (норми) поста та час між двома найближчими написаними постами.

12: обраховується частота.

13: головна формула функціональної залежності між метриками. Так як чим більше частота написання поста та залученість – тим вище активність, метрики у чисельнику формули. У знаменнику обраховується різниця між загальною середньою нормою слів та середньою кількістю слів у одному пості гравця. Чим більше його пост відхиляється від норми, тим нижче результат рейтингу активності.

14: модуль виходить з циклі, масив результатів сортується.

Під час зчитування індексу гравця (`player_index`), по черзі обраховуються дані за кожного персонажа окремо, формуючи масив, і в кінці обирається той персонаж, за якого на виході найбільше значення ($Z[\max]$, на користь гравця). Якщо результати низькі за усіх персонажів, модерація нотифікує гравця з порадою прибрати одного персонажа та зменшити навантаження.

Отже, алгоритм обрахунку рейтингу гравця реалізовано за новим методом, який полягає у об'єднанні якісних та кількісних метрик під час проведення форумної рольової гри, та залежить від залученості, частоти написання та середньої кількості слів користувача.

2.5 Висновок

У другому розділі магістерської дипломної роботи було розроблено удосконалений метод обрахунку рейтингу активності гравців.

Метод було розроблено на підґрунті матричного методу, який за допомогою думки експертів та визначення ваг метрик, допоміг у ранжуванні цих метрик від найбільш до найменш важливих у діяльності гравця рольової спільноти.

Для обчислення рейтингу активності гравця було обрано 15 метрик, а саме: загальна кількість постів, кількість епізодів, дедлайн, кількість слів у 1 пості, середній інтервал часу між двома постами, мінімальна кількість слів у пості, частота написання, мінімальна кількість постів, залученість, швидкість написання, кількість персонажів, час на форумі, відповідність форматування, максимальна кількість слів у пості, кількість повторювальних слів.

Запропоновано алгоритми обрахування цих метрик та загальний алгоритм обчислення рейтингу гравця, який використовує усі метрики та об'єднує в один інтегральний показник активності.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АНАЛІТИЧНОГО МОДУЛЯ ЗА НОВИМ МЕТОДОМ

3.1 Аналіз та обґрунтування вибору засобів реалізації методу

Розробка методу обчислення рейтингу гравців під час проведення форумної рольової гри реалізована у вигляді аналітичного модулю на web-сайті, для більшої зручності у її користуванні та підвищення якості взаємодії модуля з користувачем. Саме тому для реалізації цієї задачі використовуються технології типу web.

За останнім рейтингом за 2019 рік, найпопулярнішими мовами програмування вважаються: Java, C# і JavaScript [16]. Всі вони призначені для роботи з web. Поділяються на:

- клієнтські – ті які виконуються на стороні користувача, тобто в браузері;
- серверні – ті що виконуються на стороні системи і є недоступними для клієнта.

Java є об'єктно-орієнтованою мовою програмування, розробленою Джеймсом Ослінгом. Запущена в 1995 році в якості одного із ключових компонентів платформи JAVA Sun Microsystems [17]. Java і C# можна розглядати як дві мови-нащадка C, розроблених з різних міркувань, що пішли внаслідок цього, різними шляхами. У зв'язку з цим становить інтерес порівняння даних мов.

З переваг мови програмування Java можна відокремити наступні:

- об'єктно-орієнтованість;
- кроссплатформенність;
- безпечність. Методи перевірки автентичності засновані на шифруванні з відкритим ключем;
- міцність. Мова усуває помилки в різних ситуаціях, спираючись в основному на час компіляції;

- багатопотоковість. Завдяки функціям багатопоточності, можна писати програми, які можуть виконувати безліч завдань одночасно, що дозволяє розробникам створювати налагоджені інтерактивні додатки;

- динамічність. Програмування на Java вважається більш динамічним, ніж на C або C #, так як він призначений для адаптації під час мінливі умови під час обробки даних.

JavaScript – це теж мова web-розробки, яка була створена в 1995 році, а зараз набуває усе більшої популярності. На період 2019-го року він майже зрівнявся по популярності з мовою Java. Це ована багаторівнева скриптова мова програмування, її основна платформа – веб-браузер, але вона також отримала нещодавно поширення на веб-серверах за допомогою Node.js [18]. Створення інтерактивності на веб-сторінках – це основне призначення цієї мови.

Тоді як мова Java була популярною на початку, але зараз її популярність падає, популярність JS продовжує зростати кожного року. Спочатку мова JS не користувалася популярністю серед методів візуалізації статистичних даних, але з розробкою нових бібліотек, фреймворків та технологій, генерувати графіки та таблиці на JS стало легше [19].

C# – це об'єктно-орієнтована мова. На відміну від інших мов, вона суворо типізована, тому вона добре підходить для початківців. Зазвичай, серверну частину (контроллер) MVC пишеться як раз на C# з використанням додаткових технологій. MVC представляє собою стандартний шаблон розробки. Деякі типи веб-додатків вигідно створювати на базі такого шаблону.

Для інших може бути цілеспрямовано використання традиційної схеми додатків ASP.NET, що базується на веб-формах та зворотній передачі. В деяких випадках можлива комбінація двох підходів: застосування однієї схеми не виключає використання іншої, що дозволяє використовувати найкраще підходи для вирішення кожної конкретної задачі [20].

Результати критеріального порівняння технологій розробки модуля наведені в табл. 3.1. Ці технології забезпечують необхідну функціональність, ефективну підтримку процесів створення програми та наповнення його інформаційним контентом [21]. Вони також забезпечують його швидку і надійну роботу, спрощену можливість створювати нові компоненти, розширювати функціональність існуючої системи і полегшують майбутню підтримку системи. У таблиці 1 – характеристика підтримується, 0 – характеристика не підтримується, 0,5 – підтримується частково.

Таблиця 3.1

Порівняння сучасних технологій розробки для веб

Технологія \ Характеристика	Java	Java Skript	C#
Бібліотека для аналітики	0	1	1
Простота використання	1	1	1
Розподіл дизайну та логіки	0,5	0,5	1
Наявність доступних програмних бібліотек	0,5	1	1
Швидкодія	0,5	1	0,5
Відкритий код	1	1	0
Сума	3,5	5,5	5

Виходячи з отриманих результатів аналізу можна зробити висновок про наявність переваг у використанні Java Script-технології. Хоча вона є досить молодою мовою, але вже користується значною популярністю серед веб-розробників і сьогодні вважається однією з найпопулярніших мов для розробки. Головними перевагами JS є: наявність бібліотеки для обробки аналітичних даних,

швидкодія рендерингу веб-сторінок та графіки на ній за допомогою GPU та технології WebGL, проста інтеграція скриптом на різні ресурси рольової гри, де необхідне використання аналітичного модуля.

Тепер потрібно вибрати середовище. Для доцільного вибору інтегрованого середовища розробки (або IDE) потрібно орієнтуватися на обрану мову програмування. У нашому випадку – це Java Script, але для того, щоб програмний додаток мав необхідний рівень гнучкості, яка забезпечується можливістю використовувати допоміжні мови програмування та скрипти, потрібно розглянути універсальні середовища розробки. Найпоширенішими IDE, які здатні працювати з Python, та допоміжними мовами: Visual Studio Code, Komodo Edit, Eclipse.

Komodo Edit – це безкоштовний текстовий редактор для динамічних мов програмування. Він відмінно підходить для розробників, які пишуть на різних скриптових мовах. На даний момент Komodo Edit підтримує роботу з Perl, PHP, Python, Ruby, Tcl, JavaScript, CSS, HTML і XML. Важливою особливістю програми є її кроссплатформенність (GNU / Linux, Apple Mac OS, Microsoft Windows).

Із основних переваг можна виділити:

- програма поширюється під безкоштовною ліцензією, не провокуючи ніяких проблем з поширенням програми та її завантаженням;
- плюсом до безкоштовності є досить широкий функціонал програми, який не обмежується одним написанням коду та доповненнями, такими як автозаповнення операторів, закриттям тегів, подвійними лапками тощо;
- різні доповнення для підвищення продуктивності написання коду, наприклад готові макети фрагментів (Snippet). Крім стандартного набору, можна створити свою частину готового коду, задавши для нього унікальне ім'я і гарячі клавіші;
- досить зрозумілий інтерфейс, який навіть англійською (русифікатор важко знайти) сприймається користувачем на інтуїтивному рівні;

- наявність готових шаблонів і прикладів написання програмного коду дуже допоможе початківцям або скоротить час на написання тривіального коду для досвідчених розробників.

Якщо є переваги – є і недоліки, серед яких виділяють:

- досить довго запускається, причому дуже довго в порівнянні з іншими програмами того ж типу. Це пояснюється надлишком розширень і інших доповнень, але кому важлива функціональна частина програми, то цей недолік просто відпадає;

- відсутність русифікатора або просто російської версії для багатьох може здатися істотним недоліком, який позначиться на виборі безкоштовного ПО;

- не завжди працює функція автозаповнення коду (операторів), особливо якщо в тексті присутні кілька елементів з іншої мови.

Eclipse – ще одна інтегрована середовище розробки ПО. Написана на мові Java в 2003 році. Також є кроссплатформенною. За рахунок приєднаних до цього середовища доповнень-модулів (які завантажуються окремо при необхідності) можна створювати програмні продукти більш ніж на п'яти мовах програмування.

Переваги Eclipse:

- постійне оновлення версій середовища розробки;

- підтримка багатьох мов;

- середовище безкоштовне;

- підтримка багатьох мов (враховуючи і російську);

- підтримка багатьох мов програмування (таких як, наприклад, Java, C та C++, PHP, Perl, Python, Cobol);

- середовище має промисловий рівень, є гнучкою – тобто легко налаштовується як під будь-яку платформу, так і під будь-якого користувача;

- використовуючи підхід RCP (Rich Client Platform) Eclipse є інструментом для створення практично будь-якого клієнтського програмного забезпечення.

Серед суттєвих загальних недоліків можна виділити небагато:

- сильно завантажує оперативну пам'ять комп'ютера;
- довго запускається, однак, якщо комп'ютер досить потужний – дана проблема легко вирішується.

Microsoft Visual Studio Code – це IDE, яке допомагає створювати додатки, що володіють інноваційним інтерфейсом і функціональними можливостями. Це інтегроване середовище спрощує виконання складних завдань, що дозволяє приділяти більше часу і сил на досягнення своїх безпосередніх цілей. Visual Studio є потужною інтегрованою середовищем розробки на мові JS для Windows. Visual Studio надає засоби з для підтримки мови JS (відкритий вихідний код), в тому числі робочі навантаження JS, обробки і аналізу даних.

З кожною версією інструментів Microsoft намагається врахувати побажання розробників і зробити їх зручнішим для створення додатків практично для будь-якої платформи. Результатом є величезний інтерес і більше 21 мільйонів установок інструменту на сьогоднішній день.

Visual Studio Code має чи то не найбільший перелік можливостей у розробці програмного забезпечення:

- розробка додатків для користувача пристроїв і хмари – повнофункціональні інструменти для створення додатків для Магазину Windows;
- створення додатків для будь-якої платформи – для декількох пристроїв, ПК і веб-додатків, використовуючи хмарні ресурси;
- більш швидка розробка програмного забезпечення – Visual Studio підпискою на MSDN включає Team Foundation Server. Отримайте інструменти, служби і ресурси, щоб забезпечити більш ефективну спільну роботу команди;
- прискорення розробки та тестування за допомогою Microsoft Azure – Microsoft Azure дозволяє розробляти і тестувати програми швидше і дешевше, надаючи можливість хмарного або локального розгортання на вибір. Ця опція не є безкоштовною та доступна лише з підпискою на повний продукт;

- легко розібратися в програмному кодї, створеному командою – Visual Studio Professional пропонує потужний функціонал, який допомагає швидко розібратися у програмному кодї. Функція кодової лупи допомагає зосередитися на роботі, показуючи зміни в кодї, відображаючи автора останніх змін методу, а також результати проходження тестів – і все це безпосередньо в програмному кодї.

У таблиці 3.2 представлена порівняльна характеристика між описаними вище інтегрованими середовищами розробки.

Таблиця 3.2

Порівняння середовищ розробки

	Komodo Edit	Eclipse	Visual Studio Code
Можливість роботи зі синтезаторами мови	-	-	+
Швидкість завантаження	-	-	-
Швидке вбудовування необхідних мов програмування та робота з ними	+/-	-	+
Автозаповнення коду	+/-	+	+
Безкоштовність	+	+	+/-
	2	2	3,5

Отже, з таблиці 3.2 можна зробити висновок, що найвищий бал серед трьох IDE має Visual Studio Code – «3,5». Це середовище є найбільш узагальненим для розробника як з боку мов програмування, так і з боку функціоналу, що дозволяє не обмежувати можливості програмного продукту при його розробці.

3.2 Розробка архітектури та алгоритму роботи аналітичного модуля

Проектування та розробка загальної архітектури – це процес розробки, задача якого перетворити вимоги до системи у вимоги до ПЗ, і побудувати на основі цього архітектуру програмного продукту, яка є структурою системи, її абстракцією елементів на певній фазі роботи. Тобто, перш ніж перейти безпосередньо до реалізації програмного забезпечення, будуються UML-діаграми [22] – діаграми на мові моделювання складних програмних систем. Найчастіше використовується три з діаграм: діаграма класів, використання (прецедентів) та послідовності. Також у дипломній роботі схематично було проілюстровано структуру програми, взаємодію модуля між його компонентами за допомогою структурної схеми компонентів.

Розпочнемо з діаграми використання [23], яка зображена на рисунку 3.1.



Рисунок 3.1 – Діаграма варіантів використання

Діаграма має одного актора – безпосереднього гравця, який, увійшовши на сторінку зі статистикою, яку генерує аналітичний модуль, може здійснювати контроль нею через клавіатуру або мишу. Генерація безпосередньо візуальної інформації відбувається за рахунок включених у процес підпроцесів: робота з базою даних персонажів, обробка GPU, формування графіків активності, формування таблиць.

Діаграма використання (прецедентів) візуально зображає різноманітні сценарії взаємодії між акторами (користувачами) і прецедентами (випадками використання); описує функціональні аспекти системи (бізнес логіку). Діаграма послідовності відповідає за моделювання синхронних процесів у часі, які описують взаємодію об'єктів. На діаграмі зображуються лише ті об'єкти, які безпосередньо беруть участь у взаємодії.

Взаємодія об'єктів модуля представлено на діаграмі послідовності [24] на рисунку 3.2.

