Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

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

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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія віртуального ігрового автомата»

Виконав: студент 2-го курсу, групи 3КН-22м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)
Крамаренко Д.А.
(прізвище та ініціали)

Керівник: к.т.н., доц. каф. КН
Озеранський В.С.
(прізвище та ініціали)
«7» 12 2023 р.

Опонент: к.т.н., доц. каф. САІТ
Варчук І.В.
(прізвище та ініціали)
«7» 12 2023 р.

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
(прізвище та ініціали)
«8.12» 2023 р.

Вінниця ВНТУ - 2023 рік

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

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
_____
(підпис)
"29" 08 2023 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Крамаренку Дмитру Антоновичу

1 Тема роботи: «Інформаційна технологія віртуального ігрового автомата».
Керівник роботи: Озеранський Володимир Сергійович, к.т.н., доц. каф. КН.
затверджені наказом вищого навчального закладу «18» 09 2023 року № 247

2 Строк подання студентом роботи 13.11.2023

3 Вихідні дані до роботи:
Кількість призових донатів – не менше 5шт;
Кількість виграшних елементів – не менше 7шт;
Мова програмування – об'єктно-орієнтована;

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити): вступ; Обгрунтування доцільності розробки інформаційної технології віртуального ігрового автомата; Розробка структури та алгоритмів роботи інформаційної технології віртуального ігрового автомата; Програмна реалізація інформаційної технології віртуального ігрового автомата; висновки, перелік використаних джерел; додатки.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): схема алгоритму функціонування інформаційної технології віртуального ігрового автомата; вигляд інтерфейсу користувача; приклади роботи програм

6 Консультанти розділів проекту (роботи)

Консультанти розділів роботи в таблиці 1.

Таблиця 1 - Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Озеранський В. С., к.т.н., доц. каф. КН	29.08.23	01.09.23
4	Кавецький В.В., к.е.н., доц. каф. ЕПВМ	18.10.23	21.10.23

7 Дата видачі завдання 29.08.23

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Обгрунтування доцільності розробки інформаційної технології віртуального ігрового автомата	01.09.23 - 07.09.23	
2	Розробка структури та алгоритмів роботи інформаційної технології віртуального ігрового автомата	8.09.23 - 24.09.23	
3	Програмна реалізація інформаційної технології віртуального ігрового автомата	25.09.23 - 11.10.23	
4	Тестування та аналіз результатів роботи розробленої програми	12.10.23 - 17.10.23	
5	Підготовка економічної частини	18.10.23 - 21.10.23	
6	Розробка інструкції користувача	22.10.23 - 24.10.23	
7	Оформлення матеріалів до захисту МКР	25.10.23 - 10.11.23	

Студент _____ Крамаренко Д.А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Озеранський В. С.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

УДК 621.374.415

Крамаренко Д.А. Інформаційна технологія віртуального ігрового автомата. Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма - Системи штучного інтелекту. Вінниця: ВНТУ, 2023. 108с.

На укр. мові. Бібліогр.: 20 назв; рис.: 17; табл. 15.

Дана магістерська кваліфікаційна робота, присвячена розробці інформаційної технології віртуального ігрового автомата. Технологія призначена для удосконалення процесів генерації виграшних комбінацій.

В ході роботи проведено аналіз предметної області віртуальних ігрових автоматів та програмні засоби генерації різних ігрових ситуацій для них. Спроектовано програмні засоби для можливості управління генерацією виграшних комбінацій з використанням нечіткої логіки. Визначено структурну організацію основних модулів та сервісів технології. Здійснено програмну реалізацію інформаційної технології віртуального ігрового автомата на мові програмування PHP та MYSQL для роботи з базою даних. Результати роботи можна використовувати в маркетинговій та розважальній сферах, за у мови проходження відповідного ліцензування.

В розділі економічної частини проведено оцінювання комерційного потенціалу розробки інформаційної віртуального ігрового автомата, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, які склали 600677,72 грн, розраховано період окупності – 1,74 року.

Ключові слова: інформаційна технологія, ігровий автомат, онлайн-гра, нечітка логіка, генерація виграшних комбінацій.

ABSTRACT

Kramarenko D.A. Information technology of the virtual gaming machine. Master's thesis on specialty 122 - computer science, educational program - Artificial intelligence systems. Vinnytsia: VNTU, 2023. 108p.

In Ukrainian speech Bibliography: 20 titles; Fig.: 17; table 15.

This master's thesis is devoted to the development of information technology of a virtual gaming machine. The technology is designed to improve the processes of generating winning combinations.

In the course of the work, an analysis of the subject area of virtual gaming machines and software tools for generating various game situations for them was carried out. Software tools have been designed for the possibility of managing the generation of winning combinations using fuzzy logic. The structural organization of the main modules and services of the technology is determined. The software implementation of the information technology of the virtual slot machine in the programming language PHP and MYSQL for working with the database was carried out. The results of the work can be used in the marketing and entertainment spheres, subject to the appropriate licensing.

In the section of the economic part, an assessment of the commercial potential of the development of an informational virtual gaming machine was carried out, the costs of carrying out scientific work and implementing the results were predicted, which amounted to UAH 600,677.72, and the payback period was calculated - 1.74 years.

Keywords: information technology, slot machine, online game, fuzzy logic, generation of winning combinations.

ЗМІСТ

ВСТУП 4
1 ОБРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВІРТУАЛЬНОГО ІГРОВОГО АВТОМАТА 7
1.1 Аналіз предметної області та програмних засобів для створення віртуальних ігрових ситуацій 7
1.2 Характеристика та аналіз аналогів віртуальних ігрових автоматів 10
1.3 Алгоритми роботи віртуального ігрового автомата 13
1.4 Постановка задачі 15
1.5 Висновок до розділу 1 16
2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРТМІВ РОБОТИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВІРТУАЛЬНОГО ІГРОВОГО АВТОМАТА 17
2.1 Розробка структури інформаційної технології віртуального ігрового автомата 17
2.2 Використання сервіс-орієнтованої архітектури для віртуального ігрового автомата 20
2.3 Розробка бази даних для віртуального ігрового автомата 26
2.4 Математична модель управління віртуальним ігровим автоматом 31
2.5 Розробка алгоритму управління віртуальним ігровим автоматом 39
2.6 Висновок до розділу 2 42
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВІРТУАЛЬНОГО ІГРОВОГО АВТОМАТА 43
3.1 Обґрунтування вибору мови програмування та середовища розробки 43
3.2 Налаштування середовища розробки для віртуального ігрового автомата 46
3.3 Структура модулів програмного забезпечення віртуального ігрового автомата 49
3.4 Програмна реалізація основних модулів віртуального ігрового автомата 52
3.5 Тестування програмного забезпечення віртуального ігрового автомата 56
3.6 Висновок до розділу 3 59

4 ЕКОНОМІЧНА ЧАСТИНА 60
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки 60
4.2 Розрахунок узагальненого коефіцієнта якості розробки 64
4.3 Розрахунок витрат на проведення науково-дослідної роботи 66
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором 78
4.5 Висновки до розділу 4 82
ВИСНОВКИ 84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 86
ДОДАТКИ 88
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень 89
Додаток Б (обов'язковий) Лістинг програми 90
Додаток В (обов'язковий) Ілюстративна частина 102
Додаток Г (довідниковий) Інструкція користувача 106
Додаток Д (довідниковий) Довідка про впровадження 109

ВСТУП

Актуальність теми дослідження. На сьогоднішній день розважальний ринок є одним із найбільш прибуткових. Починаючи з початку інформаційно-технічної революції, світ стрімко рухається в майбутнє, надаючи все більш вдосконалені комп'ютерні системи для полегшення життя людини та задоволення її розваг. З появою особистих комп'ютерів, з кожним роком їх роль у житті людей постійно збільшується. Комп'ютер став невід'ємним помічником не лише у сфері економічних розрахунків, а й потужним центром розваг.

Сучасна людина іноді навіть не розглядає свій смартфон як щось інше, крім мобільного персонального комп'ютера. Зі зростанням впливу комп'ютерів на життя людей, комп'ютерна техніка значно впливає і на їхню модель поведінки. Останні дослідження показали, що середній вік гравців користувачів комп'ютерних ігор становить від 30 років і старше.

В сучасному світі віртуальні казино зайняли дуже стійкі позиції в сфері ігрового ринку. Кількість гравців, що віддають перевагу подібному формату, неймовірно швидко зростає. Це говорить про актуальність і доречність створення ігрових онлайн систем. Ігрові автомати в віртуальних казино адаптовані під такий формат і представлені в неймовірно величезній кількості. Велика популярність прийшла після офіційної заборони ігрового бізнесу, після чого багато закладів перейшли в мережу Інтернет.

Існують кілька ключових факторів, які визначають популярність та актуальність сучасних онлайн ігрових автоматів. По-перше, фінансова система грає важливу роль. Гравцям доступні різні методи поповнення свого ігрового рахунку або гри за віртуальні кошти, а також виведення зароблених коштів.

Другий важливий аспект - безпека. Всі дані шифруються та дуже уважно приховуються, забезпечуючи конфіденційність гравців. Крім того, ніхто не може спостерігати за ігровим процесом гравця, що мінімізує ризик втручання чи дезорієнтації. Третім чинником є законодавчі обмеження. У багатьох місцевостях реальні казино можуть бути заборонені, але такі обмеження не застосовуються до

онлайн ігор, що сприяє розвитку цього сегменту. Ці фактори є стимулом для постійного розвитку та популярності онлайн ігрових автоматів.

У лінійці ігрових автоматів «однорукий бандит» займає почесне місце ветерана класики азартного жанру. Перші ігрові апарати були влаштовані доволі тривіально – три барабана, які запускалися механічним важелем і зупиняючись, видавали горизонтальні комбінації з 3 символів, при збігу яких гравцеві присуджували приз. Весь процес повністю спирався на везіння, але оскільки випадання виграшної комбінації відбувалося не часто. Так автомат придбав своє прізвисько «однорукого бандита». Механіка поступово замінювалося електроприводами, що давало більше гарантій на чесність процесу.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 122 – «Інформаційна технологія віртуального ігрового автомата» та плану навчально-методичної та наукової роботи кафедри.

Мета та завдання дослідження. Метою магістерської кваліфікаційної роботи є збільшення ймовірності виграшу ігровим віртуальним автоматом.

Об'єктом дослідження є процес управління віртуальними ігровими автоматами.

Предметом дослідження є інформаційна технологія управління віртуальними ігровими автоматами.

Для досягнення поставленої мети необхідно виконати такі задачі:

- проаналізувати предметну область і програми аналоги управління віртуальними ігровими ситуаціями;
- розробити структуру інформаційної технології віртуального ігрового автомата;
- розробити базу даних а також алгоритм роботи віртуального ігрового автомата;
- здійснити програмну реалізацію інформаційної технології віртуального ігрового автомата;
- провести тестування розробленого програмного забезпечення;
- виконати економічне обґрунтування доцільності розробки нової інформаційної технології віртуального ігрового автомата.

Методи дослідження. У роботі використані такі методи наукових досліджень: аналіз інформаційних систем віртуального ігрового автомата, нечітка логіка для реалізації алгоритму створення виграшних комбінацій, сервіс-орієнтована архітектура проектування для розробки й керування функціональних модулів системи, об'єктно-орієнтоване програмування для отримання подальших результатів роботи програми.

Наукова новизна. Удосконалено інформаційну технологію віртуального ігрового автомата, що відрізняється від відомих використанням удосконаленої математичної моделі генерації виграшних комбінацій, що дало змогу збільшити ймовірність виграшу віртуальним ігровим автоматом.

Практичне значення одержаних результатів полягає у наступному:

1. Розроблено алгоритм управління віртуальним ігровим автоматом;
2. Розроблено програмний засіб управління віртуальним ігровим автоматом.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, чітким та лаконічним виведенням аналітичних співвідношень, порівнянням результатів з відомими, та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У роботах, опублікованих у співавторстві, автору належать такі результати: [1] – описано особливості створення віртуального ігрового автомата та проаналізовано підходи до їх організації.

Апробація результатів роботи. Результати роботи були апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)» (м. Вінниця, Україна, 2023 р.) [1]. Результати дослідження впроваджено в роботу ТОВ «ІТІ».