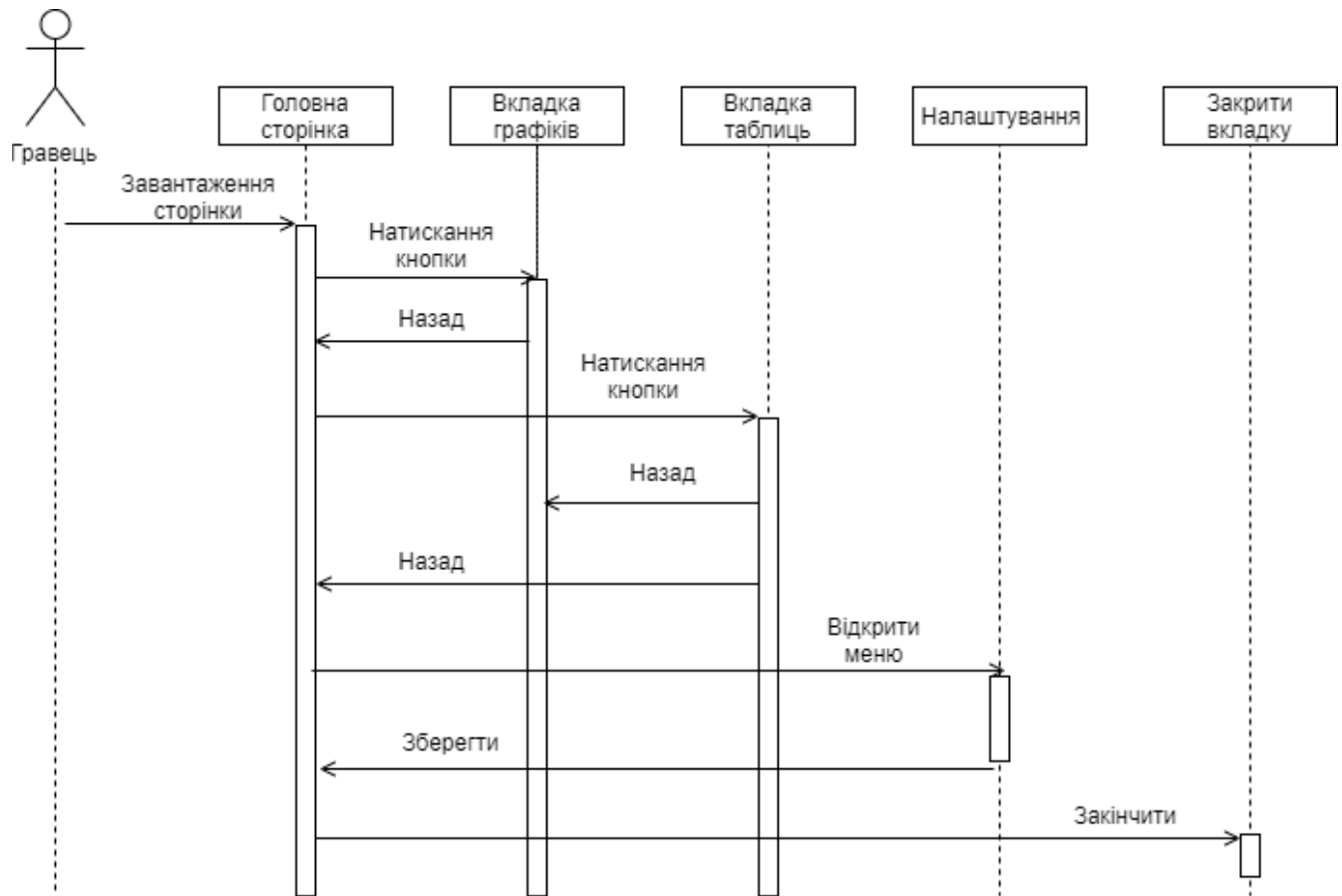


Рисунок 3.2 – Діаграма послідовності

Об'єктами взаємодії є «Гравець», «Головна сторінка», «Вкладка графіків», «Вкладка таблиць», «Налаштування» та «Вихід». Продемонстровано послідовність роботи користувача в часі. При вході на сторінку статистики гравець може залишитися на ній, щоб ознайомитись із загальним рейтингом активності, або перейти на одну з вкладок окремих метрик з графіками або таблицями. З активного вікна гравець може перейти у вікно налаштування, змінити параметри та зберегти нові дані, не перезавантажуючи сторінки. Після завершення ознайомлення з даними здійснюється закриття сторінки.

3.3 Розробка інтерфейсу аналітичного модуля, структура та опис основних елементів форуму для проведення гри

Будь-який форум для рольової гри має конкретну структуру та перелік необхідних (базових) елементів, які необхідні гравцю для зручного користування ресурсом.

Графічна схема структури форуму приведена на рисунку 3.3.

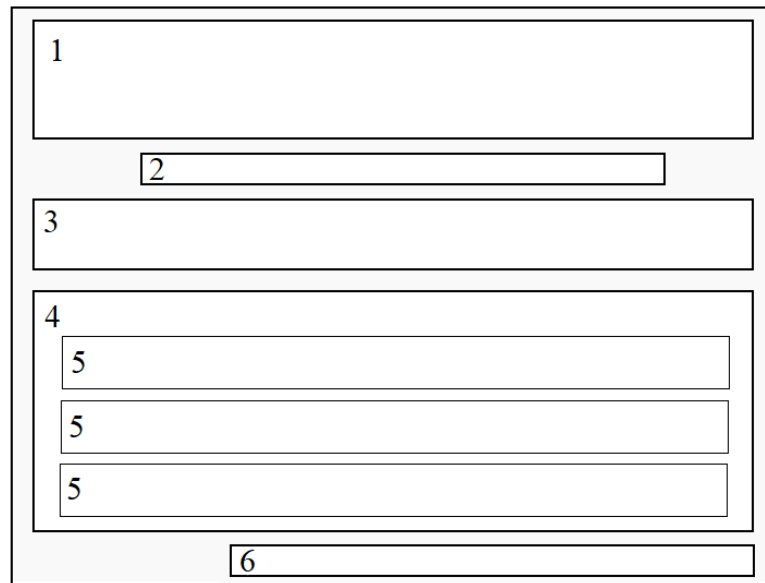


Рисунок 3.3 – Графічна схема головної сторінки форуму для рольових ігор

Форум має такі основні секції:

1: Баннер. Зазвичай, банер містить лише тематичне зображення та назву рольової гри, але якщо форумом займається досвідчений програміст, на баннер можна додати додаткові інтерактивні кнопки, випадаючі меню, анімацію, лічильники часу та усе те, що стосується тематики спільноти.

2: Панель кнопок швидкого доступу. При бажанні вміст можна змінювати, але по замовчуванню їх перелік такий:

- на головну;
- учасники;

- нагороди – дошка пошани активних гравців, але у нашому випадку там буде розміщено сторінку статистики;

- пошук;

- сюжет у рамках якого діють гравці, може бути оригінальним або по мотивам вже існуючих світів, частіше створюється на основі вже існуючих книг, серіалів, фільмів, мультфільмів, з яких гравці беруть канонічних персонажів та відповідають їх характеру;

- ЛОР – усі знання про світ конкретної рольової, її особливості, відомості про раси чи зброю, про часові рамки, відомості, які повинні бути відомі новим для світу гравцям;

- реєстрація;

- увійти;

- активні теми.

3: Панель останніх оновлень, новин, імена адміністрації та модераторів, інформація про ведення гри (наприклад, яка пора дня у грі відповідає дню тижня у реальному житті);

4: Головний блок тем форуму.

5: Категорії тем форуму (структурна схема зображена на рисунку 3.4).

6: Футер. Містить невелику статистику по користувачам (не по персонажам, яка реалізована у статистичному модулі), та чат для спілкування IRT (у реальному часі). Статистика містить наступні дані: кількість тем, кількість повідомлень, скільки всього зареєстрованих гравців, хто зареєструвався останнім, кількість активних у поточний час (гостей, зареєстрованих гостей, рекорд активних на сайті гравців за увесь час, дата встановлення рекорду).

Повернемося до пункту №5. Теми можуть поділятися або не поділятися на секції/категорії за різними критеріями:

- за активністю (як часто там пишуть);

- за актуальністю (якщо до гравців потрібно донести важливу інформацію та закріпити її зверху усіх тем);
- за кількістю коментарів;
- за кількістю переглядів.

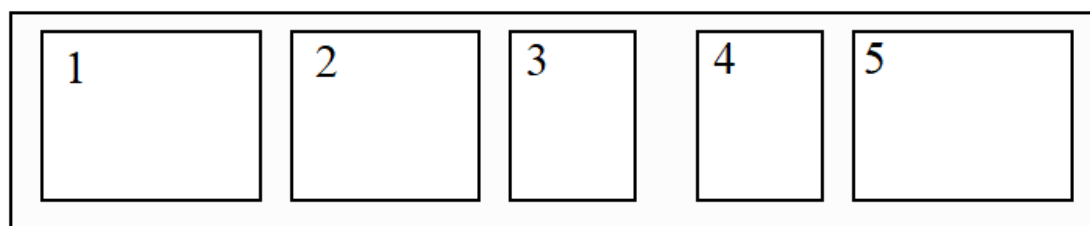


Рисунок 3.4 – Графічна схема інтерфейсу теми форуму на головній сторінці

1: Містить таку інформацію – назва теми, опис теми, перелік модераторів, які відповідають за тему форуму.

2: Опціональне зображення. Зручно оформити його у вигляді надпису Стара/Нова – так гравець буде знати, які теми зараз актуальні. Надпис «Стара тема» відображається, якщо усередині категорії діяльність була відсутня більше місяця.

3: Кількість тем усередині категорії.

4: Кількість переглядів.

5: Інформація про назву теми, у яку додали останній коментар, і який саме гравець його додав, дату та час коментування.

Варто приділити увагу тому, з чого складається коментар на форумі, так як він має деякі необхідні елементи. Графічну схему запропоновано на рисунку 3.5.

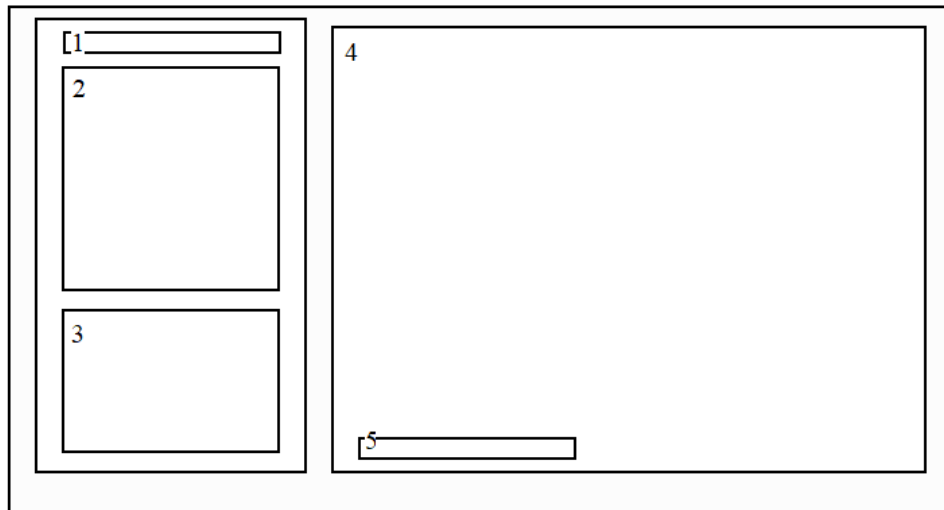


Рисунок 3.5 – Графічна схема коментаря на форумі для рольових ігор

1: Ім'я персонажа.

2: Аватар персонажа. Аватар – це зображення його зовнішності для зручного асоціативності під час гри.

3: Невеликий список даних про гравця, який володіє персонажем. Містить кількість позитивних та негативних оцінок, дату реєстрації, кількість повідомлень на форумі всього, скільки часу провів на форумі, останній візит.

4: Область для коментаря. Сюди гравець поміщує безпосередньо ігровий пост.

5: Тег. Теги потрібні для того, щоб знаходити епізоди або пости конкретного гравця або за конкретною темою.

Тепер, перейдемо безпосередньо до аналітичного модуля. Маючи справу з візуалізацією даних та взагалі з веб-технологіями, варто пам'ятати, що зовнішній вигляд та юзабіліті [25] мають велике значення для ефективної комунікації між гравцем та системою. Незайвим буде зазначити знову те, що у спільноті форумних рольових ігор знаходяться невимогливі люди, які вважають, що зручніше підраховувати статистику вручну, а отже переконати їх в зручності використання аналітичного модуля – це необхідність.

Як ми бачили вище, будова форуму доволі громіздка, тому доречніше розглядати лише головну сторінку аналітики. Структурна схема інтерфейсу реалізованого модуля продемонстрована на рисунку 3.6.

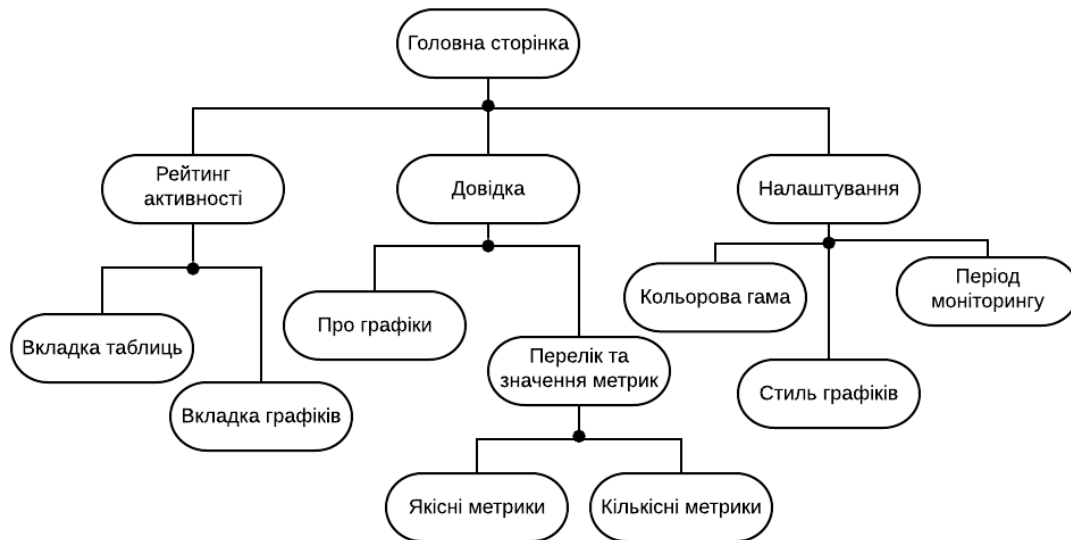


Рисунок 3.6 – Структурна схема інтерфейсу модуля

Зручна навігації – це, по-перше, лаконічність опцій, але достатньо повна функціональність для інформативності. У випадку аналітичного модуля необхідно розробити зручне вікно налаштування для швидкої та однозначної зміни бажаних параметрів вигляду графіків. До того ж, усе це має відбуватися на одній сторінці, у чому допоможе динаміка мови JS. Перемикаючись між вкладками, сама сторінка не оновлюється, що дозволяє підвищити швидкодію, особливо враховуючи те, що багато хто з гравців може користуватися мобільно браузерною версією форуму та мати повільний мобільний інтернет. Це заощаджує трафік.

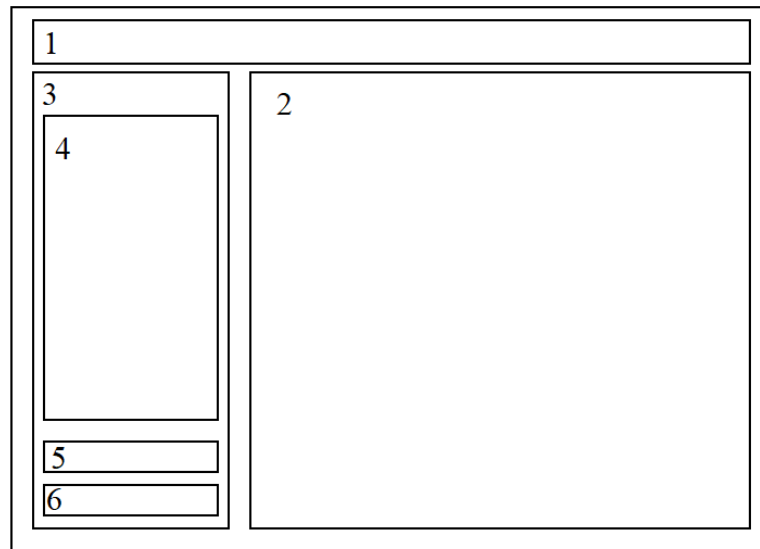


Рисунок 3.7 – Графічна схема інтерфейсу аналітичного модуля

1: Заголовок. Міститиме дані про поточний період моніторингу, дату та час останнього оновлення, таймер до кінця періоду моніторингу.

2: Головне активне вікно, у якому відображатимуться графіки активності.

3: Панель меню. Містить список доступних графіків. Меню має дві головні категорії: статистика для модераторів та статистика для гравців. У категорій є декілька спільних графіків, але у модераторів статистичних даних, що потрібні їм для роботи, дещо більше.

4: Елементи панелі меню.

5: Кнопка налаштувань.

6: Довідка, яка містить описи графіків, трактування їх абсцис та ординат, ліміти значень, метрики, норми метрик та інші початкові дані, які вказується для рольової модераторами.

3.4 Розробка та опис основних графіків

Для того, щоб репрезентувати інформацію, замало одного типу графіків. У період, коли візуалізація даних використовується майже всюди, винайшли безліч видів графіків, які відрізняються між собою виглядом, призначенням, .