Публікації. За результатами магістерської кваліфікаційної роботи опубліковано 1 тезу доповіді на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)» [1].

1 ОБРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВІРТУАЛЬНОГО ІГРОВОГО АВТОМАТА

1.1 Аналіз предметної області та програмних засобів для створення віртуальних ігрових ситуацій

Ігровий автомат, відомий також як слот або «однорукий бандит», представляє собою особливий тип азартних ігрових автоматів. Ігри, які граються на цих автоматах, відомі як «слоти», і кожна з них має свою власну назву. Гравець може випробувати удачу не лише на фізичному слот-машині, але й на комп'ютері, встановивши спеціальне програмне забезпечення або граючи онлайн.

Слоти є дуже популярними як у реальних казино, так і в Інтернеті. Термін "slot" англійською мовою перекладається як "щілина" або "проріз" (для монети в автоматі). Раніше в усіх слот-машин була фізична щілина, куди гравець мав опустити монету. Навіть зараз існують "монетні" слоти. Однак у віртуальному середовищі, де гравець грає в інтернеті, монету опускати не потрібно. Гравець спочатку поповнює свій рахунок у віртуальному казино, якщо він грає за реальні гроші, а не у режимі, де відсутні ставки з реальними грошима.

Слот-машини складаються з трьох або більше обертаючихся коліс (барабанів), які розташовані паралельно один одному та обертаються у вертикальній площині. Кожен барабан має різні картинки-символи, такі як фрукти, гроші, тварини і т. д.

У ході гри, коли гравець робить ставку, кидаючи одну або кілька монет або використовуючи інші доступні способи ставок, він взаємодіє з ігровим автоматом, натискуючи кнопку "Spin". Це запускає процес обертання барабанів, який завершується їхньою зупинкою. Після завершення обертання символи на барабанах вирівнюються випадковим чином.

Кожен символ, відображений на барабанах, має свою унікальну вартість. Важливим є фактор того, що формування комбінацій може визначитися розташуванням однакових символів в ряду. Якщо, наприклад, в один ряд

вирівнюється комбінація із декількох ідентичних символів, це рахується як виграш, і гравець отримує виплату відповідно до встановленої виграшної таблиці.

Головною метою гравця є досягнення виграшу шляхом утворення найвигідніших комбінацій символів на барабанах. В такий спосіб грає ключову роль в визначенні успіху гравця, особливо враховуючи випадковий характер вирішення розташування символів після обертання барабанів.

Винайдений в 1887 році в Англії, перший прототип грального автомата поклав початок захоплюючій історії цього розважального пристрою. У 1905 році в США був створений відомий Liberty Bell, що визначив новий етап у розвитку гральних автоматів. З цього часу доступність гри значно зросла, що призвело до збільшення числа гравців по всьому світу. Історія гральних автоматів продовжується, привносячи радість та азарт в життя тисяч гравців усього світу.

Створення гри проходить через послідовний ряд етапів [2], розпочинаючи з розробки алгоритмів, програмного коду та бази даних. Наступним кроком є розробка контенту, який включає в себе створення малюнків, моделей, музики та анімацій. Попереду цих етапів стоїть проектування, де дизайнер генерує ідеї для майбутньої гри, обирає жанр, тематику, розробляє сценарій та образи персонажів, а також визначає оточення.

Менеджер відповідає за координацію роботи різних учасників розробки гри. Він складає план роботи, встановлює терміни виконання завдань і планує витрати. У сфері ігор працює велика кількість фахівців з різними професіями та ролями, таких як програмісти, що відповідають за технічні аспекти гри, художники, моделювальники та аніматори, відповідальні за створення графічного контенту, а також композитори та звукорежисери, які створюють звукове оформлення та музичний супровід. Це різноманіття професій вносить свої унікальні внески у створення ігрового виробу. За успішне завершення робіт над проектом відповідають керівники проекту, координуючи всі аспекти створення гри та забезпечуючи її вдале завершення [3].

Розробка ігор користується обширним спектром інструментів [3], які призначені для різних аспектів створення геймплейу, оптимізації графіки, реалізації

фізичних ефектів та інших важливих завдань. Ці інструменти сприяють не лише полегшенню процесу розробки, але й покращенню якості готового продукту.

Інноваційні рішення у сфері ігор дозволяють розробникам використовувати різноманітні платформи та технології, щоб створювати неперевершені враження для гравців. Наприклад, інструменти для віртуальної реальності розкривають нові можливості для ігор, де гравець може повністю поглибитися у віртуальний світ [4].

Однак важливо враховувати, що кожен інструмент має свої унікальні характеристики та області використання. Наприклад, інструменти для оптимізації графіки можуть бути спеціалізовані для роботи з конкретними типами графічних об'єктів чи рівнем деталізації. Розробники вибирають засоби відповідно до конкретних потреб свого проекту, жанру гри та цільової платформи [4].

Загальне використання інструментів для створення ігор сприяє розвитку галузі, вносячи нові можливості та піднімаючи якість ігрових продуктів на новий рівень.

Більшість інструментальних засобів для розробки ігор можна розбити на три основні групи:

- Фреймворки. Фреймворк – це програмна платформа, яка визначає структуру програмної системи. Він є програмним забезпеченням, яке спрощує розробку та об'єднання різних компонентів у великому програмному проекті [17]. Відзначається тим, що фреймворк визначає архітектуру програмного продукту та накладає правила на його побудову. У порівнянні з бібліотекою, фреймворк визначає не лише набір функціональності, але й структуру програмного продукту, встановлюючи правила та поведінку за умовчанням. На відміну від бібліотеки, яка може використовуватися як набір підпрограм, фреймворк впливає на архітектуру програми та диктує її структуру, яку розробник може розширювати та змінювати, враховуючи конкретні вимоги проекту. Ця група інструментів визначає величезні можливості для розробників ігор, дозволяючи їм ефективно створювати та організовувати складні програмні проекти;

- Ігровий рушій є ключовим програмним компонентом комп'ютерних і відеоігор, а також інших інтерактивних додатків з графікою, що оброблюється в реальному часі [2]. Цей невід'ємний елемент забезпечує основні технологічні можливості, спрощуючи процес розробки і часто надаючи грі універсальність, яка

дозволяє їй запускатися на різних платформах. Ігровий рушій відіграє важливу роль у створенні захоплюючих ігор, включаючи графічні, звукові та фізичні ефекти, які створюють неповторні враження для гравців. Він розширює можливості розробників, дозволяючи їм створювати високоякісні та інноваційні ігрові вироби. Що стосується платформ, ігровий рушій дозволяє грі бути багатоплатформенною, тобто запускатися не лише на персональних комп'ютерах, але і на ігрових консолях, таких як PlayStation чи Xbox. Також можливість розміщення на серверах ігрових систем дозволяє грі бути доступною для користувачів через Інтернет, що розширює її досяжність і створює можливості для мережевого геймплею та взаємодії гравців;

- Конструктор ігор – це програмне забезпечення, що поєднує в собі ігровий рушій та інтегроване середовище розробки, і, як правило, оснащене власним редактором рівнів. Своєрідний універсальний інструмент, конструктор ігор істотно полегшує процес розробки ігор, роблячи його доступним навіть для тих, хто не володіє глибокими знаннями у програмуванні. Інтеграція ігрового рушію та середовища розробки в єдиний інтерфейс надає користувачам широкі можливості для творчого вияву індивідуальності та створення унікальних геймплейних виробів. Окрім того, наявність редактора рівнів сприяє зручній роботі з геометрією гри, позначаючи важливі точки та об'єкти на віртуальних рівнях [5].

Ці програми є ідеальними для тих, хто тільки починає вивчати основи програмування, оскільки вони забезпечують простоту використання та інтуїтивний інтерфейс. Конструктори ігор стають платформою для творчості та експериментів, відкриваючи шлях для створення власних ігрових світів, навіть для тих, хто раніше не брав участі у процесі розробки ігор.

1.2 Характеристика та аналіз аналогів віртуальних ігрових автоматів

В даний момент різноманітність ігрових автоматів досягло свого піку. Але незважаючи на це в Україні найвідомішими залишаються:

– класичні слоти [7] – хороший вибір для початківця, оскільки вважаються простими і інтуїтивно зрозумілими. Вважається що представники цього виду мають високий пейаут.
– відео-слоти – вони не мають обмежень в дизайні і кількість ліній часом перевалює за тисячу. Нові технології дозволяють інтегрувати високоякісну графіку, яка скрашує ігровий процес.
– фруктові машини – за своєю механікою схожі на класичні автомати онлайн, з однією явною відмінністю, яка полягає в присутності бонусних систем, а також вимагають більшої участі з боку гравця.
– відео-покер [4] – практично не відрізняється від онлайн покеру в якому потрібно збирати комбінації з карт, тільки тут в якості вашого основного противника буде комп'ютер.

При такому розмаїтті апаратів, вибір багатьох гравців часто падає на сучасні онлайн ігрові автомати Україна з наявністю мультиплікаторів і прогресивних джекпотів від розробників Microgaming, NetEnt і Endorphina у яких відсоток віддачі досягає 98%.

Також ігрові автомати класифікуються за кількістю барабанів:

– 3 барабана. Перевагою і недоліком цих автоматів є простота гри і найменша сума можливого виграшу. Відмінний варіант для новачка або людини з обмеженим бюджетом. Яскравим представником виступають класичні слоти.
– 5 барабанів [8]. Найпоширеніший вид, оскільки має різноманітне кількість ліній виплат, а значить і ймовірність перемогти вище. Вони мають високоякісну HD графіку, різноманітну бонусну систему і включають в себе різні міні ігри на віртуальні гроші.
– 7 барабанів. Крім двох додаткових барабанів і великої кількості ліній виграшу, нічим не відрізняються від своїх попередників.

До останнього часу в Україні топовими вважалися ігрові автомати онлайн на 3 барабана, так як вони позбавлені непотрібних надлишків і досить прості у використанні, але останнім часом на перше місце виходять більш іноваційні емулятори з 3d графікою.

Популярні бонусні символи:

– wild – символ при випаданні, якого може замінити відсутні зображення для виграшної комбінації.
– scatter – символ, при випаданні якого можна збільшити виграш або запуститися додаткова міні гра.

Дані бонуси характерні для слот-машин новітнього типу і в разі вдалого збору комбінації гравцеві завжди обіцяють велику перемогу [8]. Розглянемо популярні аналоги віртуальних ігрових автоматів: TwinSpin (рис. 1.1), Pyramid.

Рисунок 1.1 – Приклад ігрового вікна віртуального ігрового автомата TwinSpin

Даний ігровий автомат є одним з перших та найпопулярніших у світі. Це приклад класичного ігрового слота. Кожний раунд починається з ідентичних суміжних подвійних барабанів, які пов'язані один з одним - звідси і назва Twin Spin. Під час обертання подвійні барабани можуть розширюватися і перетворюватися в потрійні, четверні або навіть п'ятирічні барабани. Унікальна функція синхронізації і зв'язування барабанів, яка з'являється на кожному обертанні. Відповідні символи в будь-якій позиції на трьох або більше суміжних барабанах, починаючи з крайнього лівого барабана і закінчуючи самим правим барабаном, є виграшною комбінацією. Виплачується тільки найдовша підходяща комбінація на символ. За один раунд може бути лише три виграшних комбінації.

Віртуальний ігровий автомат Pyramid є дещо складнішим за попередній аналог, так як тут присутній символ Wild який заміняє будь який інший елемент на ігровій площині (рис. 1.2).

Рисунок 1.2 – Приклад ігрового вікна віртуального ігрового автомата Pyramid

Даний символ може з’явитись лише на 2,3,4 барабанах і подовжити виграшну комбінацю. Але все ж, так як і у попередньому автоматі зарахуються лише співпадіння символів починаючи з крайнього лівого барабана і закінчуючи самим правим барабаном і можливі лише три виграшні комбінації.

1.3 Алгоритми роботи віртуального ігрового автомата

Всі ігрові автомати онлайн і оффлайн працюють на основі деякого алгоритму, RNG – random number generator. Цей термін можна перевести як генератор випадкових або рандомних чисел/значень. Це складний алгоритм, який лежить в основі абсолютно кожного слота. І при ліцензуванні онлайн казино і автоматів, перевіряється саме RNG [3]. Його задача – це генерація псевдовипадкових значень. Саме цей алгоритм робить гру непередбачуваною та справедливою. Рандомні значення відповідають за певні комбінації та символи. Ні один гравець, ні адміністрація онлайн казино не може прогнозувати результати нового повороту барабану.

Незважаючи на той факт, що історія ігрових апаратів налічує вже багато десятиліть, прогрес у цій області продовжує швидко розвиватися. Однак, можливо, однією з основних стійких констант у цьому еволюційному процесі залишається принцип та надійність генератора випадкових чисел (ГВЧ).

Генератор випадкових чисел є ключовою складовою у світі азартних ігор, і від нього залежить чесність та випадковість результатів гри. На протязі років, механізм генерації випадкових чисел в ігрових програмах постійно опрацьовується та удосконалюється, приділяючи увагу особливостям та вимогам сучасного ігрового програмного забезпечення.

Ця постійна допрацюваність генератора випадкових чисел не тільки забезпечує високий рівень випадковості результатів гри, але і враховує нові виклики та можливості, що виникають у зв'язку із зростанням технологій. Таким чином, навіть з великим прогресом у сфері ігрової технології, принципи та надійність генераторів випадкових чисел залишаються актуальними та невід'ємними частинами азартних розваг.

У створенні генератора випадкових чисел брали участь математики, програмісти і навіть психологи, і існували подібні генератори з давніх часів - їх надійність перевірена часом. Той генератор, який використовується сьогодні - це розробка професора Массачусетського технологічного університету, яка представляє собою 128-бітний алгоритм md5. При цьому використовується вона не тільки в онлайн-слотах, а й в охоронних системах і персональних комп'ютерах.

Віддача слота, RTP [5] гри, теоретичний відсоток повернення - це все один і той же параметр. RTP - return to player, означає повернення гравцеві. Термін є офіційним і використовується в наземних і онлайн казино для всіх типів азартних ігор. Зазвичай return to player виражається у відсотках і обчислюється за простою формулою: RTP = розмір виграшу / розмір ставок * 100%. Всі ігрові автомати в реальних казино і онлайн слоти в віртуальних казино зобов'язані вказувати точний показник віддачі [3]. Це одне з головних вимог, що пред'являється розробнику при ліцензуванні. При першій перевірці регулятор обов'язково перевіряє відповідність заявленого розробником значення RTP, і видає ліцензію грі, тільки якщо цей параметр вказаний вірно. Через

час автомат проходить повторну перевірку. Мало знати відсоток їх віддачі, важливо ще й розуміти, як це працює на практиці. Так, RTP = 96,34% означає, що гравець поверне 96,34% всіх зроблених на автоматі ставок, але на довгій дистанції. Не вірно вважати, що якщо за 10 або 100 обертань витратити 100 монет, то обов'язково буде виграш в розмірі 96,34 монети. Мінімальна довжина дистанції становить 1000 спинив, а регулятори і самі розробники перевіряють віддачу на кілька мільйонів обертань. Протилежним показником до RTP є House Advantage - перевага казино. Воно також обчислюється за простою формулою: House Advantage = 100% - RTP.

Значення RTP слота закладається в програмне забезпечення, як ігрових автоматів в реальному казино, так і віртуальних слотів. Для реальних автоматів створюють спеціальні чіпи, які містять програму всієї роботи гри, алгоритм генератора випадкових чисел і параметри RTP. Чіп програмується один раз і проходить перевірку. Якщо перевірка успішна, чіп монтується в автомат і пломбується. Що стосується віртуальних казино ігор, відсоток віддачі також є частиною програмного коду, створюваного девелопером. Коли розробка завершена, заявлений RTP [5] перевіряється регулятором. Тільки після успішного її проходження видається ліцензія, слот надходить у продаж і розміщується на ліцензованих ігрових майданчиках.

1.4 Постановка задачі

У даній роботі необхідно розробити інформаційну технологію віртуального ігрового автомату, що буде генерувати виграшні комбінації на стороні серверу а взаємодіяти з користувачем – за допомогою клієнт-серверної технології. У головному вікні програми для користувача будуть доступні функції:

- вибір величини початкової ставки гри;
- вибір розміру значення монети для гри;
- вибір режимів прокручування барабану: автоматичний або ручний.

Після установки таких налаштування гравець має натиснути гральну кнопку ”Spin”. Після цього клієнт надсилається запит на ігровий сервер, відбувається

подальша обробка вхідної інформації щодо раунду за допомогою вбудованих алгоритмів, і сервер повертає відповідь з результатами раунду назад до клієнта. Результати раунду формуються на базі інформації про всі минулі і наступні раунди гравця за поточний день, що зберігаються в базі даних.

Ігровий автомат повинен працювати як Web-сервіс на базі Web API, через стандартні протоколи обміну HTTP/HTTPS, а також використовувати автентифікацію гравця по зашифрованому id. Даний Web-сервіс має підтримувати формат JSON для обміну даних з клієнтом. Веб-сервіс повинен працювати під управлінням операційної системи Windows XP/7/8/10/Server, Linux, які є найбільш поширеними серед операційних систем. Ємність ОЗУ залежить від навантаження на веб-сервер, але повинно бути не меншою за 512 Мбайт.

Стабільність роботи програми та її функціональна надійність прямо залежать від вхідних даних, що передаються у тілі HTTP-запиту. Таким чином, важливо передбачити відповідну перевірку вхідних даних на коректність.

1.5 Висновок до розділу 1

Під час роботи над розділом проаналізовано предметну область, розглянуто: основні поняття інформаційної технології віртуального ігрового автомата; процеси та етапи, які будуть використовуватись при організації технології віртуального ігрового автомата.

Проаналізувавши об’єкт дослідження визначено: ключові вимоги до інформаційної технології, вхідні та вихідні дані для програмних засобів та вимоги до програмно-апаратного забезпечення системи.

У результаті аналізу систем аналогів визначено їхні характеристики, основні критерії, переваги та недоліки, область використання, розглянуто основі проблеми, які виникають при розробці подібних систем та технології за допомогою яких, можна їх усунути.

2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРТМІВ РОБОТИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВІРТУАЛЬНОГО ІГРОВОГО АВТОМАТА

2.1 Розробка структури інформаційної технології віртуального ігрового автомата

Усі веб-додатки в інтернеті використовують архітектуру клієнт-сервер. Вихідні файли додатку зберігаються на веб-сервері, який обробляє запити, що приходять через мережу інтернет від численних користувачів. На стороні клієнта веб-додаток працює в браузері, наприклад в Google Chrome або Mozilla Firefox. Користувальницький інтерфейс програми передається на клієнтську машину у вигляді сторінок написаних мовою HTML (Hypertext Markup Language), де браузер інтерпретує та відображає їх. Обмін даними здійснюється на апаратному рівні на основі протоколу TCP/IP і протоколу більш високого логічного рівня HTTP або захищеному HTTPS. Хоча протокол HTTP був розроблений ще у 1990 році, він стрімко розвивався. Існує дві версії протоколу найголовнішою їх відмінністю стала можливість тільки одного підключення до серверу для завантаження сайту, що дало величезний приріс у швидкодії. Мультиплексування – дозволяється відправка декількох запитів одночасно, з тим самим з'єднанням. Раніше, за допомогою HTTP/1.1, для оброки кожного нового запиту доводилось чекати завершення інших переказів. Пріоритезація – запитам призначаються рівні залежностей, які сервер може використовувати для швидшого надання ресурсів з більш високим пріоритетом.

У ідеалі, сервер забезпечує стандартний прозорий інтерфейс для клієнтів, приховуючи від них деталі щодо конкретної специфіки системи, такої як апаратне та програмне забезпечення, яке використовується для надання послуги.

Приклад роботи моделі клієнт-сервер та переваги першої версії протоколу HTTP над другою зображено на рисунку 2.1.

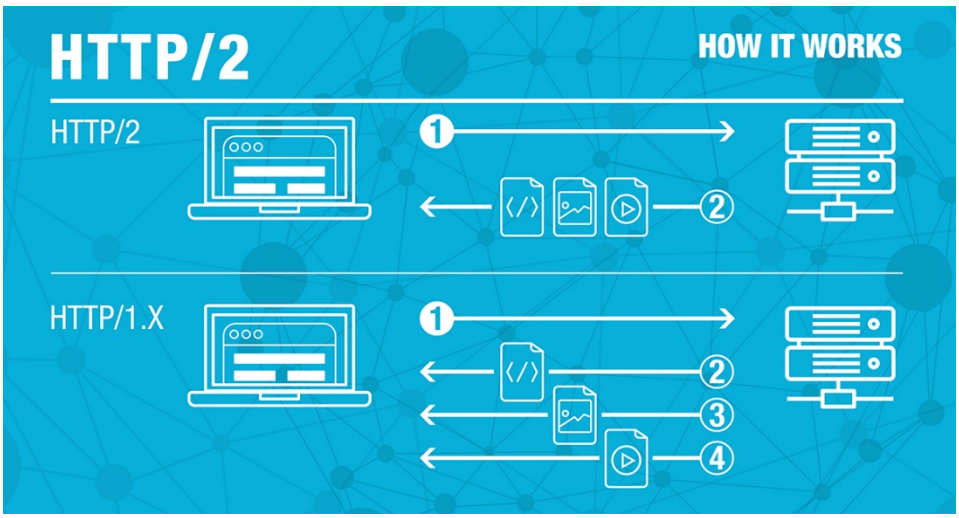

Рисунок 2.1 – Робота протоколу HTTP під управлінням клієнт-серверної моделі

Часто клієнти розташовані на персональних комп'ютерах, тоді як сервери розташовані в інших частинах мережі, як правило, на більш потужних машинах. Ця модель обчислень особливо ефективна, коли клієнти та сервери мають відокремлені завдання, які вони виконують за замовчуванням. Такого ефекту можна легко досягти використовуючи серверний php фреймворк Laravel, Slim у зв’язці з клієнтським дадотком написаним на мові програмування javascript.

У структурі інформаційної технології віртуального ігрового автомата серверною частиною буде управляти Slim Framework, взаємодіючи з певними модулями Laravel.

Slim – це мікроструктура PHP, яка допомагає швидко писати прості, але потужні веб-додатки та API. По суті, Slim – диспетчер, який отримує HTTP-запит, викликає відповідну процедуру зворотного виклику та повертає HTTP-відповідь.

Slim – це ідеальний інструмент для створення API-інтерфейсів, які використовують, обробляють чи публікують дані. Також Slim чудово підходить для швидкого прототипування. Чорт, ви навіть можете створити повнофункціональний веб-додатки з інтерфейсами користувача. Що ще важливіше, Слім дуже швидкий і має дуже мало коду. Фактично, ви можете читати та розуміти його вихідний код лише за один день. Вам не завжди потрібний кухонний комбайн на кшталт Symfony або Laravel. Це безперечно відмінні інструменти, але вони дуже переповнені зайвим

функціоналом. Slim, навпаки, надає мінімальний набір потрібних інструментів, і не більше.

Як почати працювати з цим фреймворком? По-перше, потрібний веб-сервер, такий як Nginx або Apache. Ми повинні налаштувати свій веб-сервер так, щоб він надсилав усі відповідні запити до одного файлу PHP «фронт-контролер». Ми створюємо екземпляр програми Slim і запускаємо його в цьому файлі.

Програма Slim містить маршрути, що відповідають певним HTTP-запитам. Кожен маршрут викликає обробник і повертає відповідь HTTP. Щоб розпочати роботу, спочатку створіть екземпляр програми та налаштуйте програму Slim. Потім ви визначаєте маршрути своєї програми. Нарешті, ви запускаєте програму. Коли ви створюєте програму Slim, ви часто працюєте безпосередньо з об'єктами запиту та відповіді.Ці об'єкти є фактичним HTTP-запитом, отриманим веб-сервером, і можлива HTTP-відповідь, що повертається клієнту.

Кожному маршруту Slim-додатку надаються поточні об'єкти запиту та відповіді як аргументи його обробника. Ці об'єкти реалізують найпопулярніші інтерфейси PSR-7. Маршрут Slim-програми може перевіряти або маніпулювати цими об'єктами при необхідності. Зрештою, кожен маршрут програми Slim повинен повернути об'єкт PSR-7 сумісної відповіді.