Існує кілька різних типів діаграм і графіків. Чотири найбільш поширені:

- лінійні графіки – показують, як змінилися числа з часом;
- стовпчикові гістограми – показують числа, незалежні один від одного;
- кругові діаграми – щоб показати, як ціле ділиться на різні частини;
- декартові графіки – мають числа на обох осях, що дозволяє показати, як зміни однієї речі впливають на іншу [26].

З технологією WebGL та мовою програмування Java Script можна створити набагато більше видів, що базуються на основних графіках, перерахованих вище. Розглянемо створення графіків для візуалізації деяких метрик. Спочатку візьмемо до уваги графіки, які переважно використовуватимуться модератором.

1. Аналіз середнього рейтингу гравців за 2019 зображено на рисунку 3.8.

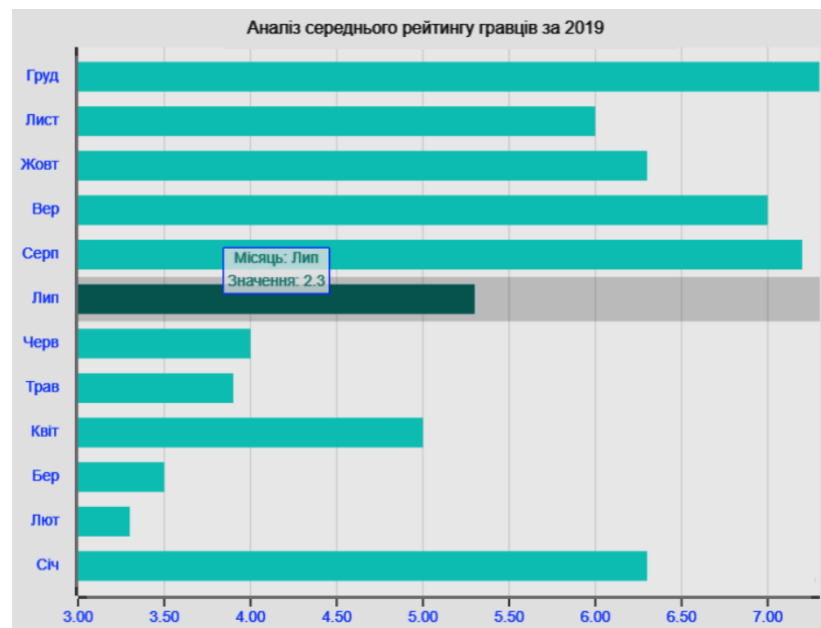


Рисунок 3.8 – Стовпчастий графік аналізу середнього рейтингу гравців

Він допоможе модераторам встановити, за який моніторинговий період падало середнє значення рейтингу – показник того, що гравець втрачає зацікавленість. На графіку-прикладі таким місяцем став лютий.

Фрагмент коду, що описує ініціалізацію графіку, наведено на рисунку 3.9.

```
// Визначте інтерфейсу для створення горизонтальних стовбців.
let barChart
{
  barChart = (options) => {
    // Червоний стиль заливки
    const flatRedStyle = new SolidFill().setColor(ColorRGBA(242, 67, 79))
    let y = 0
    const figureThickness = 10
    const figureGap = figureThickness * .5
    const bars = []

    // Створюється XY-діаграму та додайте до неї RectSeries для візуалізації
    //прямокутників.
    const chart = lc.ChartXY(options)
    .setTitle('Аналіз середнього рейтингу гравців за 2019 рік')
    .setAutoCursorMode(AutoCursorModes.onHover)
    // Представлення у вигляді прямокутників
    const rectangles = chart.addRectangleSeries()
```

Рисунок 3.9 – Фрагмент коду ініціалізації графіку горизонтальних стовбців

Наступний графік – так званий коробковий графік з вусами [27]. Він є зручним способом візуального відображення розподілу даних через їх квантілі. Фрагмент коду його ініціалізації зображено на рисунку 3.10.

```
var chart = d3.box()
  .whiskers(iqr(1.5))
  .height(height)
  .domain([min, max])
  .showLabels(labels);

var svg = d3.select("body").append("svg")
  .attr("width", width + margin.left + margin.right)
  .attr("height", height + margin.top + margin.bottom)
  .attr("class", "box")
  .append("g")
  .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
```

Рисунок 3.10 – Фрагмент коду ініціалізації коробкового графіку з вусами

Лінії, що проходять паралельно від коробок, відомі як «вуса», які використовуються для позначення мінливості поза верхніми та нижніми квантілями. На нашому графіку вони будуть зображені, як окремі крапки, які

узгоджуються з вусами. Поля можна намалювати вертикально або горизонтально (в роботі вони вертикально).

Хоча графічні коробки можуть здаватися примітивними порівняно з гистограмою або графіком щільності, вони мають перевагу зайняти менше місця, що корисно при порівнянні розподілів між багатьма групами або наборами даних.

Коробковий графік для аналізу норми кількості слів у пості (рис. 3.11) дозволить зрозуміти, як щільно розташовані між собою дані та допоможе модераторам визначити значення метрик word_{\min} , word_{\max} , $\text{word}_{\text{norm}}$.

У цьому випадку, медіана коробкового графіку і буде значенням $\text{word}_{\text{norm}}$.

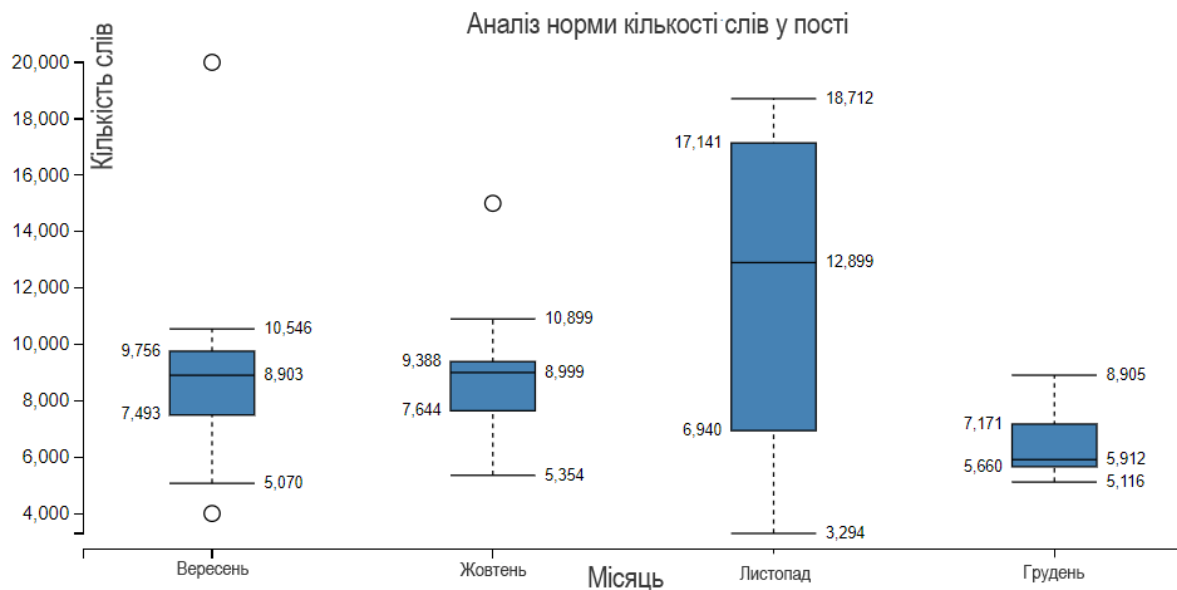


Рисунок 3.11 – Коробковий графік для аналізу норми кількості слів у пості

Наступний графік – звичайний лінійний тип. Для побудови лінійних графіків нам необхідні координатні осі. І оскільки лінійні графіки не уявляються без осей, то в бібліотеці `d3.js` є спеціальний метод `d3.svg.axis()`, який дозволяє будувати їх (рисунок 3.12).

По-перше, необхідно створити об'єкт `svg`, який буде укладати вісь.

Вся логіка зі створення осі X розміщується в функції createAxes і розбивається на кілька етапів.

По-друге, треба приблизно усвідомлювати розкид значень даних, які повинні бути відображені на графіку. У нашому випадку діапазоном таких значень є кількість місяців у вересні (вісь X) та число можливих відвідувань.

```
// Створюємо вісь X
var xAxis = d3.svg.axis()
    .scale(scaleX)
    .orient("bottom");
// Створюємо вісь Y
var yAxis = d3.svg.axis()
    .scale(scaleY)
    .orient("left");

// Малювання вісі X
svg.append("g")
    .attr("class", "x-axis")
    .attr("transform", // сдвиг осі вниз и вправо
        "translate(" + margin + ", " + (height - margin) + ")")
    .call(xAxis);

// Малювання вісі Y
svg.append("g")
    .attr("class", "y-axis")
    .attr("transform", // сдвиг осі вниз и вправо на margin
        "translate(" + margin + ", " + margin + ")")
    .call(yAxis);
```

Рисунок 3.12 – Фрагмент коду створення осей лінійного графіку

З урахуванням того, що на нашому форумі кількість гравців – 50, робимо розмах на те, що кожен з гравців запропонує приєднатися ще трьом особам. Тобто, вісь Y буде містити 135 пікселя, значення width (ширина об'єкта svg) мінус відступи справа і зліва по 40 пікселів. І для зіставлення значень задаємо функцію лінійної інтерполяції (рисунок 3.13).

Щоб створити вісь, потрібна функція інтерполяції scale для зіставлення значень даних з екранними координатами. Так як осі можуть бути горизонтальними і вертикальними, то треба задати орієнтацію осі за допомогою функції orient ().

У контейнері `svg` осі повинні розміщуватися всередині елемента `g`. За допомогою атрибута `transform`, якому в якості значення задається `translate`. А функція `call ()` передає і відмальовує в елементі `g` вісь.

```
// ф-я інтерполяції для вісі X
var scaleX = d3.scale.linear()
    .domain([0, 30])
    .range([0, xAxisLength]);

// ф-я інтерполяції для вісі Y
var scaleY = d3.scale.linear()
    .domain([300, 0])
    .range([0, yAxisLength]);

for(i=0; i<rawData.length; i++)
    data.push({x: scaleX(rawData[i].x)+margin, y: scaleY(rawData[i].y) + margin});
```

Рисунок 3.13 – Фрагмент коду реалізації функцій лінійної інтерполяції

Кінцевий вигляд графіку динаміки відвідувань за вересень місяць зображено на рисунку 3.14.



Рисунок 3.14 – Лінійний графік динаміки відвідувань гравців

Починаючи з відкриття форуму, на початку кількість гравців невелика. Але з початком квесту або іншого промоушену, вона зростає і досягає 300

користувачів 17-го вересня. Графік використовуватиметься переважно модераторами.

Останній графік, який варто розглянути – точкова гістограма або `d3.bin`. Він групує точки даних у контейнери (у нашому випадку – це місяці) однакової ширини вздовж осі порівняння.

Ці контейнери можуть бути представлені гістограмою (діаграма частоти), яка показує їх у вигляді смуг, довжиною пропорційною кількості елементів кожного контейнера.

Фрагмент коду функції, що маніпулює даними у контейнері, представлено на рисунку 3.15.

```
{
  const values = Array.from({ length: 100 }, (d, i) => ({
    label: i,
    p: Math.random()
  })),
  bin = d3.bin().value(d => d.p);
  return bin(values);
}
```

Рисунок 3.15 – Фрагмент метод доступу `bin.value`

Елементи нашого контейнера – це кількість завершених епізодів в місяць. Графік зображено на рисунку 3.16 та демонструє динаміку завершених гравцями епізодів у місяць.

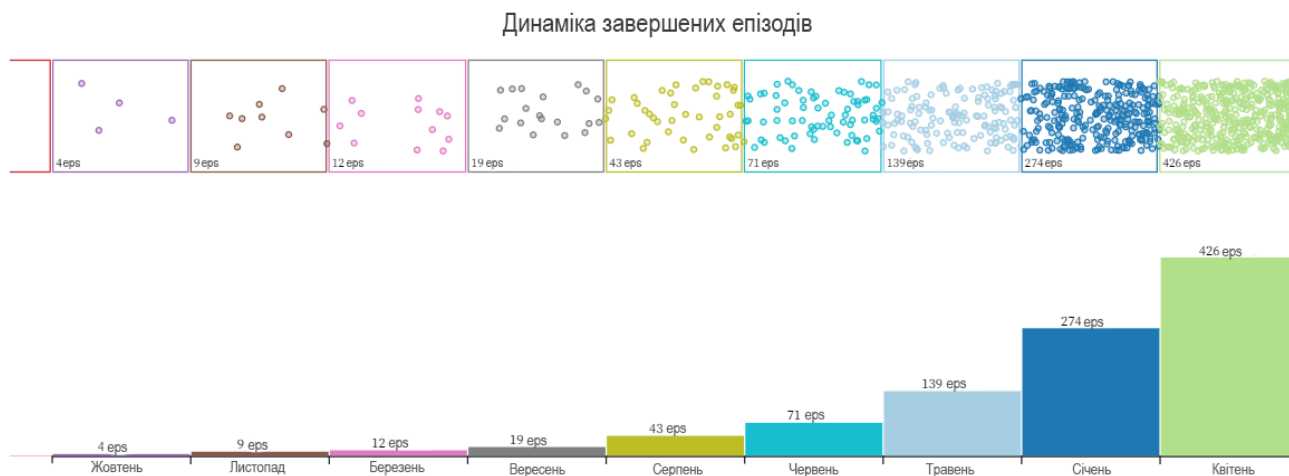


Рисунок 3.16 – Точкова гістограма із зображенням динаміки завершених епізодів

Як ми бачимо, найбільшу кількість епізодів закрили у квітні. Зумовлено це тим, що зазвичай у цей період влаштовуються квести, а більшість гравців вільна від навчання та менше навантажена на роботі.

Отже, у підрозділі 3.4 було викладено створення основних типів графіків, що фігуруватимуть у аналітичному модулі, а саме: стовбчастий графік зі статистикою рейтингу, так званий «коробковий графік з вусами» для контролю норми кількості слів у пості, звичайна лінійна гістограма для моніторингу відвідувань у моніторинговий період та точкова гістограма-контейнер із зображенням динаміки завершених епіходів.

3.5 Висновок

Третій розділ магістерської дипломної роботи було присвячено розробці реалізації удосконаленого методу обчислення рейтингу гравців у вигляді незалежного аналітичного модуля.

Проаналізовано та обгрунтовано вибір засобів реалізації: мову програмування та середовище розробки. Варіантний аналіз продемонстрував, що найдоцільніше для веб-системи використовувати мову програмування Java Script із допомогою додаткових бібліотек для побудови графіків та схем.

Була розроблена архітектура та алгоритм роботи модуля, що проілюстровано у вигляді діаграм варіантів використання, послідовності та графічних схем елементів форуму та майбутнього модуля.

4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО МОДУЛЯ

4.1 Обґрунтування вибору методики тестування системи

Тестування ПЗ – процес перевірки між реальним і очікуваним результатом роботи програмної системи, що відбувається за допомогою деякого обраного переліку тестів. У ширшому розумінні, тестування – це техніка контролювання якості ПЗ. Ця техніка включає в планування робіт або Test Management, проектування або Test Design, виконання або Test Execution, та аналіз отриманих результатів або Test Analysis.

Існує безліч підходів до тестування, видів, його градацій (рівнів), тому у сфері розробки програмного забезпечення, під час розділення обов'язків, на тестування часто виділяється ціла команда або щонайменше не зайнята іншими обов'язками людина-спеціаліст [28].

Як підхід тестування для поточного продукту було обрано тестовий випадок або Test Case – це артефакт, що описує сукупність кроків, конкретних умов і параметрів, необхідних для перевірки реалізації функції, що тестується, або її частини. Тобто, у подальшому ми оберемо для тестування найскладніші та найбільш значущі функції аналітичного модуля та перевіримо, чи очікуваний результат їх виконання відповідає дійсності.