Slim також добре працює з іншими компонентами PHP. Ви можете зареєструвати додаткові сторонні компоненти, такі як Slim-Csrf, Slim-HttpCache або Slim-Flash, які ґрунтуються на функціональних можливостях Slim за замовчуванням. Також легко інтегрувати сторонні компоненти, знайдені Packagist.

Це означає, що всі ключові компоненти додатку, такі як веб-сервіси, управління базами даних, маршрутизація, кешування, аутентифікація, локалізація та управління групами користувачів, будуть підкорені цьому фреймворку (рис. 2.2). Генерація HTML документів та інші операції, пов'язані з клієнтською частиною, виконуватимуться на стороні клієнта, обробляючи відповіді від сервера у форматі JSON. Це не тільки зменшить навантаження на сервер, але й призведе до покращення продуктивності додатку.

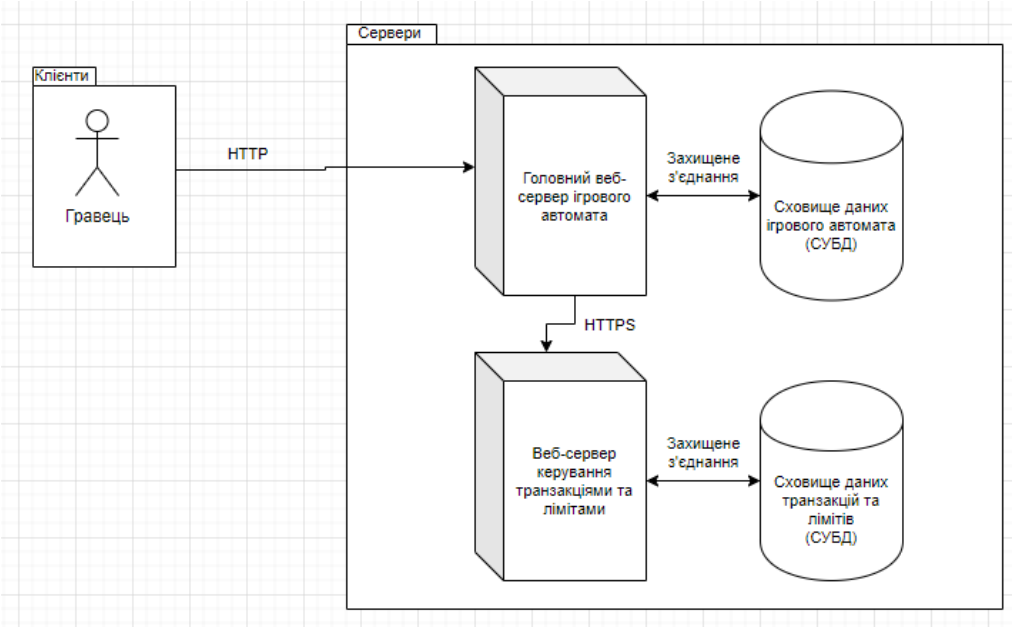

Рисунок 2.2 – Структурна схема віртуального ігрового автомата

2.2 Використання сервіс-орієнтованої архітектури для віртуального ігрового автомата

Сервіс-орієнтована архітектура (СОА) – це концепція проектування, розробки та управління функціональними модулями, відомими як сервіси. Кожен з цих сервісів є доступним через мережу та призначений для виконання конкретних дій. СОА створює комунікаційне середовище для модулів, які реалізують прикладну бізнес-логіку. Інформація про ці модулі публікується у такій формі, що їх використання не вимагає глибоких знань про використані в них рішення і технології. Ця концепція дозволяє розробникам взаємодіяти з сервісами, концентруючись лише на вхідних і вихідних даних та специфікації виклику сервісів, без необхідності в глибокому розумінні внутрішньої роботи програм. Розробнику достатньо розуміти, які дані вводяться та виводяться з сервісів, а також як і коли ці сервіси викликаються для виконання необхідних завдань.

Сервіс-орієнтовані обчислення є обчислювальною парадигмою, яка використовує сервіси як основні будівельні блоки для розробки застосувань. Ці обчислення ґрунтуються на концепції СОА і забезпечують виконання операцій, пов'язаних з управлінням сервісами. Процес розробки таких систем включає в себе

пошук, дослідження та композицію сервісів, які задовольняють вимогам користувача. Можливість композиції сервісів вважається однією з ключових переваг використання цієї парадигми. Вона передбачає знаходження набору елементарних сервісів, необхідних для реалізації функцій, які використовуються у запиті користувача, і визначення порядку їх виконання. Такий підхід дозволяє побудувати складні та гнучкі системи, які легко адаптуються до змінних потреб користувача (рис. 2.3).

Рисунок 2.3 – Структура сервіс-орієнтованої архітектури

Сервісно-орієнтована архітектура – це архітектурний шаблон програмного забезпечення, що реалізує модульний підхід до розробки програм, базуючись на використанні розподілених та слабко пов'язаних компонентів, які мають стандартизовані інтерфейси для взаємодії за нормалізованими протоколами. Технологія СОА в управлінні бізнес-процесами представляє значний прогрес у плані підвищення ефективності розробки систем і може бути порівняна за значущістю із створенням компіляторів мов високого рівня наприкінці 50-х років. Цей підхід спрощує використання веб-сервісів з будь-якого місця та їх виконання на основі бізнес-правил. Заснований на розподіленості та стандартизованих інтерфейсах, СОА

дозволяє побудовувати гнучкі, модульні системи, які легко адаптуються до змін у вимогах та умовах користувача.

СОА спонукає до використання різноманітних технологій та підходів, таких як обмін повідомленнями, для створення застосувань за допомогою зв'язування сервісів, замість написання нового програмного коду. При належному проектуванні використання обміну повідомленнями дозволяє компаніям ефективно реагувати на зміну ринкових умов, налаштовуючи процес обміну повідомленнями, замість розробки нових програм. СОА – це термін, що використовується для опису виконуваних компонентів, таких як веб-сервіси, які можуть бути викликані іншими програмами, що виступають як клієнти або споживачі цих сервісів. Такий підхід дозволяє створювати гнучкі та легко адаптовані системи, які забезпечують ефективну взаємодію компонентів підприємства та підтримують оперативність у змінюючихся умовах ринку.

У загальному сенсі, архітектура Сервісно-орієнтована архітектура передбачає наявність трьох основних учасників:

- постачальника сервісу;
- споживача сервісу;
- реєстру сервісів.

У взаємодії між учасниками СОА застосовується досить простий механізм: постачальник сервісу реєструє свої сервіси в реєстрі, а споживач звертається до реєстру із запитом. Основна умова для використання сервісу полягає в дотриманні угоди про інтерфейс для звернення до сервісу, і цей інтерфейс повинен бути платформонезалежним.

Архітектура СОА реалізує масштабованість сервісів, що означає можливість додавання нових сервісів і модернізації існуючих. Постачальник сервісу та його споживач є незалежними одне від одного, вони взаємодіють за допомогою повідомлень. Оскільки інтерфейс повинен бути неприв'язаним до платформи, то технологія визначення повідомлень також повинна бути платформонезалежною. Тому, як правило, повідомлення реалізуються у вигляді XML-документів, що відповідають XML-схемі.

Концепція Web-сервісів, що з'явилася наприкінці 90-х років XX століття, стала неоціненним галузевим стандартом у сфері інформаційно-комунікаційних технологій. Стандарти Web-сервісів були розроблені видатними компаніями, такими як IBM, Microsoft, Ariba, Sun Microsystems, SAP, і отримали підтримку Консорціуму W3C. В рамках W3C було створено робочу групу Web Services Architecture Working Group, яка опублікувала глосарій термінів у сфері Web-сервісів, що сприяє уніфікації та розумінню цього технологічного підходу. Web-сервіси використовують XML для обміну даними між застосуваннями, незалежно від операційної системи, апаратної платформи і розробника. Одним із ключових аспектів Web-сервісів є їх здатність до логічного пов'язання функцій, які можуть бути програмно викликані через мережу Інтернет. Web-сервіс, здебільшого ідентифікується через URI, представляє собою комплексну програму, інтерфейс якої може бути описаний у вигляді мови XML, що сприяє зрозумілості та взаємодії у розподіленому середовищі.

Web-сервіси представляють собою програмно реалізовану систему, спроектовану для підтримки міжкомп'ютерної взаємодії в телекомунікаційних мережах, і опираються на такі ключові стандарти, як SOAP (Simple Object Access Protocol) – протокол обміну повідомленнями, WSDL – мова опису програмних інтерфейсів Web-сервісів та UDDI (Universal Description, Discovery and Integration) – класифікатор Web-сервісів. Процес комунікації між Web-сервісами та іншими системами базується на використанні повідомлень у форматі SOAP, передаваних за допомогою протоколу HTTP і взаємодіючих за допомогою XML, в поєднанні з іншими Web-стандартами. Кожен конкретний Web-сервіс описується у спеціальному документі WSDL, який містить інформацію про функції, які він надає. Цей документ виступає основою для взаємодії інших систем із Web-сервісами, дозволяючи їм здійснювати ефективний обмін інформацією в розподіленому середовищі.

Для ефективного пошуку Web-сервісів використовують спеціалізовані реєстри, що опираються на стандарт UDDI (Universal Description, Discovery and Integration). Процес публікації Web-сервісів для користувачів може використовувати два основні методи – UDDI і DISCO. UDDI є централізованим та структурованим сервісом реєстрації, який діє як центральне сховище для специфікацій та інформації про

підприємства, включаючи послуги, які вони надають через Інтернет. Використовуючи UDDI, користувачі можуть легко знаходити та отримувати доступ до різноманітних Web-сервісів. DISCO, натомість, пропонує більш вільний механізм пошуку через браузер. Використовуючи DISCO, користувачі можуть відкривати доступ до різноманітних Web-сервісів за допомогою відкритих і вільно форматованих методів пошуку. SOAP Web-сервіси стають доступними для користувачів через протоколи HTTP GET, HTTP POST і HTTP SOAP, що спрощує взаємодію та обмін даними між різними програмними системами через Інтернет.

SOAP (Simple Object Access Protocol) представляє собою стандарт передачі повідомлень через Інтернет, розроблений фірмою Microsoft для віддаленого виклику процедур (RPC, Remote Procedure Call) за допомогою протоколу HTTP. Його основна мета – забезпечити можливість обміну інформацією у форматі XML між різними обчислювальними пристроями та програмами.

SOAP надає гнучкість та універсальність у використанні, дозволяючи використовувати будь-яку мережу, будь-який протокол передачі даних та різні обчислювальні пристрої, включаючи мобільні.

Специфікація SOAP визначає XML як "конверт" для передачі повідомлень, а також визначає метод для кодування програмних структур даних у форматі XML. Вона також забезпечує засоби зв'язку через протокол HTTP, що робить SOAP ефективним засобом для взаємодії між різними системами, незалежно від їхньої архітектури чи мови програмування.

Щодо WSDL (Web Services Description Language), це стандарт, заснований на XML, який надає опис того, як користуватися конкретним Web-сервісом. Визначений Консорціумом W3C, WSDL включає інформацію про доступні операції, структуру повідомлень, формати даних та інші деталі, що дозволяють розробникам ефективно взаємодіяти із сервісом.

REST (Representational State Transfer) - це підхід до архітектури мережевих протоколів, що забезпечує доступ до інформаційних ресурсів. Цей підхід був описаний та популяризований у 2000 році Роєм Філдінгом, який є одним із творців протоколу HTTP. Основною ідеєю REST є використання принципів функціонування

Інтернету, зокрема, можливостей HTTP. Роєм Філдінгом був розроблений REST паралельно з HTTP 1.1, базуючись на попередньому протоколі HTTP 1.0. Заснований на принципах клієнт-серверної архітектури та безстандартності, REST спрощує взаємодію між компонентами системи. Одним із ключових аспектів REST є використання невеликої кількості стандартних форматів для передачі даних, таких як HTML, XML, JSON. Це сприяє легкості розуміння та обміну інформацією між клієнтами і серверами. Крім того, REST використовує стандартні HTTP-методи (GET, POST, PUT, DELETE) для взаємодії з ресурсами, що робить його простим та ефективним для розробки та впровадження.