Наступний крок – обрати вид/тип тестування. Серед основних виділяють:

1. Functional testing (функціональне тестування).

Функціональне тестування – це тестування функціональності і поведінки ПЗ, для того щоб переконатися що поведінка програми та її функціонал відповідає вимогам складеної на перших етапах створення специфікації. Зазвичай виконується як тестування «чорного ящика» [29], подаючи на вхід якийсь набір даних і чекаючи чогось на виході.

Тестування «чорного ящика» – це тестування, при якому тестер має доступ до ПЗ тільки через ті ж інтерфейси, що й замовник або користувач. Як правило,

тестування чорного ящика ведеться з використанням специфікацій чи інших документів, що описують вимоги до системи. В даному виді тестування критерій покриття складається з покриття структури вхідних даних, покриття вимог і покриття моделі (у тестуванні на основі моделей).

2. Smoke testing (димове тестування).

Тестування основних модулів додатка з метою визначення чи придатна програма для релізу нової версії. Зазвичай, проводиться перед великим релізом нової версії додатка, який включає в себе нові модулі. Проводиться для переконання що нові модулі справно працюють і що старі модулі і функціональність теж не вийшли з ладу.

3. Security testing(тестування безпеки).

Перевірка прав доступу, неможливість авторизації для сторонніх осіб, SQL ін'єкції, XSS атаки. Перевірка що додаток відповідає необхідним вимогам до безпеки.

4. Stress//load testing (стресове тестування/тестування навантаження).

Це тестування програми або окремого модуля на межі або вище його можливостей. Проводиться для визначення цієї межі і поведінку при переході цієї межі. Отримати збій і проаналізувати, чи достатньо цього кордону на поточний момент розвитку проекту додатка.

5. Usability testing (тестування юзабіліті).

Тестування того, як швидко користувач звикне до інтерфейсу програмного продукту. Проводиться для оптимізація, поліпшення і спрощення інтерфейсу до інтуїтивно зрозумілого. Сюди можуть входити підказки, хелпери, зручний дизайн елементів, мінімізація кроків проходження для доступу до функції і т. д.

Було прийнято рішення, що доцільно використати функціональний вид тестування, до якого входять:

- функціональне тестування;
- тестування користувацького інтерфейсу;

- тестування безпеки;
- тестування взаємодії.

Функціональне тестування розглядає заздалегідь вказану поведінку і ґрунтується на аналізі специфікацій функціональності компонента або системи в цілому.

Тестування користувацького інтерфейсу (GUI Testing) – функціональна перевірка інтерфейсу на відповідність вимогам – розмір, шрифт, колір, consistent behavior (незмінна поведінка).

Тестування безпеки – це стратегія тестування, яка використовується для перевірки безпеки системи, а також для аналізу ризиків, пов'язаних із забезпеченням цілісного підходу до захисту додатку, атак хакерів, вірусів, несанкціонованого доступу до конфіденційних даних.

Тестування взаємодії (Interoperability Testing) – це функціональне тестування, що перевіряє здатність додатку взаємодіяти з одним і більше компонентами або системами і включає в себе тестування сумісності (compatibility testing) і інтеграційне тестування

Що стосується рівнів тестування, варто обрати модульне тестування, оскільки програмна система має відповідну об'ємну модульну структуру. Компонентний (модульне) тестування перевіряє функціональність і шукає дефекти в частинах додатку, які доступні і можуть бути протестовані окремо (модулі програм, об'єкти, класи, функції і т.д.). Велика кількість модулів робить неможливим повне тестування системи.

Після модульного тестування використовується інтеграційне, яке впроваджено для тестування взаємодії між модулями системи.

4.2 Проведення тестування

Для того, щоб успішно виконати тестування аналітичного модуля, необхідно переконатися, першочергово, що його функціональність виконує поставлену на

початку мету, а саме: підвищення ефективності проведення форумної ролівої гри за рахунок автоматизованого та більш точного підрахунку рейтингу активності гравців на основі розширеного діапазону як кількісних, так і якісних метрик активності.

Спочатку варто перевірити, чи відповідає фінальний вигляд інтерфейсу його графічній схемі. Інтерфейс аналітичного модуля представлений на рисунку 4.1.

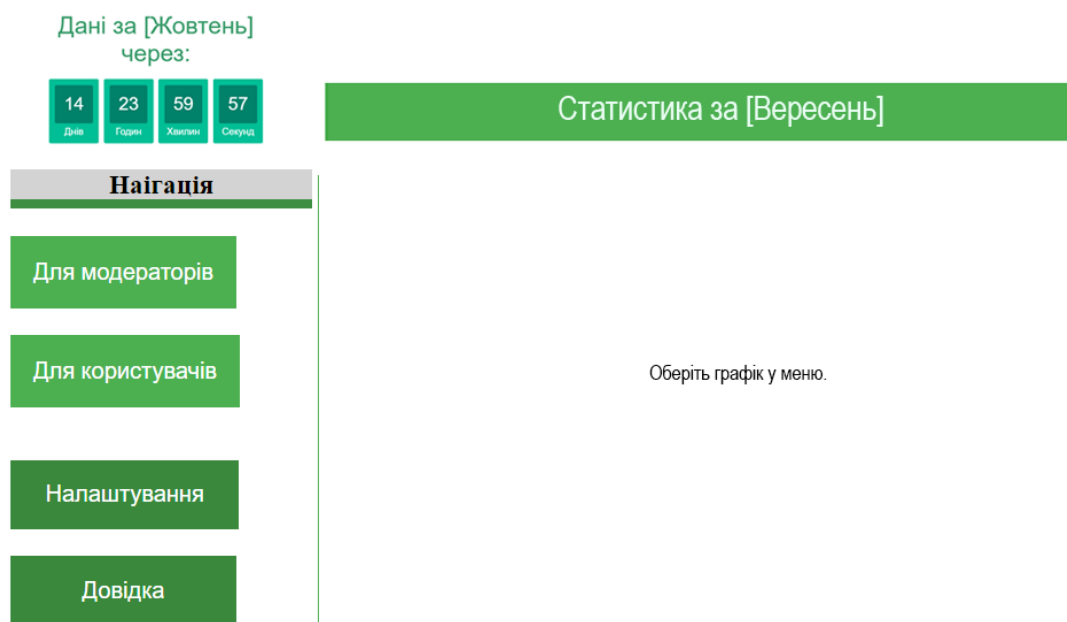


Рисунок 4.1 – Інтерфейс аналітичного модуля.

Для того, щоб почати з ним роботу, потрібно обрати один з графіків у меню. Для модераторів і користувачів графіки різні.

Інтерфейс містить заголовок, який складається з таймера до оновлення з останнім числом нового місяця, тобто через 14 днів, коли закінчиться жовтень, а аналітичний модуль накопичить всю необхідну статистику, таймер почне відраховувати з нового місяця.

Головне активне вікно поки що не містить нічого, тому що не обрано жожного графіка.

Панель меню складається з чотирьох кнопок. Якщо навести на одну з трьох (окрім кнопки Налаштування), з'явиться випадаючий список (у випадку кнопок «Для модераторів» та «Для користувачів») або динамічне вікно підказки (у випадку «Довідка» і тільки якщо вже обрано графік, щоб надати по ньому інформацію).

Якщо порівнювати готовий інтерфейс із готовою схемою, можна побачити, що один пункт не відповідає дійсності, а саме – час останнього оновлення. Справа в тім, що час останнього оновлення для деяких графіків різний, тому щоб побачити дані про останні оновлення, потрібно обрати графік збоку у випадаючому меню (рисунок 4.2).

Очікуваний результат при виборі пункту меню з назвою графіку: побачити цей графік у головному активному вікні.

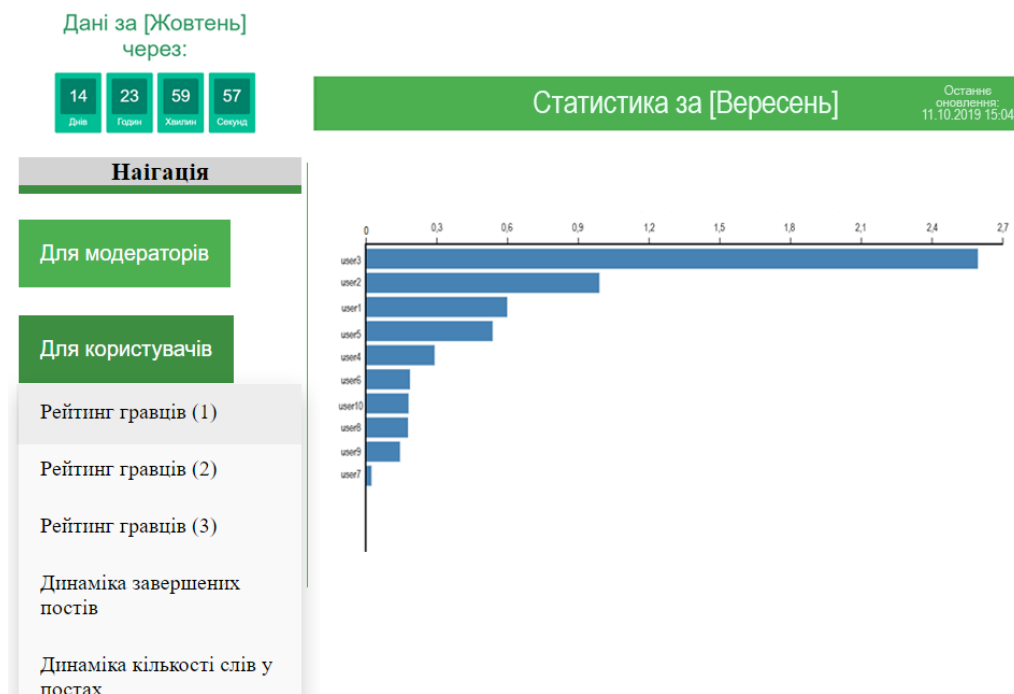


Рисунок 4.2 – Вибір графіка у меню

Рейтинг гравців (1), (2) та (3), це рейтинги за кількістю персонажів, щоб кожен гравець змагався за перше місце у таблиці з тими, у кого приблизно таке саме навантаження.

У графіка немає надпису, але якщо навести курсор миші на довідку, то очікуваним результатом буде впливаюче вікно підказки (рисунок 4.3).

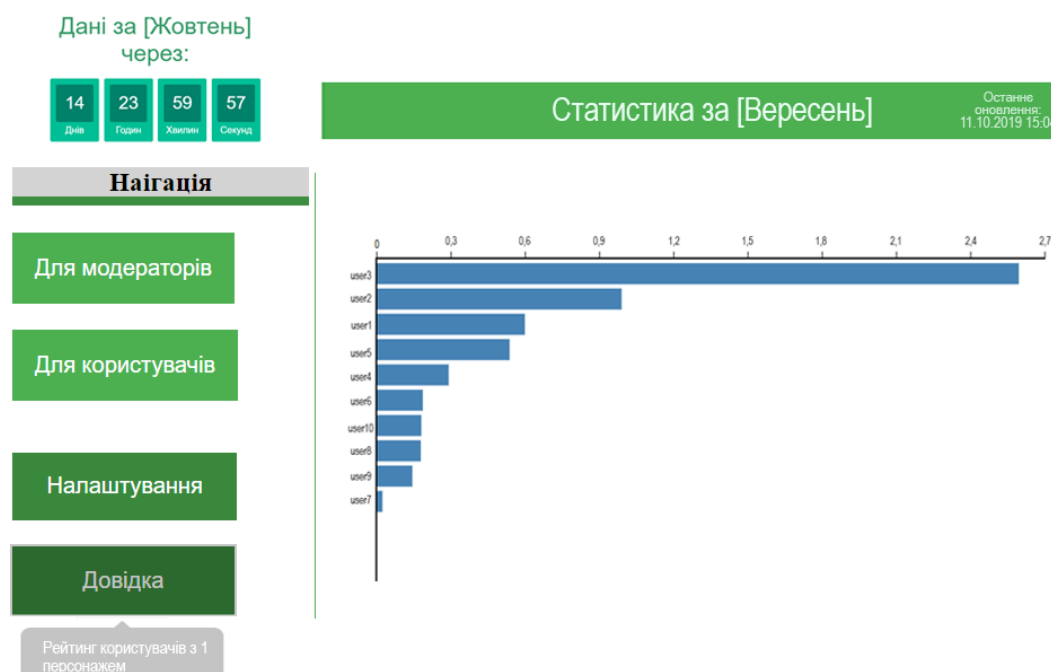


Рисунок 4.3 – Випадаюче меню підказки кнопки Довідки.

Зі створенням аналітичного модуля та проведенням тестування на його функціональність, доведено, що поставленні на початку задачі дослідження було виконано у повній мірі. За допомогою розширеної системи показників аналітичний модуль вирішує проблему низької ефективності роботи модераторів, а отже і форумної гри, шляхом створення методу обчислення рейтингу гравців та критично важливих кількісних та якісних характеристик.

У середньому, модератор витрачає від 2-х до 3,5-х години (в залежності від кількості персонажів на форумі) на те, щоб власноруч створити, наприклад,

таблицю не завершених гравцями епізодів або визначити, як довго ніхто не писав у епізод. Аналітичний модуль збирає дані та формує з них графік або таблицю для швидкого орієнтування як з боку модераторів, так і з боку користувачів, економлячи час та знижуючи навантаження на 55% за рахунок введення автоматизації.

4.3 Висновки

У четвертому розділі були визначені, проаналізовані та описані технології, типи та рівні тестування, що використовувались для валідації програмного продукту – програмного модуля, що реалізувався як результат запропонованого методу дипломної роботи.

Як підхід тестування було обрано тестовий випадок або Test Case. Як тип було доцільно використати функціональний тип тестування – «чорний ящик». ПЗ тестувалося з боку користувача, без втручання вихідного коду програми. Була продемонстрована робота функцій синтезу тексту та інших елементів на екрані, таких як: кнопки, таблиці, генерація графіків.

При тестуванні аналітичного модуля помилок та невідповідності специфікації виявлено не було. Модуль був розроблений на мові JavaScript з використанням бібліотеки для побудови графіків – D3.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Оцінювання комерційного потенціалу розроблюваного модуля

Результатом магістерської кваліфікаційної роботи «Розробка методів та програмних засобів для підвищення ефективності проведення форумних рольових ігор» є розробка аналітичного модуля, що опрацьовує ігрову статистику гравців. Для проведення технологічного аудиту залучено двох незалежних експертів: Романюк Оксана Володимирівна (к.т.н., доцент каф. ПЗ), Колодний Володимир Володимирович (к.т.н. доцент каф. КН).

Оцінювання комерційного потенціалу буде здійснене за критеріями, що наведені в таблиці 5.1. Оцінка комерційного потенціалу здійснюється за 11-ма критеріями [30].

Таблиця 5.1

Критерії оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
1	2	3	4	5	6
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Мало аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші	Технічні та споживчі властивості продукту трохи гірші	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі	Технічні та споживчі властивості продукту значно кращі

Продовження таблиці 5.1.

1	2	3	4	5	6
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
11	Необхідна розробка регламентних документів	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Процедура отримання дозвільних документів для виробництва та реалізації продукту	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами зведено в таблицю 5.2.

Таблиця 5.2

Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Романюк	2. Колодний
	Бали, виставлені експертами:	
1	4	4
Ринкові переваги (недоліки)		
2	4	3
3	3	4
4	4	3
5	3	3
Ринкові перспективи		
6	4	4
7	4	3
Практична здійсненність		
8	3	4
9	4	3
10	4	4
11	3	3
Сума балів	СБ ₁ = 46	СБ ₂ = 44
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 45$	

За даними таблиці 5.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Рівень комерційного потенціалу розробки, становить 45 балів, що відповідає рівню «високий».

5.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.

Проведемо прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи для розробки програмного забезпечення.

Виконаємо розрахунок витрат приймаючи до уваги те, що розробкою займався один розробник програмного забезпечення. Основна заробітна розробника-дослідника Z_o , що обраховується за формулою 5.1.

$$Z_o = \frac{M}{T_p} \cdot t \text{ (грн.)} \quad (5.1)$$

де

M – місячний посадовий оклад – 4175 грн;

T_p – число робочих днів в місяці; $\sim T_p = 22$ дні;

t – число робочих днів роботи розробника-дослідника - 55.

$$Z_o = \frac{4175}{22} \cdot 55 = 10\,437 \text{ (грн.)}$$

Додаткова заробітна плата Z_d розробника розраховується як 12% від основної заробітної плати:

$$Z_d = 0,12 \cdot 10\,437 = 1\,252 \text{ (грн.)}$$

Нарахування на заробітну плату $H_{зп}$ розробника розраховується за ф. 5.2.

$$H_{зп} = (Z_o + Z_d) \cdot \frac{\beta}{100} \text{ (грн.)} \quad (5.2)$$

де Z_o – основна заробітна плата розробника;

Z_d – додаткова заробітна плата розробника;

β – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування – 22%.

$$H_{зп} = (10\,437 + 264) \cdot 0,22 = 2354,22 \text{ (грн.)}$$

Амортизація обладнання, комп’ютерів та приміщень, які використовувалися під час виконання даного етапу роботи. Розраховується за формулою 5.3.

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12}, \quad (5.3)$$

де

$Ц$ – загальна балансова вартість обладнання, приміщення тощо, грн;

H_a – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

T – фактична тривалість використання, міс ($T = 6$ міс.);

Розробка програмного забезпечення проводилася протягом 6 місяців. Зроблені розрахунки зведено до таблиці 5.3.

Таблиця 5.3

Амортизаційні відрахування

Найменування	Балансова вартість, грн	Фактична трив. використання, міс.	Величина амортизаційних відрахувань, грн
Офісне приміщення	100000	6	2666,60
Ноутбук	18700	6	935,00
Всього			3601,60

Інформацію про матеріали, що використовуються при виготовленні даного продукту внесено до таблиці 5.6.

Таблиця 5.5

Матеріали, що використовуються при виготовленні даного продукту

Найменування матеріалу	Ціна за одиницю, грн	Витрачено, шт.	Вартість витраченого матеріалу, грн
Папір 1(пачка)	82,00	2	164,00
Ручка	6,00	1	6,00
Флешка	300	1	300,00
Всього	470,00		

Витрати на силову електроенергію розраховуються за формулою 5.4.

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi} \quad (5.4)$$

V – вартість 1кВт-години електроенергії ($V=1,56$ грн/кВт);

Π – установлена потужність комп'ютера ($\Pi=0,8$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=440$ год.);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,5$).

$$V_e = 1,56 \cdot 0,8 \cdot 440 \cdot 0,5 = 274,56 \text{ (грн.)}$$

Також потрібно врахувати витрати на доступ до мережі Інтернет, що використовувався під час виконання роботи.

Витрати за доступ до Інтернет можна розрахувати за формулою 5.5.

$$V_{\text{ді}} = C_{\text{ді}} \cdot T \text{ (грн.)} \quad (5.5)$$

де

$C_{\text{ді}}$ – це ціна доступу за місяць;

T – кількість місяців використання доступу до мережі.

Отже, витрати на доступ до мережі Інтернет становлять:

$$V_{дi} = 140 \cdot 2 = 280 \text{ (грн.)}$$

Розрахуємо інші витрати V_{iH} .

Інші витрати I_B можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу.

$$V_{iH} = (1..3) \cdot Z_o . \quad (5.6)$$

Отже, розрахуємо інші витрати:

$$V_{iH} = 1 \cdot 10\,437 = 10\,437 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$V = Z_o + Z_d + H_{зп} + A + K + V_{дi} + I_B$$

$$V = 10\,437 + 1\,252 + 2354,22 + 280 + 10\,437 = 24\,760,22 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи $V_{заг}$ за формулою 5.7.

$$V_{заг} = \frac{V_{iH}}{\alpha} \quad (5.7)$$

де

α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відносних одиницях = 1.

$$V_{\text{заг}} = 24\,760,22$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою 5.8.

$$V_{\text{заг}} = \frac{V_{\text{ін}}}{\beta} \quad (5.8)$$

де

β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Так, якщо розробка знаходиться:

- на стадії науково-дослідних робіт, то $\beta \approx 0,1$;
- на стадії технічного проектування, то $\beta \approx 0,2$;
- на стадії розробки конструкторської документації, то $\beta \approx 0,3$;
- на стадії розробки технологій, то $\beta \approx 0,4$;
- на стадії розробки дослідного зразка, то $\beta \approx 0,5$;
- на стадії розробки промислового зразка, $\beta \approx 0,7$;
- на стадії впровадження, то $\beta \approx 0,9$.

Отже, підставимо дані в формулу й отримаємо результат:

$$V_{\text{заг}} = \frac{24\,760,22}{0,9} = 27\,511,3(\text{грн.})$$

5.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спробуємо кількісно спрогнозувати, яку вигоду, можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Зрозуміло, що всі зроблені тут розрахунки будуть приблизними і не передбачають деталізації.

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження, розраховується за формулою 5.9.

$$\Delta\Pi_i = \sum_i^n (\Delta\Pi_o \cdot N + Ц_o \cdot \Delta N) \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right) \text{ (грн.)} \quad (5.9)$$

де

$\Delta\Pi_o$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$Ц_o$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

λ – коефіцієнт, який враховує сплату податку на додану вартість.

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $\rho = 0,25$;

v – ставка податку на прибуток (18%).

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаються на 50 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 50 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 150 користувачів, протягом другого року – на 200 користувачів, протягом третього

року – 100 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 1 користувача, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 600 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta\Pi_1$ протягом першого року складатиме:

$$\Delta\Pi_1 = 50 + (600 + 50) \cdot 1500 = 142\,500 \text{ грн.}$$

Збільшення чистого прибутку підприємства $\Delta\Pi_i$ протягом другого року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta\Pi_2 = 50 + (600 + 50) \cdot (1500 + 200) = 272\,500 \text{ грн.}$$

Збільшення чистого прибутку підприємства $\Delta\Pi_i$ протягом третього року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta\Pi_3 = 50 + (600 + 50) \cdot (1500 + 200 + 100) = 337\,500 \text{ грн.}$$

5.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Основними показниками, які визначають доцільність фінансування наукової розробки інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Припустимо, що загальні витрати ЗВ на виконання та впровадження результатів НДДКР (або теперішня вартість інвестицій PV) дорівнює 5 210,1 грн. Результати вкладених у наукову розробку інвестицій почнуть виявлятися протягом трьох років. У першому році підприємство отримає збільшення чистого прибутку на 142 500 грн відносно базового року, у другому році – збільшення чистого прибутку на 272 500 грн (відносно базового року), у третьому році – збільшення чистого прибутку на 337 500 грн (відносно базового року).

Тоді рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 5.1.



Рисунок 5.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Абсолютна ефективність $E_{абс}$ вкладених інвестицій розраховується за формулою 5.10.

$$E_{абс} = (ПП - PV), \quad (5.10)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн;

PV – теперішня вартість інвестицій $PV = 3B$, грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою 5.11.

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t} \quad (5.11)$$

де

$\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

Розрахуємо вартість чистих прибутків за формулою 5.11

$$ПП = \frac{5210}{(1+0,1)^0} + \frac{142500}{(1+0,1)^2} + \frac{272500}{(1+0,1)^3} = 266\,923,7 \text{ (грн.)}$$

Тоді розрахуємо $E_{\text{абс}}$:

$$E_{\text{абс}} = 266\,923,7 - 5210 = 261\,713,7$$

Оскільки $E_{\text{абс}} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_B за формулою:

$$E_B = \sqrt[T]{1 + \frac{E_{abc}}{PV}} - 1 \quad (5.12)$$

де

E_{abc} – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $PV = 3B$, грн;

T_j – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_B = \sqrt[3]{1 + \frac{261\,713,7}{5210}} - 1 = 1,71 \text{ або } 171 \%$$

Далі, розраховану величина E_B порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f \quad (5.13)$$

де

d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,05$

Якщо величина $E_B > \tau$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

$$\tau = 0,05 + 0,1 = 0,15$$

Оскільки $E_B = 171\% > \tau_{\min} = 0,15 = 15\%$, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_B} \quad (5.14)$$

$$T_{ок} = \frac{1}{1,71} = 0,36 \text{ років}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

5.5 Висновки

В даному розділі було виконано оцінювання комерційного потенціалу розробки інтелектуальної технології розподілу часу розробки програмних продуктів. Проведено технологічний аудит з залученням двох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки вище середнього. Щорічна ефективність вкладених в наукову розробку інвестицій складає 171%, що вище за мінімальну бар'єрну ставку дисконтування.

$E_{abc} > 0$, а отже вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним. Термін окупності вкладених у реалізацію проекту інвестицій становить 0,36 років.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи був розроблений метод подальшого розвитку для обчислення рейтингу активності гравців форумних рольових ігор.

Основні результати, отримані при виконанні роботи такі:

- Були визначені найбільш суттєві проблеми під час проведення форумної рольової гри для аналізу способів їх вирішення та обґрунтування актуальності цих проблем шляхом визначення відсотку тих проблем, що вирішуються впровадженням автоматизації.

- Проведено аналіз існуючих методів обчислення рейтингу гравця, щоб запропонувати шляхи їх подальшого розвитку та втілити у новий метод. Основою для удосконалення є матричний метод.

- Були визначені 15 метрик для обрахунку рейтингу гравця, за допомогою матричного методу було виконано ранжування метрик, враховуючи думки експертів.

- Запропоновано новий метод обчислення рейтингу гравців на основі матричного методу активності задля об'єднання якісних та кількісних метрик у один інтегральний показник.

- Розроблено програмні компоненти та систему розрахунку рейтингу гравців на основі запропонованого методу у вигляді незалежного аналітичного модуля для аналізу метрик и побудови графіків за результатами обчислень.

- Отримана система норм, логічних умов та правил для чіткого ведення рольової гри на форумах без особливого втручання людського фактору для підвищення ефективності та швидкості підрахунку статистики для гравців.

- Розроблено архітектуру програмної системи за допомогою графічних схем та UML-діаграм.

- Проведено тестування модуля, в результаті чого було підтверджено, що обчислення є точними та надійними.

Для розробки обрано мову програмування Java Stript для створення веб-технологій, та додатково використовувались бібліотеки для побудови графіків та схем: Chart.js та D3. js. Для візуалізації результатів було використано 6 видів графіків.

Результати роботи опубліковані в наукових публікаціях.

Проведено аналіз комерційного потенціалу розробки, який довів, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти і є перспективною розробкою. Він має кращі функціональні показники, а тому є конкурентоспроможним товаром на ринку. Існуючі переваги нової розробки дозволять швидко її поширити та популяризувати.

Задачу магістерської кваліфікаційної роботи виконано в повному обсязі: проаналізовано наявні ризики та системи конкуренти, розроблено та описано алгоритми основних блоків роботи модуля, розроблено програмну реалізацію системи та проведено ряд тестів для підтвердження коректності роботи програмного додатку та подальшої підтримки системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. О ревых играх в интернете для новичков и не только. URL: <https://4gameforum.com/threads/561977/> (дата звернення: 16.09.2019).
2. Меліхов Д., Сарматов І. Веб-аналітика: шаг к совершенству. Киев, 2010. 24 с.
3. Как правильно сортировать контент на основе оценок пользователей. URL: <https://habr.com/ru/company/darudar/blog/143188/> (дата звернення: 18.09.2019)
4. Какой же всё-таки должна быть формула расчёта рейтинга? URL: <https://habr.com/ru/sandbox/52221/> (дата звернення: 17.09.2019).
5. Романюк О. В., Лапко М. С. Особливості впровадження та розробки аналітичного модуля для проведення форумних рольових ігор *Збірник матеріалів Міжнародної науково-практичної конференції «Інформаційні технології і автоматизація»*, Одеса 2019.
6. Романюк О. В., Лапко М. С. Метод визначення рейтингу гравця форумних (play-by-post) рольових ігор. *Збірник матеріалів Міжнародної науково-практичної конференції «Електронні інформаційні ресурси в освіті і науці: створення, використання, доступ»*, Вінниця, 2019.
7. Play-by-post role-playing game. URL: https://en.wikipedia.org/Play/bypost_playing_game (Last accessed: 20.09.2019).
8. History of role-playing games. URL: https://en.wikipedia.org/wiki/History_of_role-playing_games (Last accessed: 23.09.2019).
9. Steinkuehler C. Massively multiplayer online gaming as a constellation of literacy practices, Massachusettsp, 2011. 297-318p.
10. Vampire: The Masquerade – Bloodlines. URL: https://ru.wikipedia.org/wiki/Masquerade_Bloodlines (Last accessed: 21.09.2019).
11. RPNation. URL: <https://www.rpnation.com/> (Last accessed: 21.09.2019).

12. Eladhari, M., Lindley, C. A.: Narrative Structure in Trans-Reality Role-Playing Games: Integrating Story Construction from Live Action, Table Top and Computer-Based Role Playing Games, 2005. 14p.
13. Ранговий коефіцієнт Спірмена. URL: https://pidruchniki.com/13560615/psihologiya/rangova_korelyatsiya_spirmena(дата звернення: 28.09.2019).
14. Сортировка контента на основе оценок пользователей веб-сайта. URL: <https://habr.com/ru/blog/143183/> (дата звернення: 25.09.2019).
15. Методы сбора статистической информации. URL: http://www.wikiznanie.ru/ru-wz/index.php/методы_сбора_статистической_информации (дата звернення: 30.09.2019).
16. Павловская Т. А. С/С++. Программирование на языке высокого уровня: Учебник для ВУЗов / Т.А. Павловская. Санкт-Петербург, 2003. 14с.
17. Stroustrup B. The C++ Programming Language, 4th edition. Addison-Wesley, 2013 – 1368 p.
18. Формулы расчёта персонального рейтинга. URL: <https://docs.istina.msu.ru/depadmin/ratings.html> (дата звернення: 30.09.2019).
19. Дудзяний І. М. Об'єктно-орієнтовне моделювання програмних систем. І.М. Львів: Вид-во ЛНУ імені Івана Франка, 2007. 38 с.
20. Java – обзор языка, преимущества. URL: <http://proglang.su/java/introduction-to-programming> (дата звернення: 29.10.19).
21. Яшина О. В., Земляна С. В. Мова об'єктно-орієнтованого проектування UML. РВВ ДДТУ, 2003. 21 с.
22. Структурное проектирование програм. URL: <http://poznayka.org/s86055t1.html> (дата звернення: 05.11.2019).
23. Шмюллер Д. UML за 24 години. Київ, 2005. 64с.
24. Петрик М. Р. Моделювання програмного забезпечення: науково- методичний посібник / М. Р. Петрик, О.Ю. Петрик – Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2015. 32 с.

25. International Journal of Role-Playing. URL: <https://ijrp.subcultures.nl/>
(Last accessed: 20.09.2019).
26. Степанченко И. В. Методы тестирования программного обеспечения. Волгоград: ВолгГТУ, 2006. 74 с.
27. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. Питер, 2004. 320с.
28. Поняття про тестування програмного забезпечення. URL: <http://helpiks.org/5-101392.html> (дата звернення: 07.10.2019).
29. Java – обзор языка, преимущества. URL: <http://proglang.su/java/introduction-to-programming> (Last accessed: 25.11.2019).
30. Козловський В.О. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт – Вінниця: ВНТУ, 2012. – 22 с.

ДОДАТКИ

Додаток А. Технічне завдання
Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
д.т.н., проф. О. Н. Романюк
" ____ " _____ 2019 р.

Технічне завдання
на магістерську кваліфікаційну роботу «Розробка методів та програмних засобів для підвищення ефективності проведення форумних рольових ігор»
за спеціальністю
121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:

_____ к.т.н., доц. О.В. Романюк
" ____ " _____ 2019 р.

Виконала:

_____ студентка гр.1ПІ-18м М. С. Лапко
" ____ " _____ 2019 р.

Вінниця – 2019 року

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Розробка методів та програмних засобів для підвищення ефективності проведення форумних рольових ігор».

Галузь застосування – комп'ютерна ігрова статистика.