Будь-який протокол REST, зокрема HTTP, має властивість підтримувати кешування, не залежати від мережевого прошарку та утримувати найменше можливу інформацію про стан між парами "запит-відповідь". Це ключові принципи, які призначені забезпечити масштабовність системи та надати можливість системі еволюціонувати з урахуванням нових вимог. У контексті підходу REST відзначається тим, що він активно використовує кешування, що дозволяє зберігати копії ресурсів на клієнтському або проміжному (проксі) рівні. Це дозволяє зменшити навантаження на сервер та покращити продуктивність системи. Ключовий принцип безстандартності означає, що сервер не зберігає стан між парами "запит-відповідь", що робить систему більш масштабовною та відповідною до принципів REST. Зменшення кількості методів та простота протоколу дозволяють системі бути більш легко зрозумілою та піддається змінам, що є важливими як для розробників, так і для користувачів системи.

Мова опису Web-сервісів (Web Services Description Language або WSDL) виконує ключову роль у забезпеченні інтеграції Web-сервісу в застосування. Опис Web-сервісу мовою WSDL надає технічні деталі, такі як формат повідомлень та доступні операції, які є необхідними для успішної взаємодії з сервісом. На даний момент існують продукти, які підтримують стандарт WSDL, такі як SOAP Toolkit 2.0 (WSDL Generator) від Microsoft та WSDL Toolkit від IBM. SOAP Toolkit 2.0 від Microsoft та WSDL Toolkit від IBM є продуктами, які допомагають генерувати WSDL-описи для Web-сервісів, сприяючи їхній ефективній інтеграції в різноманітні

застосування. Ці інструменти дозволяють розробникам отримувати та надавати необхідну інформацію для ефективного використання Web-сервісів. У контексті Web-сервісів стандарт Universal Description, Discovery, and Integration (UDDI) відіграє роль виявлення доступних сервісів. UDDI формує бізнес-реєстр (UDDI Business Registry), де провайдери Web-послуг можуть реєструвати свої послуги, а розробники можуть знаходити та вибирати необхідні їм сервіси для подальшої інтеграції в свої застосування.

Компанії активно реєструють свою присутність у Business Registry, що є загальнодоступною базою даних. UDDI, як ключовий компонент цього процесу, дозволяє компаніям описувати, інтегрувати та публікувати свої сервіси. Цей централізований Business Registry створює уніфіковане середовище, в якому компанії можуть представляти свої Web-сервіси, забезпечуючи доступність інформації для інших учасників.

UDDI, як самостійний Web-сервіс, використовується для пошуку і вибору необхідних сервісів. Користувачі та застосування можуть взаємодіяти з UDDI для знаходження конкретних послуг, що відповідають їхнім потребам.

За допомогою семантичного опису Web-сервісів, який реалізований у OWLS (Ontology Web Language for Services), можливі інтелектуальний пошук та автоматичне компонування Web-сервісів. OWLS визначає стандарт для опису семантики Web-сервісів, що дозволяє більш глибоко розуміти їхню функціональність та сприяє автоматизації процесу пошуку і вибору сервісів.

2.3 Розробка бази даних для віртуального ігрового автомата

База даних – упорядкований набір логічно взаємопов'язаних даних, що використовуються спільно, та призначені для задоволення інформаційних потреб користувачів. В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Базою даних можна вважати будь-який впорядкований набір даних. В даному випадку інформацію про склади, товари,

категорії до яких вони можуть відноситись також інформацію про клієнтів та історію їх замовлень.

MySQL – це потужна та відкрита система керування реляційними базами даних (СКБД), яка була спеціально розроблена як альтернатива комерційним аналогам. Зараз MySQL є однією з найбільш поширених та визнаних систем у цьому сегменті. Вона знаходить широке застосування, зокрема, в створенні динамічних веб-сайтів, завдяки відмінній підтримці різноманітних мов програмування. MySQL виділяється своєю ефективністю та надійністю, що робить її популярним вибором серед розробників. Її відкритий код сприяє активному співтовариству користувачів та постійному вдосконаленню. Широкий спектр функціональності MySQL робить її ідеальним інструментом для зберігання та обробки даних у великих та складних веб-проектах.

MySQL вражає своїм різноманіттям функцій, що забезпечують безпечне та надійне середовище для зберігання, обслуговування та отримання даних. Її винятковою швидкодією, стійкістю та простотою у використанні робить цю систему керування реляційними базами даних ідеальним інструментом для високоефективної обробки великих обсягів інформації. Розроблена з метою оптимізації роботи з обширними базами даних, MySQL дозволяє не лише забезпечити стабільність та безпеку даних, але й значно прискорити виконання операцій. Ця система виявляється важливим ресурсом для тих, хто вирішує завдання зі зберігання та опрацювання великого обсягу інформації в реальному часі.

Для ефективного функціонування інформаційної технології віртуального ігрового автомтата необхідне сховище інформації про користувачів, транзакції раундів в яких вони брали участь та значення ліміту для контролю випадіння виграшних комбінацій. Також потрібно зберігати інформацію про сесії гравців, звичайні та бонусні раунди (спіни), занчення ставок та виграшів для контролю виграшних комбінацій та ведення звітності у системі. Для реалізації сервісу технічної підтримки у системі, а також для вирішенню можливих конфліктів і відтворення результатів раундів необхідно зберігати повну інформацію про відповіді сервера

клієнту гри. На основі цих потреб було створено реляційну базу даних з такими таблицями: users, sessions, limits, transactions, spins, freespins, bets, wins, games.

Нормалізація – це формальний процес, в ході якого атрибути даних організовуються у таблиці, а самі таблиці об'єднуються в базу даних. Основні завдання нормалізації включають:

- вилучення повторної інформації. Під час нормалізації виявляються і вилучаються дубльовані дані з таблиць, що сприяє уникненню зберігання зайвої інформації;

- створення гнучкої структури. Нормалізація дозволяє створити структуру бази даних, яка легко адаптується до майбутніх змін. Це означає, що можна додавати нові дані чи змінювати існуючі, не порушуючи цілісності даних.

- мінімізація впливу змін. Структура бази даних після нормалізації створюється так, щоб вплив структурних змін на програми, які використовують дані цієї бази, був мінімізований. Це допомагає забезпечити стабільність та надійність функціонування додатків.

Для досягнення першої нормальної форми (1НФ), таблиця повинна бути двовимірною і не містити груп, що повторюються. У таблиці 1НФ існують тільки дві характеристики: довжина (кількість записів або рядків) та ширина (кількість полів або стовпців). Кожна комірка таблиці містить тільки одне значення, і не допускається наявність комірок, які включають кілька значень.

За потреби, якщо необхідно зберігати кілька значень в одній комірці, це може вимагати введення третього виміру - глибини. Глибина дозволяє створювати більш складні структури, де кожна комірка може містити набір значень. Однак для забезпечення нормалізації даних і відповідності принципам 1НФ важливо уникати зберігання масивів чи списків у межах одного поля.

Для досягнення другої нормальної форми (2НФ), дані у всіх не ключових стовпцях повинні повністю залежати від первинного ключа, а також від кожного елемента (стовпця) первинного ключа, якщо ключ є складеним. Це означає, що кожне поле в таблиці повинно бути повністю залежним від первинного ключа, і значення в

кожному не ключовому стовпці має однозначно визначатися значенням первинного ключа.

Якщо одне з полів не залежить від всіх компонентів первинного ключа, то для виправлення ситуації може бути необхідно включити це поле в склад первинного ключа або розбити таблицю на декілька, включаючи доповняльні таблиці.

Перед перевіркою на відповідність другій нормальній формі, таблиця повинна бути приведена до першої нормальної форми.

Для досягнення третьої нормальної форми (3НФ), всі неключові стовпці таблиці повинні не тільки залежати від первинного ключа таблиці, але також бути незалежними один від одного. Це означає, що в таблиці не повинно бути транзитивних функціональних залежностей між стовпцями.

Для досягнення цієї нормальної форми, спочатку потрібно переконатися, що таблиця вже відповідає першій та другій нормальним формам. Після цього слід аналізувати функціональні залежності в таблиці та розглядати можливі випадки транзитивних залежностей.

Якщо виявляються транзитивні функціональні залежності, можуть бути вжиті заходи для їх вирішення, такі як розбиття таблиці чи перегляд структури з метою усунення залежностей.

Відношення "один-до-одного" між таблицями є найпростішим та найбільш прозорим способом організації даних в базі даних. У такому відношенні кожному запису в одній таблиці відповідає тільки один запис у іншій таблиці, і навпаки. Це відношення може бути корисним у випадках, коли важливо зберігати чітку і однозначну взаємодію між записами.

Для поліпшення ефективності та зручності обробки даних, таблиці, які зв'язані відношенням "один-до-одного", можна об'єднати в єдину таблицю, що складається з полів обох початкових таблиць. Це дозволяє уникнути зайвого повторення даних та прискорює доступ до необхідної інформації.

У випадках, коли важливо контролювати доступ до конфіденційних чи важливих даних, можна використовувати механізми управління доступом для

обмеження прав користувачів до конкретних частин таблиць. Це забезпечить безпеку та конфіденційність важливої інформації.

Відношення "один-до-багатьох" є одним із найпоширеніших у базах даних та встановлює зв'язок між одним записом в першій таблиці та декількома записами в другій. Це досягається за допомогою використання первинного ключа базової таблиці та зовнішнього ключа зв'язаної таблиці. Зовнішній ключ в зв'язаній таблиці є посиланням на первинний ключ в базовій таблиці.

Зовнішній ключ в зв'язаній таблиці може бути частиною складеного первинного ключа, але він завжди є зовнішнім по відношенню до базової таблиці. Це дозволяє створювати зв'язки між даними, де одному запису в першій таблиці може відповідати будь-яка кількість записів у другій таблиці. Використання відношення "один-до-багатьох" є дуже розповсюдженим, оскільки воно дозволяє ефективно виражати та управляти зв'язками між різними частинами інформації в базі даних.

На рисунку 2.4 продемонстровані зв’язки таблиць “spins”, “bets”, “wins”, “freespins”.

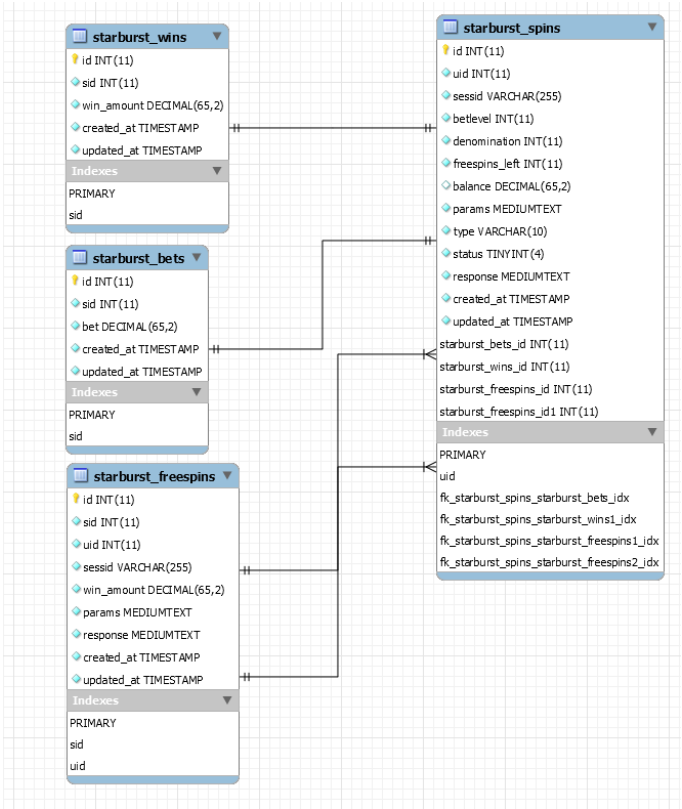

Рисунок 2.4 – Приклад зв’язків таблиць “spins”, “bets”, “wins”, “freespins”

В даному випадку використані такі типи зв’язків: “Один-до-одного” – таблиця “spins” зв’язана з “bets” за допомогою індексного поля sid, тобто одному раунду може належати одна ставка. Таким же чином таблиця “spins” зв‘язана із таблицею “wins”, тобто кожний раунд може мати лише один виграш. “Багато-до-одного” – з іншого боку таблиця “freespins” зв’язана з “spins” по тому ж полю sid, що означає приналежність одного або більше бонусних раундів до однієї звичайного. Таким же зв’язком, “Багато-до-одного”, зв’язані транзакції з раундами, тобто кожний раунд може мати дві транзакції, які розділяються за типом, у випадку якщо це ставка або виграш.