2. Підстава для розробки.

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ №___ ректора по ВНТУ про закріплення тем МКР.

3. Мета та призначення розробки.

Метою роботи є підвищення ефективності проведення форумної рольової гри за рахунок розробки нового методу автоматизованого розрахунку інтегрального показника активності гравців на основі розширеного діапазону як кількісних, так і якісних метрик активності.

Призначення роботи – розробка методу та засобів обчислення рейтингу гравців форумної рольової гри.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись МКР.

1. Peinado, F. and Gervás, P. Transferring Game Mastering Laws to Interactive Digital Storytelling. 2004, 48-54p.
2. Mackay, D. The Fantasy Role-Playing Game: A New Performing Art. McFarland and Company. 2001. 21-35p
3. Copier, M. Beyond the magic circle: A network perspective on role-play in online games. PhD thesis, 2007. 19p

4. Eladhari, M., Lindley, C. A. Narrative Structure in Trans-Reality Role-Playing Games: Integrating Story Construction from Live Action, Table Top and Computer-Based Role Playing Games, 2005. 14p.
5. Steinkuehler C. Massively multiplayer online gaming as a constellation of literacy practices, Massachusettsp, 2011. 297-318p.

5. Технічні вимоги

Тип рольової гри – форумна (текстова), моніторинговий період – 30 днів, метод аналізу статистичних даних – матричний, максимальна кількість гравців форуму – 50 чоловік, метрики активності – якісні та кількісні, кількість метрик активності – 15, кількість експертів для визначення ваг метрик – 10.

6. Конструктивні вимоги.

Конструкція пристрою повинна відповідати естетичним та ергономічним вимогам, повинна бути зручною в обслуговуванні та керуванні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до МКР;
- технічне завдання;
- лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз предметної галузі та порівняння аналогів	10.09.2019 – 20.09.2019
2	Обґрунтування вибору методу збору статистики та вибір показників	21.09.2019 – 8.10.2019
3	Розробка розширеної системи метрик активності гравців та удосконаленого методу обчислення рейтингу активності гравця на основі матричного методу	09.10.2019 – 15.10.2019
4	Розробка програмної реалізації та тестування	16.10.2019 – 20.11.2019
5	Економічна частина	21.11.2019 – 04.12.2019

10. Порядок контролю та прийняття.

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи.

Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

Додаток Б. Лістинг коду

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <script src="http://d3js.org/d3.v3.min.js"> </script>
<style>

.axis path, .axis line {
  fill: none;
  stroke: #333;
}
.axis .grid-line {
  stroke: #000;
  stroke-opacity: 0.2;
}
.axis text {
  font: 10px Verdana;
}
.dot {
  stroke: steelblue;
  fill: lightblue;
}
</style>
<body>
  <div class = "holder">
<script type="text/javascript">

var height = 500,
    width = 500,
    margin=50,
    rawData = [
      {x: 1, y:55}, {x: 2, y: 22},{x: 3, y: 15},
      {x: 4, y: 40}, {x: 5, y: 50}, {x: 6, y: 70},
    ],
    data=[];

var svg = d3.select("body").append("svg")
  .attr("class", "axis")
  .attr("width", width)
  .attr("height", height);

// длина оси X= ширина контейнера svg - отступ слева и справа
var xAxisLength = width - 2 * margin;

// длина оси Y = высота контейнера svg - отступ сверху и снизу

```

```

var yAxisLength = height - 2 * margin;

// ф-я інтерполяції для вісі X
var scaleX = d3.scale.linear()
    .domain([0, 30])
    .range([0, xAxisLength]);

// ф-я інтерполяції для вісі Y
var scaleY = d3.scale.linear()
    .domain([300, 0])
    .range([0, yAxisLength]);

for(i=0; i<rowData.length; i++)
    data.push({x: scaleX(rowData[i].x)+margin, y: scaleY(rowData[i].y) + margin});

// Створюємо вісь X
var xAxis = d3.svg.axis()
    .scale(scaleX)
    .orient("bottom");
// Створюємо вісь Y
var yAxis = d3.svg.axis()
    .scale(scaleY)
    .orient("left");

// Малювання вісі X
svg.append("g")
    .attr("class", "x-axis")
    .attr("transform", // сдвиг оси вниз и вправо
        "translate(" + margin + "," + (height - margin) + ")")
    .call(xAxis);

// Малювання вісі Y
svg.append("g")
    .attr("class", "y-axis")
    .attr("transform", // сдвиг оси вниз и вправо на margin
        "translate(" + margin + "," + margin + ")")
    .call(yAxis);

// создаем набор вертикальных линий для сетки
d3.selectAll("g.x-axis g.tick")
    .append("line") // добавляем линию
    .classed("grid-line", true) // добавляем класс
    .attr("x1", 0)
    .attr("y1", 0)
    .attr("x2", 0)
    .attr("y2", - (height - 2 * margin));

// рисуем горизонтальные линии

```

```

d3.selectAll("g.y-axis g.tick")
  .append("line")
  .classed("grid-line", true)
  .attr("x1", 0)
  .attr("y1", 0)
  .attr("x2", yAxisLength)
  .attr("y2", 0);

// функція лінійної інтерполяції
var line = d3.svg.line()
  .x(function(d){return d.x;})
  .y(function(d){return d.y;});

var g = svg.append("g");
  g.append("path")
  .attr("d", line(data))
  .style("stroke", "steelblue")
  .style("stroke-width", 2);
// додаємо заголовок
g.append("text")
  .attr("x", (width / 2))
  .attr("y", margin - 10)
  .attr("text-anchor", "middle")
  .style("font-size", "22px")
  .text("Динаміка відвідувань (Вер)");

g.append("text")
  .attr("x", margin + 11)
  .attr("y", margin - 11)
  .attr("text-anchor", "end")
  .style("font-size", "11px")
  .text("Кількість");

g.append("text")
  .attr("x", width - margin + 11)
  .attr("y", height - margin - 5)
  .attr("text-anchor", "end")
  .style("font-size", "11px")
  .text("День");
// додаємо відмітки к точкам
svg.selectAll(".dot")
  .data(rawData)
  .enter().append("circle")
  .attr("class", "dot")
  .attr("r", 3.5)
  .attr("cx", function(d) { return scaleX(d.x)+margin; })
  .attr("cy", function(d) { return scaleY(d.y)+margin; });
</script>

```



```
</div>
</body>
<style>
  .dropbtn {
    background-color: #4CAF50;
    color: white;
    padding: 16px;
    font-size: 16px;
    border: none;
    cursor: pointer;
  }

  .dropdown {
    position: relative;
    display: inline-block;
  }

  .dropdown-content {
    display: none;
    position: absolute;
    right: 0;
    background-color: #f9f9f9;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
  }

  .dropdown-content a {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
  }

  .dropdown-content a:hover {background-color: #f1f1f1;}

  .dropdown:hover .dropdown-content {
    display: block;
  }
  .holder div {
    border-right: solid #3e8e41;
  }
  .dropdown:hover .dropbtn {
    background-color: #3e8e41;
  }
  h3 {
    border-bottom: 6px solid #3e8e41;
    background-color: lightgrey;
  }
</style>
```

```

        text-align: center;
    }
    h2 {
        border-style: groove #3e8e41;
        background-color: rgb(243, 243, 243);
        text-align: center;
        position: fixed;
        top: 0;
        left: 0;
    }
</style>
</head>
<!-- <body>
    <div class="dropdown" style="float:left;">
        <h3>Навігація</h3>
        <button class="dropbtn">Для модераторів</button>
        <div class="dropdown-content" style="left:0;">
            <a href="#">Link 1</a>
            <a href="#">Link 2</a>
            <a href="#">Link 3</a>
        </div>
<br><br>
        <button class="dropbtn">Для користувачів</button>
        <div class="dropdown-content" style="left:0;">

            </div>
        </div>

</body> -->
</html>

```

Додаток В. Алгоритм роботи модуля

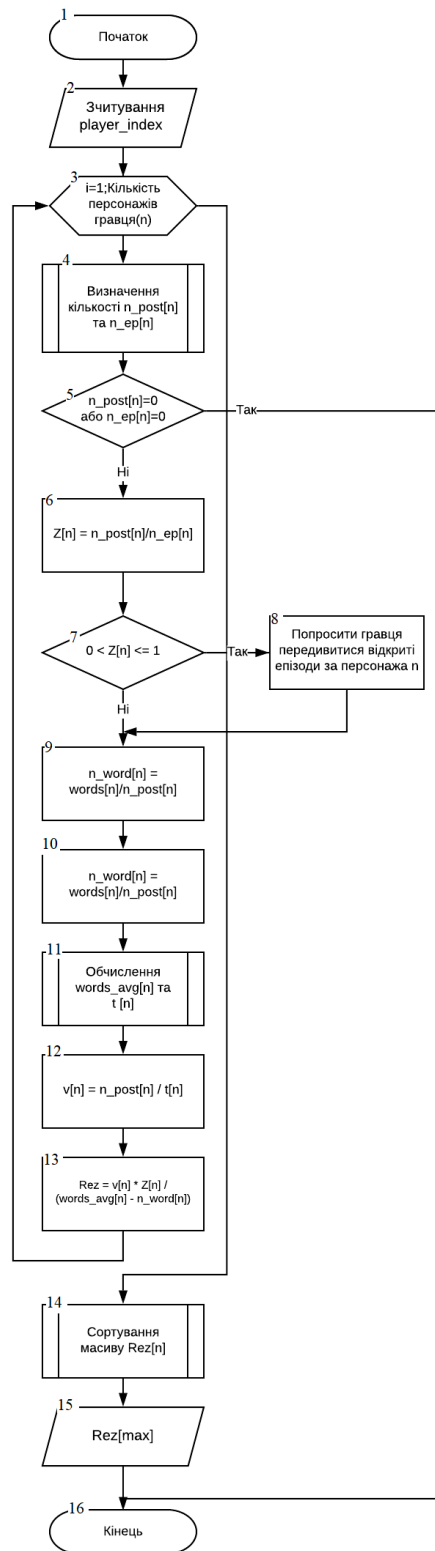


Рисунок В.1 – Алгоритм роботи аналітичного модуля

Додаток Г. Ілюстративний матеріал

Ілюстративний матеріал до захисту магістерської
кваліфікаційної роботи

Завідувач кафедри ПЗ, д. т. н., професор _____ О. Н. Романюк

Науковий керівник, к. т. н., доцент _____ О. В. Романюк

Рецензент, к. т. н., доцент кафедри КН _____ В. В. Колодний

Нормоконтроль, к. т. н., доцент _____ О. В. Романюк

Виконавець, студентка гр.1ПІ-18м _____ М. С. Лапко

Вінницький національний технічний університет
 Факультет інформаційних технологій та комп'ютерної інженерії
 Кафедра програмного забезпечення

Магістерська кваліфікаційна робота

на тему: Розробка методів та програмних засобів для підвищення ефективності проведення форумних рольових ігор

Виконала:
 студентка групи ІПІ-18м Лапко М. С.


Науковий керівник:
 к.т.н., доцент кафедри ПЗ, Романюк О. В.

Рисунок Г.1 – Слайд 1. Тема та автор МКР

2

Мета, об'єкт та предмет дослідження

Форумна (текстова) рольова гра – це рольова гра, в якій гравці взаємодіють один з одним в заздалегідь визначеному середовищі за допомогою тексту. Це підмножина онлайн-рольової спільноти, яка обслуговує як гравців, так і творчих письменників.

 **МЕТА**

Підвищення ефективності проведення форумної рольової гри за рахунок розробки нового методу автоматизованого розрахунку інтегрального показника активності гравців на основі розширеного діапазону як кількісних, так і якісних метрик активності.

Об'єктом дослідження є процес визначення активності гравців форумних рольових ігор.

Предметом дослідження є *методи та засоби* визначення активності гравців форумних рольових ігор.




Рисунок Г.2 – Слайд 2. Мета, об'єкт та предмет дослідження

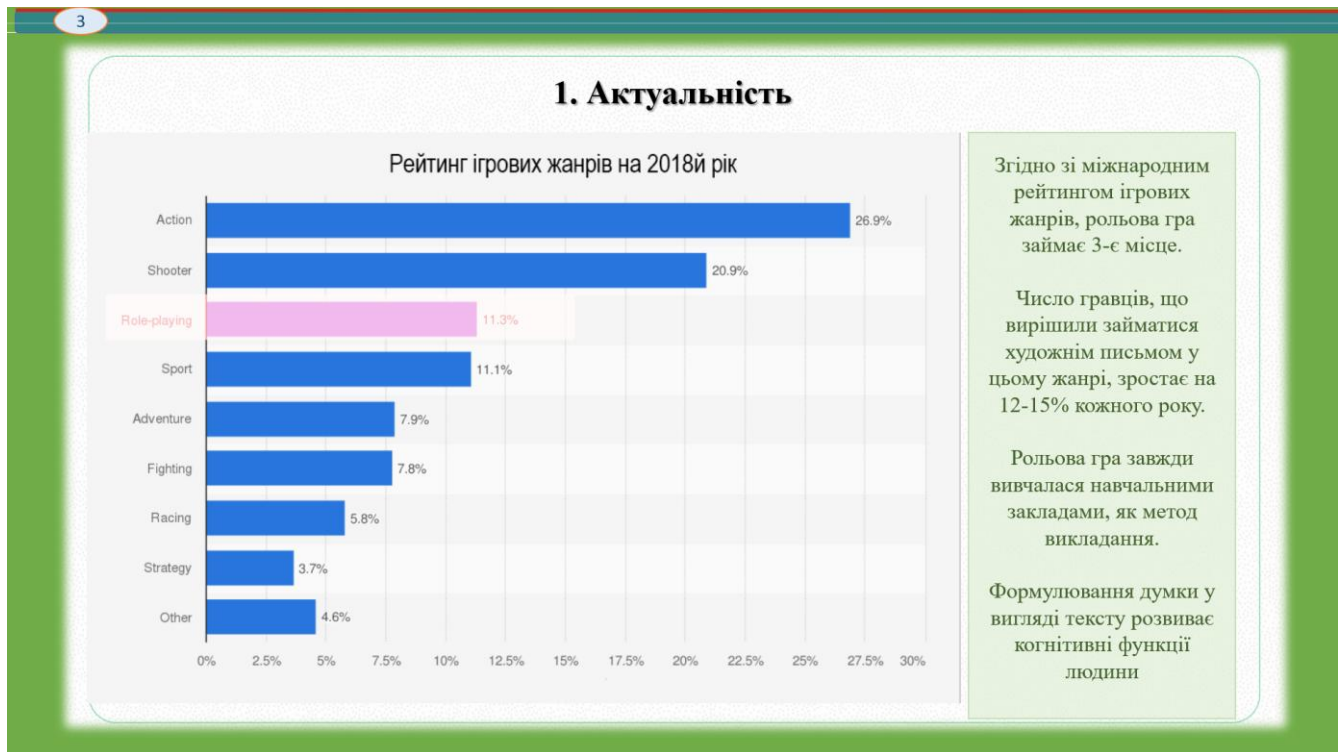


Рисунок Г.3 – Слайд 3. Актуальність



Рисунок Г.4 – Слайд 4. Проблематика під час ведення форумних ігор

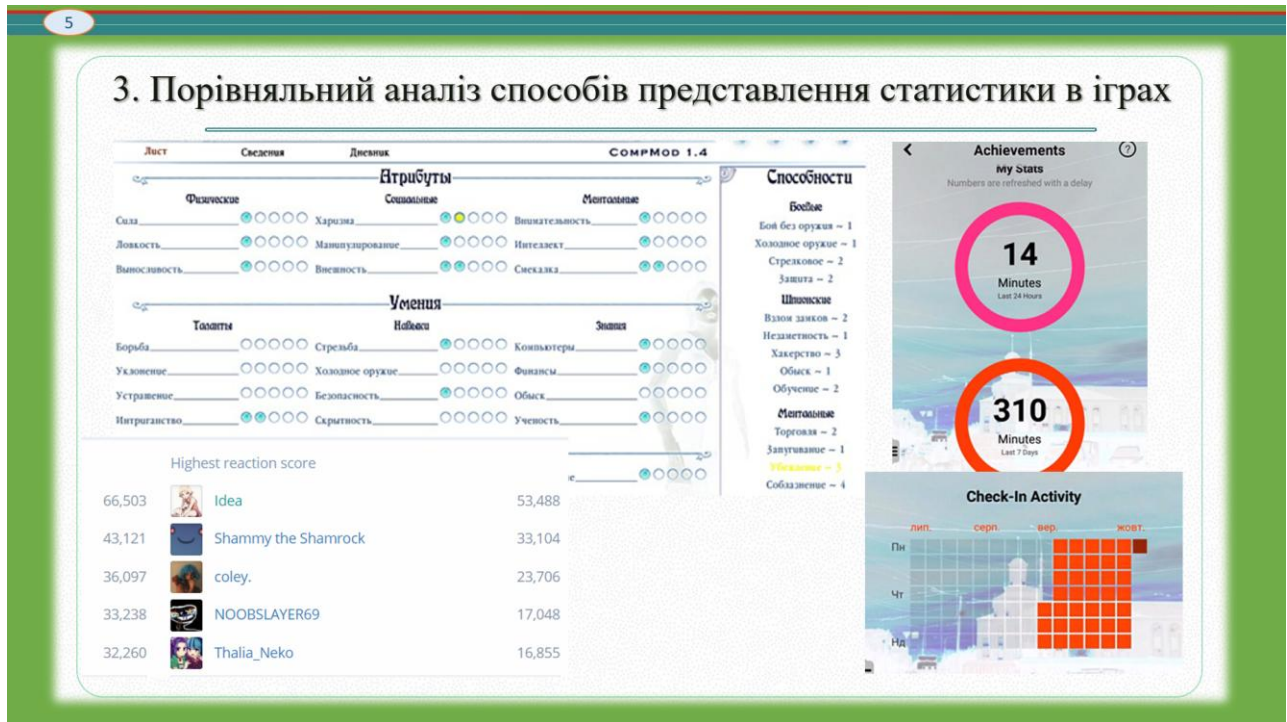


Рисунок Г.5 – Слайд 5. Порівняльний аналіз способів представлення статистики в іграх



Рисунок Г.6 – Слайд 6. Обґрунтування вибору методу статистичного спостереження

5. Визначення ваг метрик та коефіцієнта узгодженості

Метрики	Експерти										Сума рангів, S_i	d_j	d_j^2
	1	2	3	4	5	6	7	8	9	10			
1	8	9.5	8	10.5	10.5	14.5	5	13	9	9.5	97.5	17.5	306.25
2	14	13.5	13.5	10.5	14	8.5	13.5	8.5	13.5	13.5	123	43	1849
3	11	13.5	10	14	14	14.5	9	13	13.5	13.5	126	46	2116
4	14	9.5	10	7.5	10.5	12	9	13	9	13.5	108	28	784
5	8	7	6	4.5	4	8.5	9	8.5	5.5	6.5	67.5	-12.5	156.25
6	4.5	5	6	7.5	7	12	13.5	5.5	9	9.5	79.5	-0.5	0.25
7	11	9.5	13.5	14	10.5	8.5	5	5.5	13.5	9.5	100.5	20.5	420.25
8	14	9.5	13.5	14	14	8.5	13.5	8.5	13.5	9.5	118.5	38.5	1482.25
9	3	3	3	4.5	4	2	2	4	2	4.5	32	-48	2304
10	6	13.5	13.5	10.5	10.5	5.5	9	13	9	13.5	104	24	576
11	2	1.5	3	2.5	1.5	3.5	5	3	2	2.5	26.5	-53.5	2862.25
12	1	1.5	1	1	1.5	1	2	1	2	1	13	-67	4489
13	11	13.5	10	10.5	7	12	13.5	13	9	6.5	106	26	676
14	8	5	6	6	7	5.5	9	8.5	4	4.5	63.5	-16.5	272.25
15	4.5	5	3	2.5	4	3.5	2	2	5.5	2.5	34.5	-45.5	2070.25
Разом в середньому	120	120	120	120	120	120	120	120	120	120	1200		20364

$W = 0.77$

Рисунок Г.7 – Слайд 7. Визначення ваг метрик та коефіцієнта узгодженості