Відношення “багато-до-одного” протилежне відношенню “один-до-багатьох”. Якщо вибір відношення “багато-до-одного” або “один-до-багатьох” не має великої ролі, то відношення між таблицями називається рефлексивним. Відношення “багато-до-одного” є відображенням відношення “один-до-багатьох”.

Таким чином за допомогою різних типів зв’язків була побудована база даних для інформаційної технологіх віртуального ігрового автомата. Завдяки даній концепції бази було розроблено багато потрібних функцій у системі. Наприклад зберігання транзакцій кожного з раундів гравця, тобто збереження фінансового сліду у системі, для подальшого створення алгоритму для контролю за виграшними комбінаціями, а також ведення звітності для аналізу окупності гри та відслідковування за користувачами. Також дуже важливим рішенням було створення таблиць “spins”, “freespins”, “bets” та “wins” для зберігання інформації про ставки та виграші у раундах. Розділення об’єкта раунда на чотири таблиці дало змогу надійно зберігати інформацію та швидко доступатись до неї у разі необхідності не втрачаючи цілісності даних.

2.4 Математична модель управління віртуальним ігровим автоматом

У віртуальному ігровому автоматі є п'ять ігрових барабанів, кожен з яких містить сто комірок для п'яти символів, тобто $100^5=10^{10}$, що становить 10 мільярдів можливих комбінацій. Символи позначені умовно як 1, 2, 3, 4, 5, і частоти їх

випадання задані наступним чином: 1:2:3:6:8 (гравцеві не відомо про частоти, але є відома матриця виплат, з якої випливає, що «п'ятірка» зустрічається найчастіше через найменші виплати, але точна частота невідома. «Четвірка» зустрічається рідше і т.д., а «одиничка» є найбільш рідкою і «дорогою»). Оскільки 1+2+3+6+8=20, ймовірності появи символу 1, 2, 3, 4, 5 на одному з барабанів після його зупинки відповідно розподілені як p(1)=0,05; p(2)=0,1; p(3)=0,15; p(4)=0,3; p(5)=0,4.

У грі найменший виграш відзначається тоді, коли на п’яти барабанах, розташованих зліва направо, утворюються три однакові символи, і в даному випадку, це три символи «5».

При цьому, матриця виплат за різну кількість однакових символів має вигляд:

$$M_V = \begin{pmatrix} 9 & 7 & 5 & 3 & 2 \\ 45 & 35 & 26 & 16 & 10 \\ 180 & 144 & 108 & 65 & 37 \end{pmatrix} \qquad (2.1)$$

У першому рядку вказані виграші за три однакові символи («одиничка», «двійка», «трійка», «четвірка», «п’ятірка»). У другому рядку вказані виграші за чотири відповідних символи, а у третьому – за п’ять таких символів. Наприклад, якщо поставити одну гривню, і отримаєти три символи «5», то виграш буде 2 грн.. Аналогічно, за три символи «4» отримаємо виграш у розмірі 3 грн, а далі – все аналогічно.

Розрахуємо ймовірності випадіння чітко трьох однакових символів поспіль, починаючи з першого (позначення -1 означає, що символ не є одиничкою, * – будь-який символ):

$$p(1;1;1;-1;*) = 0,05*0,05*0,05*(1–0,05)*1 = 0,00011875;$$
$$p(2;2;2;-2;*) = 0,13*(1–0,1)*1 = 0,0009;$$
$$p(3;3;3;-3;*) = 0,153*(1–0,15)*1 = 0,00286875;$$
$$p(4;4;4;-4;*) = 0,33*(1–0,3)*1 = 0,0189;$$
$$p(5;5;5;-5;*) = 0,43*(1–0,4)*1 = 0,0384.$$

Аналогічно, розрахуємо ймовірність появи чітко 4 однакових символів поспіль, починаючи з самого першого:

$$p(1;1;1;1;-1) = 0{,}054*(1–0{,}05) = 0{,}0000059375;$$
$$p(2;2;2;2;-2) = 0{,}14*(1–0{,}1) = 0{,}00009;$$
$$p(3;3;3;3;-3) = 0{,}154*(1–0{,}15) = 0{,}0004303125;$$
$$p(4;4;4;4;-4) = 0{,}34*(1–0{,}3) = 0{,}00567;$$
$$p(5;5;5;5;-5) = 0{,}44*(1–0{,}4) = 0{,}01536.$$

Обчислимо ймовірність появи чітко 5 однакових символів поспіль:

$$p(1;1;1;1;1) = 0{,}055 = 0{,}0000003125;$$
$$p(2;2;2;2;2) = 0{,}15 = 0{,}00001;$$
$$p(3;3;3;3;3) = 0{,}155 = 0{,}0000759375;$$
$$p(4;4;4;4;4) = 0{,}35 = 0{,}00243;$$
$$p(5;5;5;5;5) = 0{,}45 = 0{,}01024.$$

Тоді матриця обчислених ймовірностей буде мати вигляд:

$$M_p = \begin{pmatrix} 0{,}0001187500 & 0{,}00090 & 0{,}0028687500 & 0{,}01890 & 0{,}03840 \\ 0{,}0000059735 & 0{,}00009 & 0{,}00043031255 & 0{,}00567 & 0{,}01536 \\ 0{,}0000003125 & 0{,}00001 & 0{,}0000759375 & 0{,}00243 & 0{,}01024 \end{pmatrix} \quad (2.2)$$

Розрахуємо RTP для уього варіанту. RTP - це абревіатура від англ. «return to player» - дослівно «повернення гравцеві». Це офіційний термін, що використовується для всіх видів азартних ігор. RTP визначається як теоретичний відсоток коштів, які гравець може очікувано повернути у довгостроковій перспективі при грі в конкретний ігровий автомат.

Наприклад, якщо слот має RTP 95%, це означає, що у середньому з кожних 100 одиниць ставки гравець може очікувати повернення 95 одиниць. RTP є важливим показником для гравців, оскільки він вказує на те, наскільки «виграшною» є гра.

RTP виражається у відсотках. Його обчислюють за формулою:

$$RTP = \frac{V}{S} \cdot 100\% \quad (2.3)$$

де V – об’єм виграшу гравця;

S – об’єм ставок гравця.

$V = M_v * M_p$

Розглянемо, як обчислюється коефіцієнт RTP у нашому випадку.

Сумарний виграш обчислюється, як сума добутків всіх елементів матриці ймовірності Мр на відповідні їм елементи матриці виплат MV:

V = 9606653125 грн.

А це суттєво відрізняється від 10^{10}.

Тобто для даних обчислень:

$$RTP = 9606653125 / 10^{10} = 96,06\%$$

Це означає, що гравець, граючи на даному автоматі, може очікувати повернення 96,06% від усіх зроблених ставок, але лише на довгостроковій дистанції. Важливо відзначити, що неправильно припускати, що витративши 100 грн на 10 або 100 спінів (обертань), гравець обов'язково отримає виграш у розмірі 96,06 грн. Мінімальна довжина дистанції, яку враховують для оцінки віддачі, становить близько 1000 спінів, а регулятори разом з розробниками перевіряють RTP на значно більшому обсязі обертань, який нерідко оцінюється в мільйонах.

Розгянемо наступну комбінацію, додавши новий символ «wild», що може замінити собою будь-який інший символ, тобто комбінація (1,Wild,1,Wild,2) = (1, 1, 1, 1, 2) – а це призова комбінація.

Припустимо, що кожний ігровий барабан має 100 комірок, з яких символ «Wild» займає 1; символ «1» – 5; символ «2» – 9; символ «3» – 15; символ «4» – 30; символ «5» – 40.

Розрахуємо ймовірність випадання на одному барабані таких символів:

$$p(\text{«Wild»}) = 0{,}01;$$

$$p(\text{«1»}) = 0{,}05;$$

$$p(\text{«2»}) = 0{,}09;$$

$$p(\text{«3»}) = 0{,}15;$$

$$p(\text{«4»}) = 0{,}3;$$

$$p(\text{«5»}) = 0{,}4.$$

При цьому, матриця виплат за різну кількість однакових символів має вигляд:

$$M_V = \begin{pmatrix} 9 & 7 & 5 & 3 & 2 \\ 38 & 30 & 22 & 13 & 9 \\ 152 & 120 & 88 & 53 & 33 \end{pmatrix} \quad (2.4)$$

Обчислимо ймовірності випадіння чітко 3 однакових символів «1» поспіль, починаючи з першого барабана:

- Три «1», четвертий символ не «1» і не «Wild», п'ятий - будь-який;

- Дві «1» та 1 «Wild» (причому, в будь-якому порядку), четвертий - не «1» і не «Wild», п'ятий - будь-який. Кількість таких варіантів буде: $C^1_3 = 3$;

- Одна «1» і 2 «Wild» (причому, в будь-якому порядку), четвертий -не «1» і не «Wild», п'ятий - будь-який. Кількість таких варіантів буде: $C^3_1 = 3$.

Важливо відзначити, що варіант з трьома символами «Wild», четвертий «*» і п'ятий - будь-яким, який не дорівнює «*» і не «Wild», відповідає чотирьом символам «*».

Отже, отримуємо сумарну ймовірність для всіх трьох варіантів:

$$p_{1,1,1} = 0{,}053 *(1- 0{,}05 - 0{,}01)*1+ 3*0{,}052 +0{,}01*(1- 0{,}05- 0{,}01)*1+$$
$$+ 3*0{,}05*0{,}012*(1- 0{,}05-0{,}01)*1 = 0{,}0002021;$$
$$p_{2,2,2} = 0{,}093 *(1- 0{,}09 - 0{,}01)*1+ 3*0{,}092 *0{,}01*(1- 0{,}09 - 0{,}01)*1+$$

$$+3*0,09*0,012*(1- 0,09 - 0,01)*1 = 0,0008991;$$

$$p_{3,3,3} = 0,0034398;$$

$$p_{4,4,4} = 0,0205551;$$

$$p_{5,5,5} = 0,0406628.$$

Оцінимо ймовірність появи чітко 4 однакових символів «1» поспіль, починаючи з першого ігрового барабана:

- 4 «1», п’ята не «1» і не «Wild»;

- 3 «1» і «Wild» (причому, у будь-якому порядку), п’ята - не «1» і не «Wild», таких варіантів буде $C^1_4 = 4$;

- 2 «1» і 2 «Wild» (причому, у будь-якому порядку), п’ята - не «1» і не «Wild», таких варіантів буде $C^2_4 = 6$;

- 1 «1» і 3 «Wild», п’ята - не «1» і не «Wild», таких варіантів буде $C^1_4 = 4$;

Останній варіант - 4 «Wild», п’ята «*» (або «Wild») - такий варіант прирівнюється до 5 «*» (або 5 найдорожчих одиничок).

Розподіл ймовірностей для всіх варантів по чотири має вигляд:

$$p_{1,1,1,1} = 0,05^4 *(1- 0,05- 0,01) + 4*0,05^3 *0,01*(1- 0,05 - 0,01) +$$
$$+6*0,05^2 *0,01^2 *(1- 0,05- 0,01) + 4*0,05*0,01^3 *(1- 0,05 -0,01) = 0,000012173 ;$$
$$p_{2,2,2,2} = 0,09^4 *(1- 0,09 - 0,01) + 4*0,09^3 *0,01*(1- 0,09 - 0,01)+$$
$$+6*0,09^2*0,01^2*(1- 0,09 - 0,01) + 4*0,09*0,01^3 *(1- 0,09 - 0,01) = 0,000089991;$$

Для чотирьох символів «3», «4» і «5» - обчислення аналогічні.

Обчислимо ймовірність появи тільки п’яти однакових символів «1» поспіль, починаючи з першого ігрового барабану:

- п’ять одиночок;

- чотири одиночки і 1 «Wild» (причому, у будь-якому порядку), таких варіантів буде $C^1_5 = 5$;

- три одиночки і 2 «Wild» (причому, у будь-якому порядку), таких варіантів буде $C^2_5 = 10$;

- дві одиночки і 3 «Wild» (причому, у будь-якому порядку), таких варіантів буде $C^3_5 = 10$;

- одна одиночка і 4 «Wild» (причому, у будь-якому порядку), таких варіантів буде $C^1_5 = 5$;

- 5 «Wild». Такий варіант відповідає тільки найдорожчій комбінації одиничок.

$$p_{1,1,1,1,1} = 0{,}05^5 * 5*0{,}05^4*0{,}01 + 10*0{,}05^3 *0{,}01^2 +10*0{,}05^2 *0{,}01^3+$$
$$+ 5*0{,}05*0{,}01^4 + 0{,}01^5 = 0{,}0000007776;$$

Відмітимо, що якщо p(Wild) = 0,01; p(1)=0,05, то ймовірність появи або 5 одиничок, або 5 «Wild»: p1,1,1,1,1 = $(0{,}01+0{,}05)^5$ = 0,0000007776. Отже це теж саме, що і у попередньому обчисленні.

Для 5 інших символів буде тільки 5 доданків:

$$p_{2,2,2,2,2} = 0{,}09^5 + 5*0{,}09^4 *0{,}01 + 10*0{,}09^3 *0{,}01^2 + 10*0{,}09^2 *0{,}01^3 +$$
$$+5*0{,}09*0{,}01^4 = 0{,}0000099999 .$$
$$p_{2,2,2,2,2} = (0{,}09 + 0{,}01)^5 - 0{,}01^5 = 0{,}0000099999.$$

Аналогічні до останнього обчислення і для символів «3», «4» і «5».

Для 5 символів «3», «4» і «5» - обчислення аналогічні.

Отже, результуюча матриця ймовірностей матиме вигляд:

$$M_p = \begin{pmatrix} 0{,}0002021000 & 0{,}0008991000 & 0{,}0034398000 & 0{,}020555100 & 0{,}040662800 \\ 0{,}0000121730 & 0{,}0000899910 & 0{,}0005504940 & 0{,}006372288 & 0{,}016671984 \\ 0{,}0000007776 & 0{,}0000099999 & 0{,}0001048575 & 0{,}002862915 & 0{,}011585620 \end{pmatrix}.$$

Звідси, сума добутків всіх елементів матриці M_V на відповідні їм елементи матриці M_p становить сумарний виграш: 9610688702 грн. А вкладено в гру було 10 млрд. грн, а значить віддача слоту RTP = 96,11%.

Зазначимо, що можна налаштовувати розподіл символів та довжину барабану для кожного барабана окремо. У нашому випадку ми обрали п'ять символів так, щоб частота появи кожного символу на кожному барабані була однаковою, і довжина кожного барабану дорівнювала ста елементам. З таким підходом модель може працювати з барабанами різного вигляду. Ці невеликі зміни легко програмуються і дозволяють обчислити теоретичний відсоток повернення гри (RTP) відповідно.

Розрахувавши теоретичний відсоток повернення гри (RTP) та оцінивши вигідність гри для конкретного розподілу символів на кожному барабані, казино може легко налаштувати матрицю виплат або змінити розподіл символів так, щоб гра була прибутковою для казино. Важливо зауважити, що матриця виплат зазвичай приваблює гравця великими виграшами, і коригування зазвичай відбувається через невидимий для гравця розподіл символів. У кінцевому підсумку, статистично гравець втрачає гроші при кожній ставці.

Розмір RTP встановлюють за допомогою симуляції 1 мільярда раундів на ігровому сервері. Тобто, щоб отримати ліцензію від регулятора, постачальник повинен пройти процес сертифікації, а незалежний аудитор – перевірити симуляцію ігрового сервера. Цей аудит визначає правильність оголошеного RTP. Самостійно регулювати відсоток повернення власники казино не можуть.

В онлайн-автоматах параметр віддачі закладається в код, який також перевіряють спеціальні органи. За відсутності порушень модель з'являється на гральному ринку. Тому гравцям важливо обирати саме ліцензійну платформу, адже ситуація з онлайн-казино без ліцензії зовсім інакша.

Нелегали використовують піратський софт, який працює на їхніх ресурсах, тому можуть змінювати RTP та інші параметри на свій розсуд. Не варто дивуватися, якщо реальний показник повернення в слотах нелегальних казино може становити всього 50%.

2.5 Розробка алгоритму управління віртуальним ігровим автоматом

Для створення виграшних комбінацій було розроблено алгоритм, робота якого складається з декілького частин поділених на мікро-сервіси.

Перш за все відбувається збір та підсумування інфомації з бази даних про ставки та виграші користувача за поточний день. Далі за відповідною формулою розраховується допустимий ліміт на виграш користувача.

$$\mathbf{Max = T_b * (1 - Limit / 100) - T_w} \quad (2.5)$$

де, T_b – сума ставок гравця за поточний день;

T_w – сума виграшів гравця за поточний день;

Limit – закріплене значення – коефіцієнт заробітку гри.

На рисунку 2.5 зображено роботу алгоритму збору інформації про ставки та виграші користувача. Він представляє собою sql запит у базу даних, у якому використовуються додаткові методи Sum() для підсумування значення поля, та перевірка isNull().

Рисунок 2.5 – Алгоритм розрахунку виграшного ліміта віртуального ігрового автомата

Наступним кроком, на основі результатів розрахунку виграшного ліміта будується ігрове поле та шукаються співпадіння за встановленими правилами, у випадку успіху, виграш оцінюється за трьома категоріями (низький, середній, високий) і у випадку задоволення розміру ліміта гравцеві дозволяється виграти або весь цикл запусається знову до моменту випадіння “вірної” комбінації.

На рисунку 2.6 зображено повний алгоритм генерації ігрового поля та результату раунду, тобто виграшних комбінацій.

1. Спочатку веб-сервіс гри робить HTTP запит до окремого сервісу, який відповідає за керування транзакціями користувачів та лімітами гри.
2. Запускаєтсья цикл for в якому не залежно від значення ліміта відпрацьовує метод створення ігрового поля (createReels), який працює на основі методу випадкових чисел та відсоткових співвідношень в залежності від ваги кожного символа на полі.
3. У випадку знаходження співпадінь за допомогою методу match(), розраховується майбутній виграш гравця, який в подальшому оцінюється, як низький, середній чи високий. Низький та середній виграші пропускаються системою, у випадку високого виграшу гравця, цикл створення ігрового поля та генерації виграшу повториться, але уже з параметром, для зниження відсотків генерації символів із великою вагою і далі порівняється з доступним лімітом. Якщо ліміт перевищено основний цикл алгоритму буде повторюватись до тих пір поки не буде задовонеий ліміт. Таким чином ми контролюємо ігровий процес і заробіток гравця.

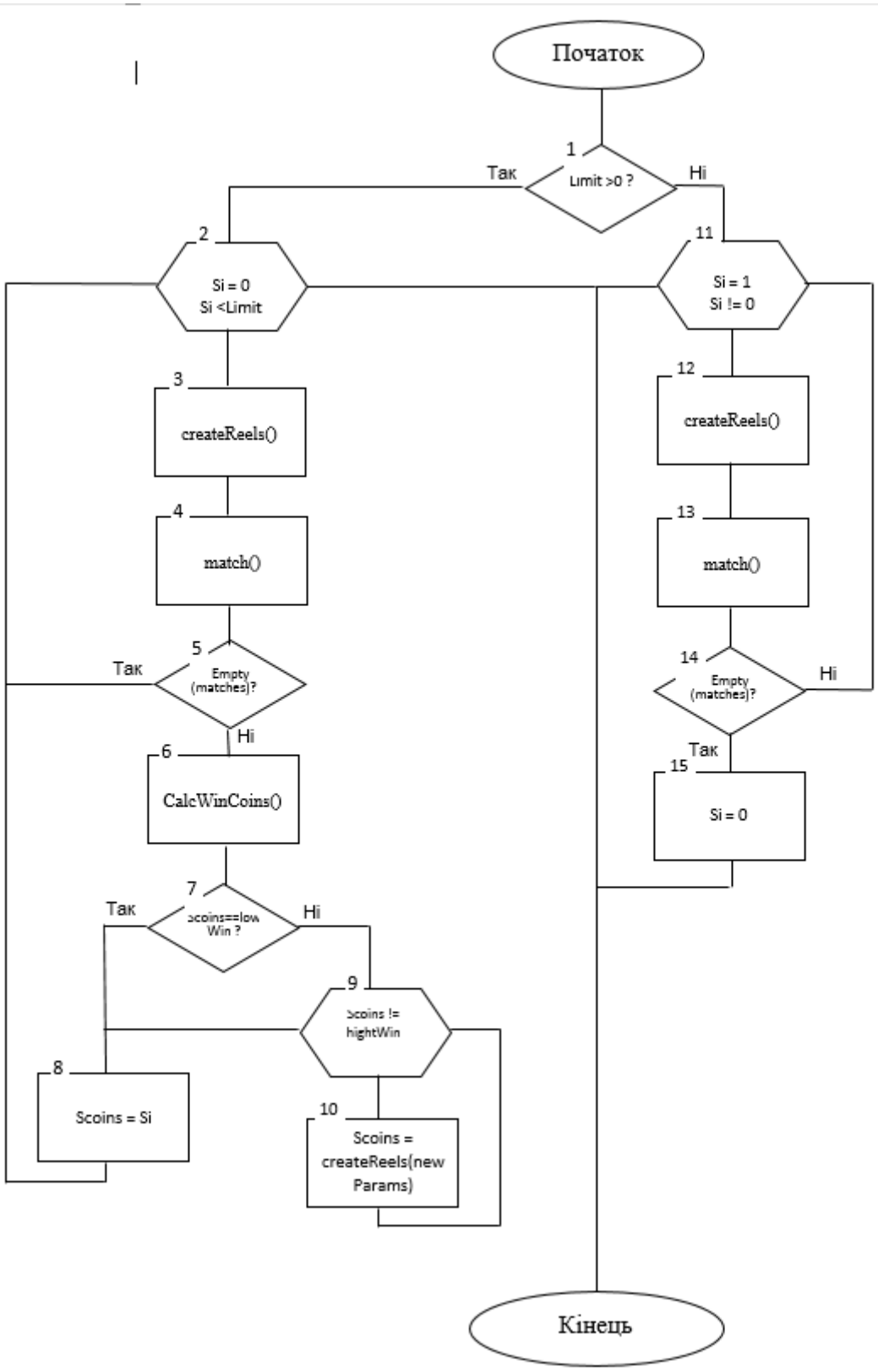

Рисунок 2.6 – Алгоритм генерації виграшної комбінації віртуального ігрового автомата

4. Якщо ліміт рівняється нулю, основний цикл алгоритму буде воторюватиць поки не згенерується програшна комбінація. У випадку програшу, навіть з доступним лімітом для виграшу, гра завершає раунд з від’ємним результатом.

2.6 Висновок до розділу 2

У даному розділі визначено особливості розробленої інформаційної технології, а саме використання сервіс-орієнтованї архітектури, визначена математична модель управління віртуальним ігровим автоматом. Сформульовано структуру програмного продукту та розроблено основні алгоритми для функціонування віртуального ігрового автомата.

Розглянуто і проаналізовано основні підходи сервіс-орієнтованої архітектури проектування основних функціональних модулів. Проаналізовано переваги та недоліки найпопулярніших підходів проектування Web-сервісів, обрано та застосовано REST підхід, перевагами якого є масштабовність системи, що дозволяє їй еволюціонувати з новими вимогами, а також не має прив’язки до одного якось одного формату передачі данних (HTML, XML, JSON). Розглянуто і проаналізовано методи реалізації процесу генерації ігрового поля та виграшних комбінацій.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВІРТУАЛЬНОГО ІГРОВОГО АВТОМАТА

3.1 Обґрунтування вибору мови програмування та середовища розробки

Технології веб-розробок швидко розвивається, і розробники серверів стикаються з вибором між потужними мовами, такими як Java, C, Perl, і сучасними веб-мовами, такими як PHP, JavaScript, Ruby, Go.