6. Перелік метрик активності гравця форумної рольової гри

1. Загальна кількість постів	2. Кількість епізодів	3. Дедлайн	4. Кількість слів у 1 пості
5. Середній інтервал часу між двома постами	6. Мінімальна кількість слів у пості	7. Частота написання	
8. Мінімальна кількість постів	9. Залученість	10. Кількість персонажів	11. Швидкість написання
12. Час, проведений на форумі	13. Середня кількість слів у пості	14. Відповідність форматування	15. Кількість помилок.

Рисунок Г.8 – Слайд 8. Перелік метрик активності гравця форумної рольової гри

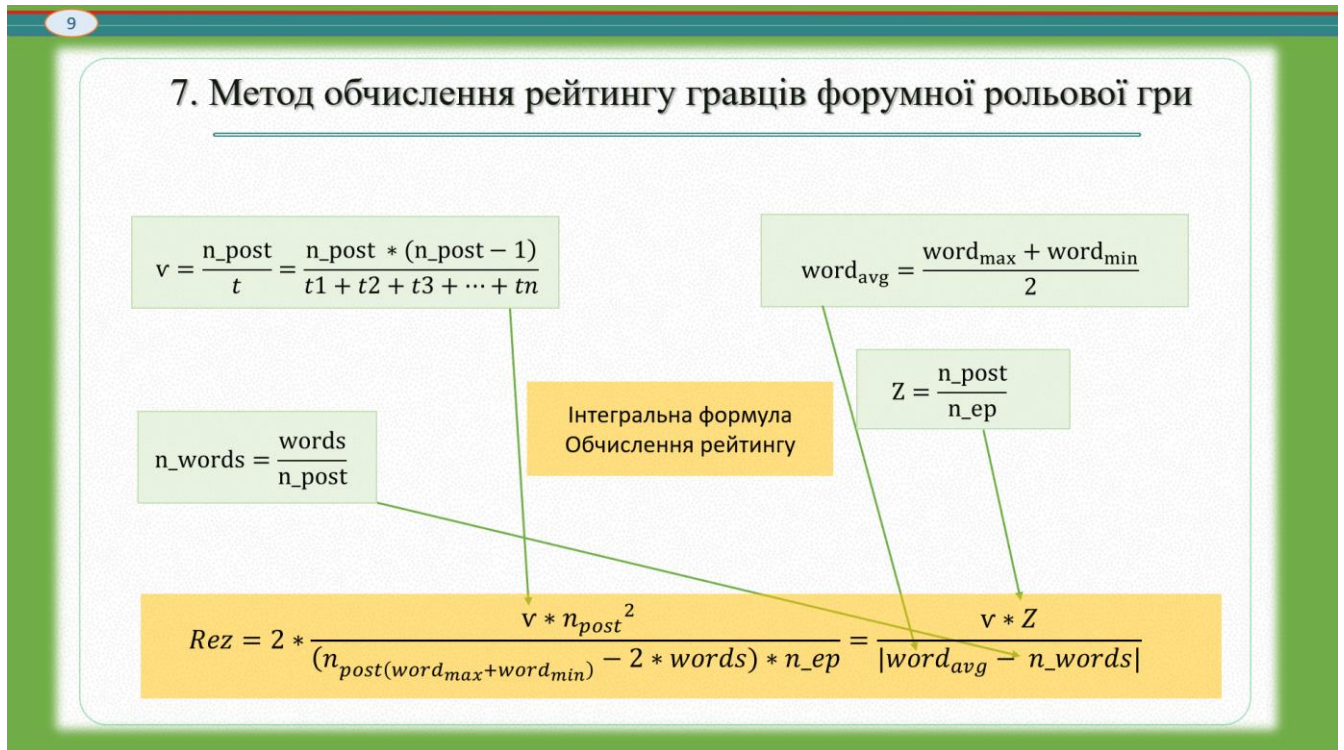


Рисунок Г.9 – Слайд 9. Метод обчислення рейтингу гравців форумної ролівої гри



Рисунок Г.10 – Слайд 10. Блок-схеми обчислення метрик активності гравців

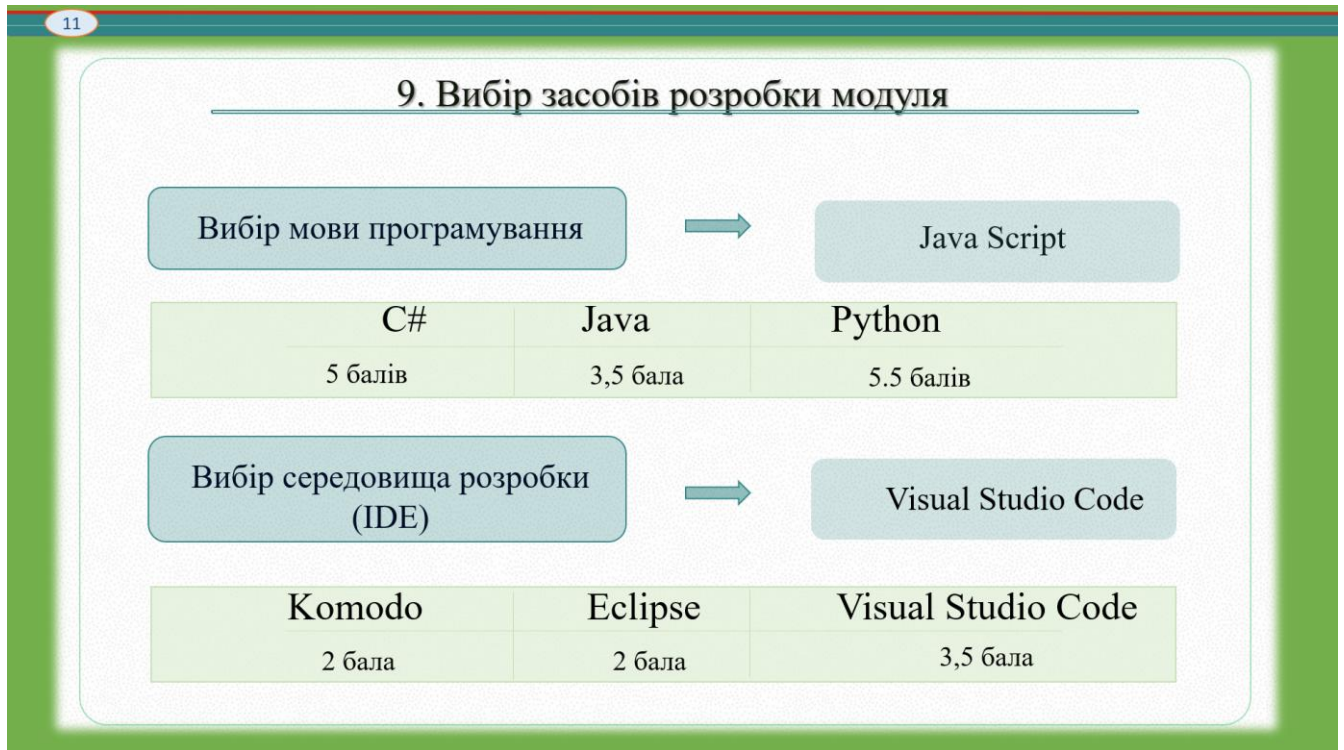


Рисунок Г.11 – Слайд 11. Вибір засобів розробки модуля

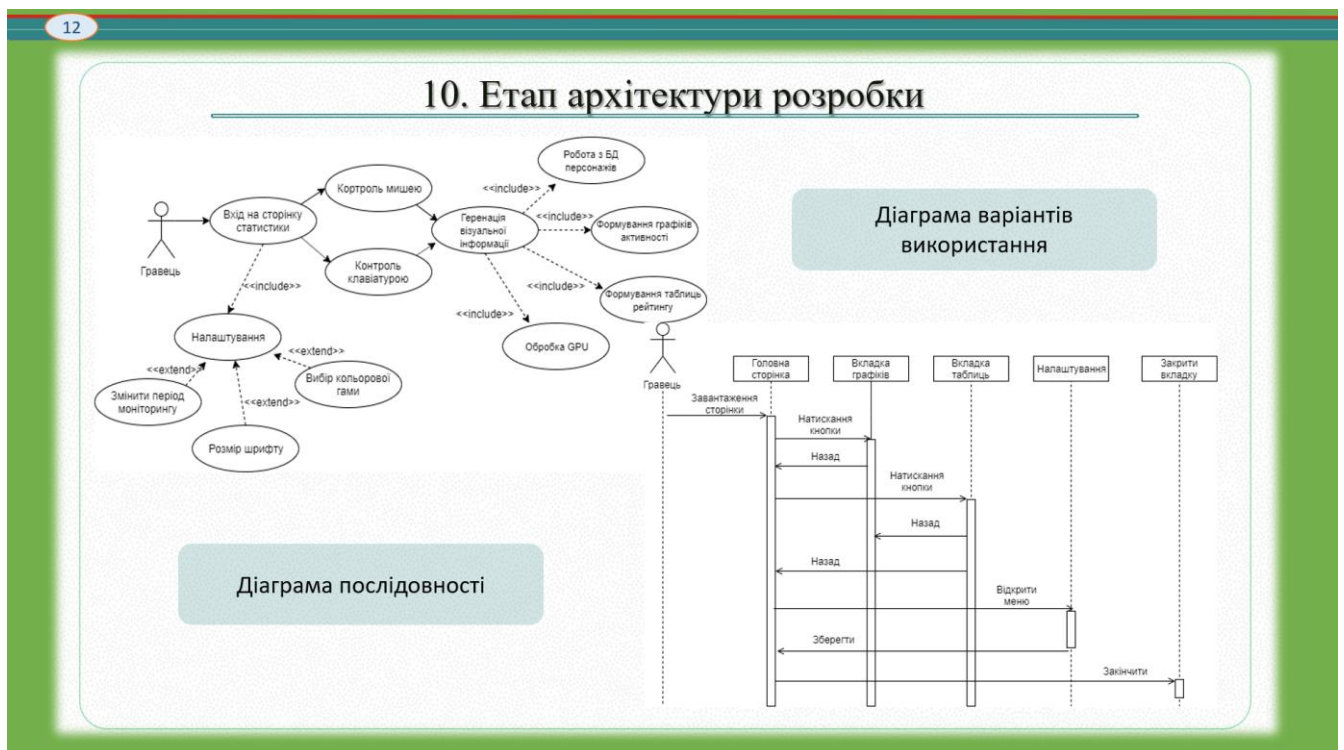


Рисунок Г.12 – Слайд 12. Етап архітектури розробки

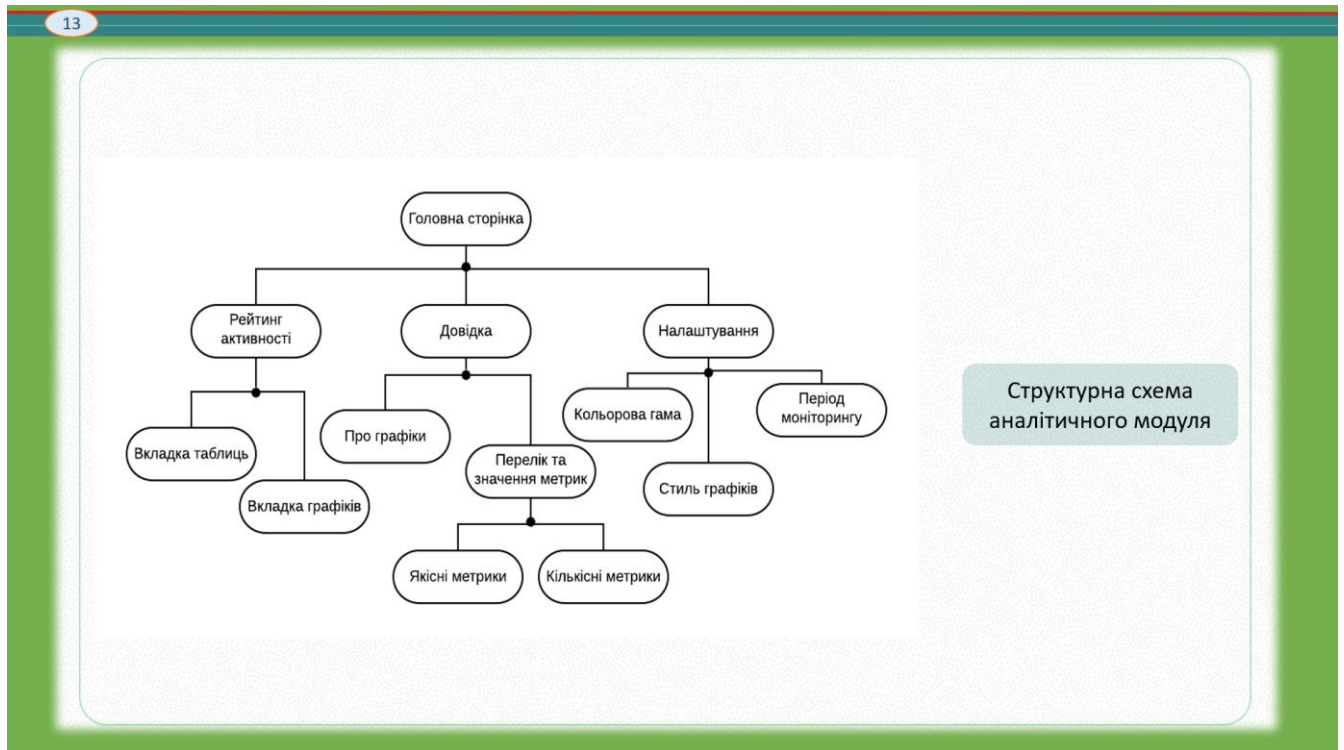


Рисунок Г.13 – Слайд 13. Структурна схема

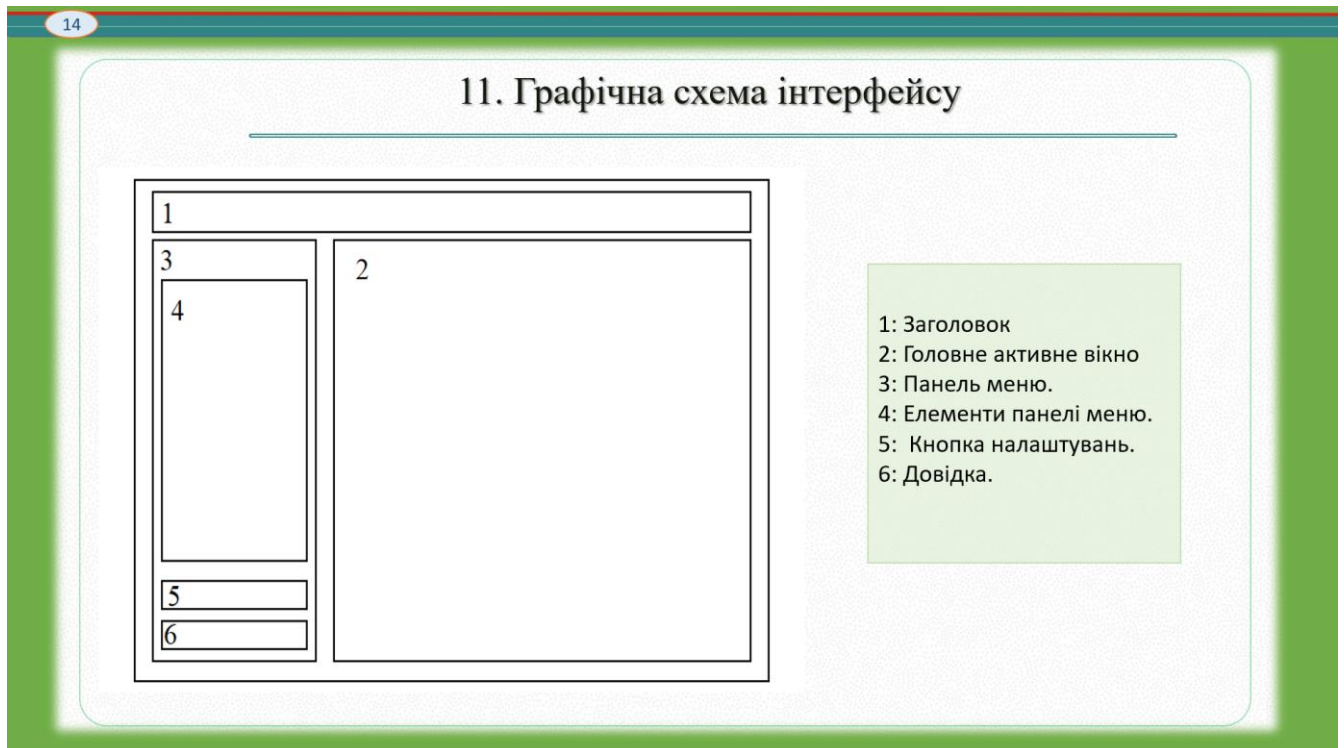


Рисунок Г.14 – Слайд 14. Графічна схема інтерфейсу

15

12. Інтерфейс програми

Дані за [Жовтень] через:

14	23	59	57
Дні	Годин	Міхвили	Сторінки

Навігація

- Для модераторів
- Для користувачів
- Налаштування
- Довідка

Статистика за [Вересень]

Оберть графік у меню.

Рисунок Г.15 – Слайд 15. Інтерфейс програми

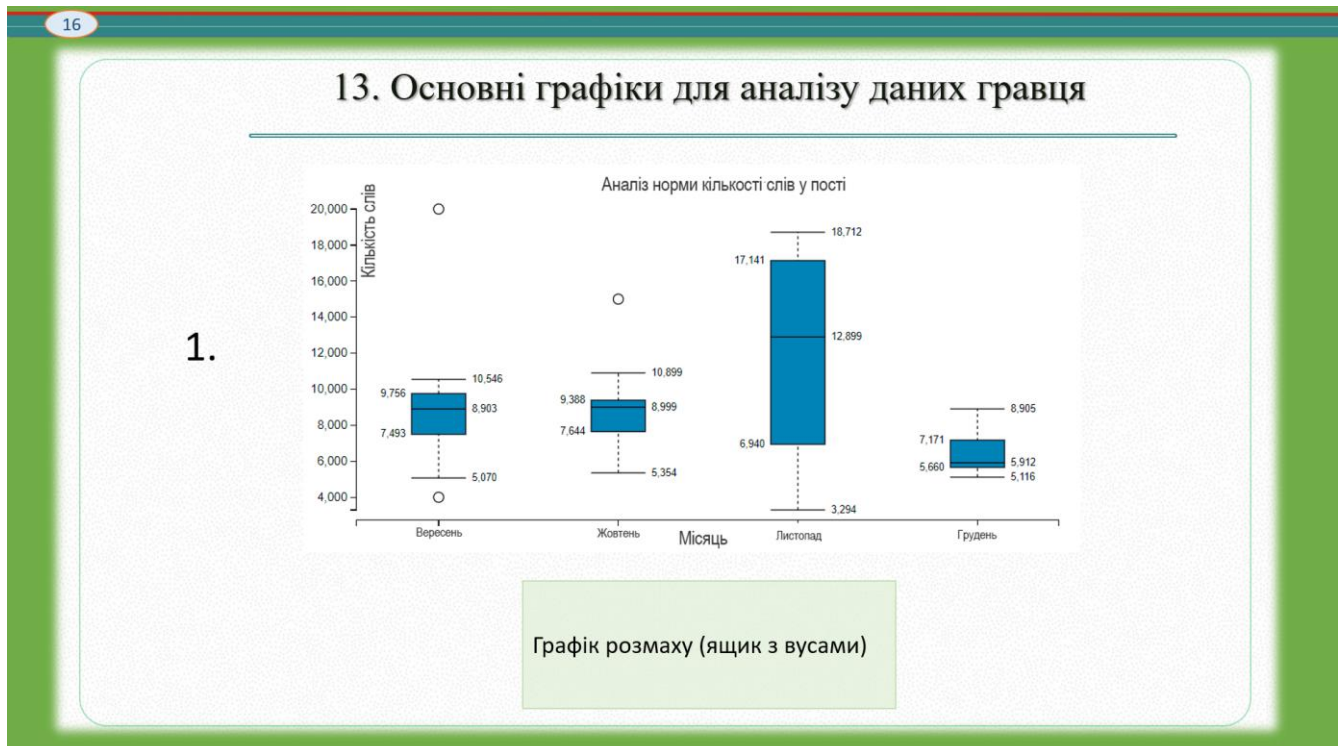


Рисунок Г.16 – Слайд 16. Основні графіки для аналізу даних гравця. Графік розмаху

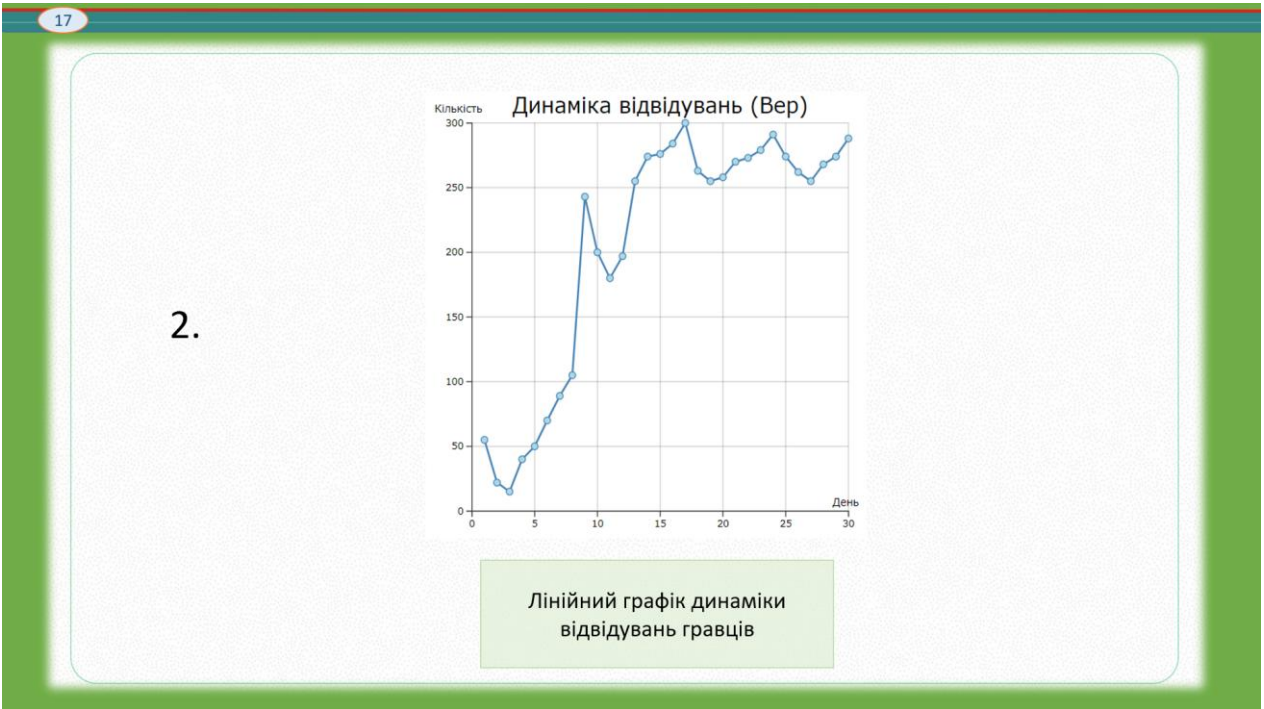


Рисунок Г.17 – Слайд 17. Основні графіки для аналізу даних гравця. Лінійний графік

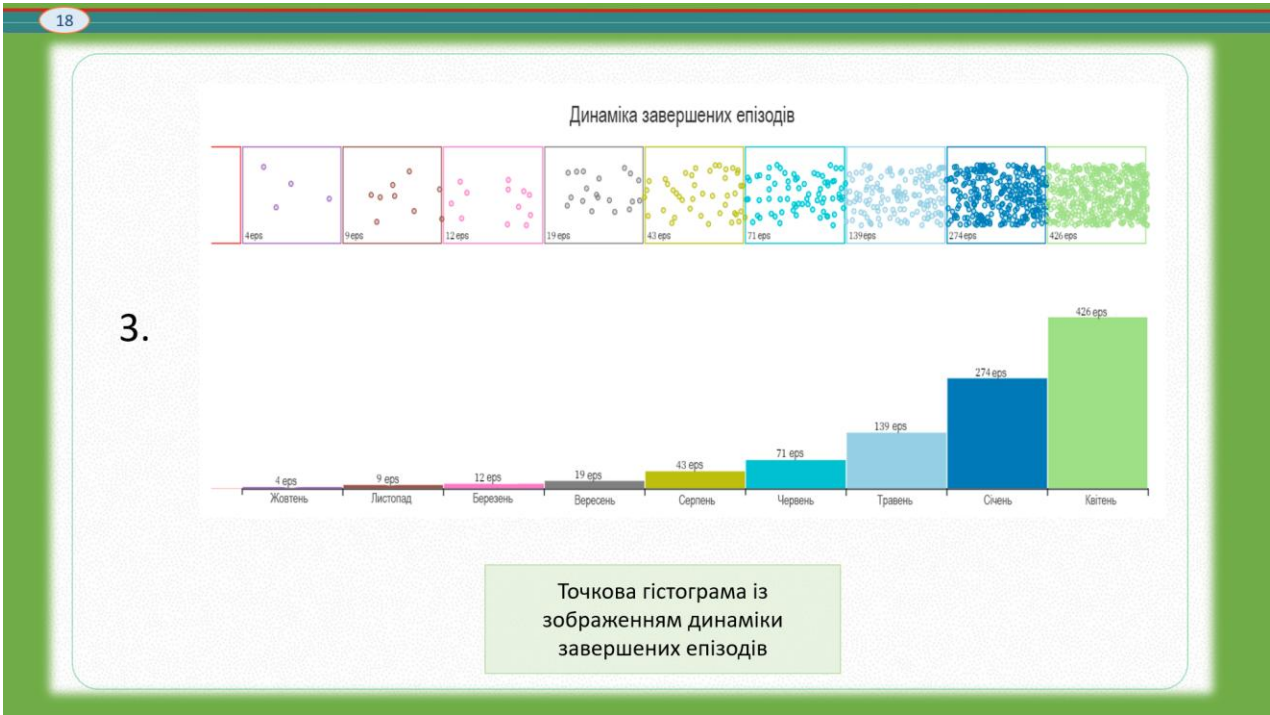


Рисунок Г.18 – Слайд 18. Основні графіки для аналізу даних гравця. Точкова гістограма

19

14. Наукова новизна та практична цінність

НАУКОВА НОВИЗНА

1. Подальшого розвитку отримав метод обчислення рейтингу активності гравця на основі матричного методу, який на відміну від існуючого використовує розширений діапазон як кількісних, так і якісних показників активності, що дозволило підвищити ефективність роботи модераторів та вилучити людський фактор при вирішенні характерних проблем при проведенні рольової гри.
2. Вперше запропоновано загальну систему показників для визначення рейтингу активності гравців форумних рольових ігор, особливість якої полягає у її універсальності та узгодженій нормативності для міжнародних рольових спільнот.

ПРАКТИЧНА ЦІННІСТЬ

Практична цінність отриманих результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби для підвищення ефективності проведення форумних рольових ігор.

Рисунок Г.18 – Слайд 18. Наукова новизна та практична цінність

20

15. Апробація та публікації

АПРОБАЦІЯ

Результати роботи доповідалися у межах XII Міжнародної науково-практичної конференції «Інформаційні технології і автоматизація – 2019» Одеської національної академії харчових технологій та Міжнародна науково-практична конференція «Електронні інформаційні ресурси в освіті і науці: створення, використання, доступ» (Вінниця, 2019).

Основні результати досліджень опубліковано в 2 наукових працях, у тому числі 2 – у матеріалах конференцій.

Рисунок Г.19 – Слайд 19. Апробація та публікації