PHP - був створений в 1994 році Расмусом Лерфордом. Він створив програмну оболонку (інтерпретатор), яка встановлюється як модуль для веб-сервера Apache або Nginx. Спочатку розроблений як препроцесор гіпертекстової сторінки, PHP можна легко інтегрувати в HTML, однак цей підхід сьогодні не є гарною практикою, але був очевидним для початківців. Це сприяло популярності мови, тому 80% веб-сайтів в Інтернеті написані на PHP [9].

Програма, яка перекладає код, написаний однією мовою програмування, на іншу, називається транслятором. Упорядник також є перекладачем. Перекладає код, написаний мовою високого рівня, у машинний код. Процес компіляції створює двійковий виконуваний файл, який можна запустити без компілятора. Перекладач – це зовсім інша категорія. Інтерпретатор не перекладає в код, а виконує його. Інтерпретатор аналізує програмний код і виконує кожен його рядок. Ви повинні використовувати інтерпретатор кожного разу, коли ви запускаєте цей код. Інтерпретатори значно поступаються компіляторам за продуктивністю, оскільки двійковий код виконується набагато швидше. Але інтерпретатори дозволяють повністю контролювати програму під час її виконання [9]. Що стосується PHP, то він не є ні компілятором, ні інтерпретатором. PHP є чимось середнім між компілятором та інтерпретатором. Сценарій надсилається на вхід PHP. Він перекладає його, перевіряючи синтаксис, у спеціальний байт-код (внутрішнє представлення). Тоді PHP виконує байт-код, а не код самої програми, і не створює виконуваний файл. Байт-код набагато компактніший, ніж звичайний програмний код, що робить його легшим і швидшим для інтерпретації або виконання. Переваги PHP:

- працює на всіх найпопулярніших операційних системах (Windows, MacOS, Linux), має власні версії пакетів розробки на PHP, а це означає, що ви можете створювати сайти на будь-якій з цих операційних систем;

- PHP може працювати в поєднанні з різними веб-серверами: Apache, Nginx, IIS;

- простота і легкість створення. Загалом, маючи невеликий досвід програмування на PHP, ви можете створювати прості веб-сторінки;

- Синтаксис PHP схожий на синтаксис C, тому, знаючи C або одну з мов з подібним синтаксисом, буде легше освоїти PHP;

- PHP підтримує роботу з великою кількістю систем баз даних MySQL, MSSQL, Oracle, Postgre, MariaDB, MongoDB тощо;

- постійний розвиток PHP, з'являються нові версії з новими функціями, адаптація мови програмування до нових завдань. І, як правило, перейти на нову версію не складно.

На даний момент веб-сторінки створюються у двох версіях PHP: старішій версії 5.6 і відносно новій версії 7 [9]. Основні відмінності в інноваціях:

- рефакторинг базових структур даних;

- покращена конвенція для виклику віртуальних машин;

- новий API параметрів розбору;

- новий менеджер пам'яті;

- численні вдосконалення виробників віртуальних машин;

- значно зменшено споживання пам'яті;

- покращені функції __call () і __callStatic ();

- покращене з'єднання струн;

- покращений пошук символів у рядках;

Фреймворки - це програмне забезпечення, яке спрощує створення та обслуговування технічно складних або інтенсивних проектів. Фреймворк, як правило, містить лише основні програмні модулі, а на їх основі всі специфічні для проекту компоненти реалізує розробник, завдяки чому досягається не тільки висока швидкість розробки, але й висока продуктивність та надійність рішень. Структура відрізняється від бібліотеки тим, що бібліотека може використовуватися в програмному продукті

просто як набір підсистем зі схожою функціональністю, не зачіпаючи архітектуру основного програмного продукту і не накладаючи на нього будь-яких обмежень. Фреймворк диктує правила побудови архітектури програми, встановлення поведінки за замовчуванням на початковому етапі розробки, створення фреймворка, який потрібно буде розширити та модифікувати відповідно до цих вимог. Структура може включати інструменти, бібліотеки, мови сценаріїв та інше програмне забезпечення, яке полегшує створення та інтеграцію різних компонентів великого програмного проекту.

Laravel - це безкоштовний PHP фреймворк з відкритим вихідним кодом, створений Тейлором Отвеллом і призначений для створення веб-додатків відповідно до шаблону модель — перегляд — контролер (MVC) [9]. Laravel пропонує модульну систему пакування зі спеціальним менеджером залежностей, різними способами доступу до реляційних баз даних, інструментами, які допомагають реалізувати та підтримувати програму, а також багатий набір функцій, які охоплюють основні функції структур PHP.

Laravel використовує Eloquent ORM для взаємодії з базою даних, що є дуже простою реалізацією шаблону проекту Active Record [10]. Використання ORM дозволяє уникнути написання запитів у SQL. Його суть полягає в тому, що кожна сутність бази даних має свою об’єктну модель. Програміст модифікує цю модель, а він у свою чергу взаємодіє з базою даних. Використання ORM набагато компактніше і швидше, ніж написання запитів SQL. При створенні будь-якого веб-додатка вам завжди потрібно писати багато подібних запитів до бази даних. З іншого боку, ORM уникає написання повторюваних кодів. Більше того, можна легко підтримувати цілісність бази даних не тільки на рівні бази даних, а й на рівні коду [10]. Це значно підвищує надійність програми. Однак ORM не завжди використовує найкращі параметри електронних таблиць, що збільшує навантаження на сервер. Дуже часто вибирається велика кількість даних, що також знижує швидкість запиту. Однак ці недоліки цілком виправдовуються тим, наскільки використання об'єктно-орієнтованої моделі збільшує швидкість розробки.Eloquent ORM забезпечує механізм оптимізації запитів до бази даних під назвою «Активний запис», який призначений

для вирішення проблеми надлишкових запитів. Відношення буде активно завантажуватися, якщо воно було вказано під час виклику методу with (). Таким чином, усі необхідні дані будуть отримані за допомогою мінімальної кількості запитів. Вам не потрібно буде знову звертатися до бази даних, оскільки вони вже будуть присутні в моделі.

3.2 Налаштування середовища розробки для віртуального ігрового автомата

При створенні будь-якої системи слід використовувати апаратне та програмне забезпечення, подібне до тих, які будуть використовуватися в готовій системі. При розробці використовувався Vagrant, щоб максимально наближати умови роботи до реальних.

Vagrant – продукт компанії HashiCorp, що спеціалізується на інструментах для автоматизації розробки та експлуатації. Він дозволяє створювати та конфігурувати легковагі, повторювані та переносні оточення для розробки.

Перше завдання будь-якого розробника на новому проекті – розгорнути оточення. Як правило, вона зводиться до наступних кроків:

- клонувати репозиторій із проектом;
- поставити пакети для роботи (наприклад, бібліотеку для xml);
- встановити додаткові програми, такі як база даних;
- правильно настроїти параметри конфігурації.

Без автоматизації подібний процес може тривати як годинник, і цілі дні, залежно від складності проекту. Причому для кожної людини у команді. Для кожної зміни робочої машини або після переустановки системи. Проблема ускладнюється тим, що чим більше розробників на проекті, тим різноманітніші машини вони використовують, включаючи різні операційні системи. У такому разі процес налаштування гарантовано буде різним, що є ще одним джерелом проблем. Весь процес налаштування має бути десь описаний, але, як правило, ці описи швидко

застарівають, і їх мало хто хоче підтримувати, а будь-які спроби підтримувати актуальність згасають.

Після додавання пари таких проектів операційна система стає сильно захащеною. Навіть якщо ви не працюєте над проектом, після увімкнення комп'ютера почнуть стартувати сервіси, які потрібні тільки для розробки.

Ще одна проблема – зміни. Практично будь-яке переналаштування встановленої конфігурації потребує ручного втручання, доустановки пакетів, програм або зміни конфігів.

І останнє: нерідко потрібна можливість розгорнути проект і для непрограмістів, наприклад верстальників або тестувальників. Наявність навіть добре описаного процесу встановлення мало їм допоможе.

Тепер можна спробувати сформулювати вимоги до ідеального оточення:

- ізольованість. Таким чином уникають можливі конфлікти з іншими оточеннями (наприклад, основною системою), і основна система залишається чистою;

- повторюваність. Перестворити робоче середовище можна за лічені хвилини, набравши буквально одну команду. Будь-яка зміна поширюється одразу для всіх;

- переносність. Оточення розгортається під будь-якою системою одним універсальним способом.

Vagrant створений для вирішення цих завдань. Багато в чому його робота спирається на віртуалізацію, для якої за умовчанням використовується продукт VirtualBox. Але, на відміну від звичайної роботи з віртуальною машиною, коли в ній стоїть система з графічною оболонкою, Vagrant створює віртуальну машину, доступну тільки в термінальному режимі (через командний рядок), при цьому сама розробка триває на хост-машині, а ось запуск коду виконання відбувається всередині машини. Іншими словами, редактор ставиться на вашу основну систему, і код також лежить в ній. Vagrant прозоро прокидає код усередину машини та дозволяє його запускати.

Використання Vagrant передбачає деяке налаштування операційної системи та наявність певних знань:

- потрібно знати, що таке віртуалізація, та як працювати з віртуальними машинами;

- командний рядок – єдиний спосіб взаємодії з Vagrant. Якщо у користувача операційна система Windows, то необхідно налаштувати її;

- базове знання мереж, концепції портів;

- на комп'ютері має бути хоча б 4 гігабайти оперативної пам'яті, хоча навіть із таким обсягом працювати проблематично. Мінімально комфортний рівень – 8 гігабайт.

Vagrant — безкоштовне програмне забезпечення з відкритим кодом для створення та налаштування віртуального середовища розробки. Крім того, фреймворк Laravel надає власний програмний пакет для Vagrant - Homestead, який включає все необхідне програмне забезпечення. Щоб розгорнути Homestead, виконайте такі дії:

- встановити Virtual Box

- встановити Vagranta

- відкрийте консоль

- перейдіть до каталогу проекту

- виконати команди: vagrant box add laravel / homestead, git clone https://github.com/laravel/homestead.git Homestead, bash init.sh.

Homestead налаштовується шляхом редагування файлу Homestead.yaml. Можна встановити такі параметри:

- вказати IP-адресу віртуального сервера;

- вказати обсяг пам'яті;

- визначити кількість ядер;

- вказати провайдера для створення віртуальної машини;

- вказати шляхи для ключів ssh для використання зовнішнього сховища для контролю версій;

- порівняти шляхи проекту на віртуальній машині та хост-машині;

- надати імена баз даних.

Приклад файлу Homestead.yaml показано на рисунку 3.1.

Рисунок 3.1 – Приклад файлу Homestead.yaml

Після налаштування віртуальної машини потрібно додати адресу віртуального сервера до файлу hosts в операційній системі. Це робиться для того, щоб адреса сервера була доступна службі DNS. У Windows 10 цей файл знаходиться в каталозі C: \ Windows \ System32 \ drivers \ etc. Просто додайте терміни «192.168.10.10 gry.test» і «192.168.10.10 dispatcher.test». Потім запустіть команду vagrant up у каталозі ~ \ Homestead. Після цього веб-сайти будуть запущені і будуть доступні за адресами, введеними у файлі hosts.

3.3 Структура модулів програмного забезпечення віртуального ігрового автомата

При створенні web-сервісів системи віртуальних машин головне завдання – чітко і правильно розділити всі функції на окремі модулі, які можна буде зручно використовувати та масштабувати в майбутньому, навіть після додавання кількох різних машин. Для виконання цього завдання було створено 2 веб-сайти з такими модулями:

- модуль маршрутизації для створення посилання для відображення клієнта машини та єдиної точки відправлення клієнту всіх HTTP-запитів;

- модуль дисплея клієнта торгового автомата;

- модуль розмежування ігрової ситуації (диспетчер);

- конфігураційний модуль для налаштування списку ігрових автоматів та базових рамок реакції сервера на ситуацію в грі;

- модуль ігрових ситуацій та логіка опрацювання їх машиною;

- Комунікаційний модуль бази даних, що зберігає інформацію про ігрові ситуації, побудований на основі шаблону програмування DataMapper;

- додаткові універсальні сервіси для розрахунку грошових операцій та створення стандартної відповіді від сервера;

- модуль для створення та перевірки сеансів та керування лімітами машини.

Зв'язок між програмними модулями показано на рисунку 3.2.

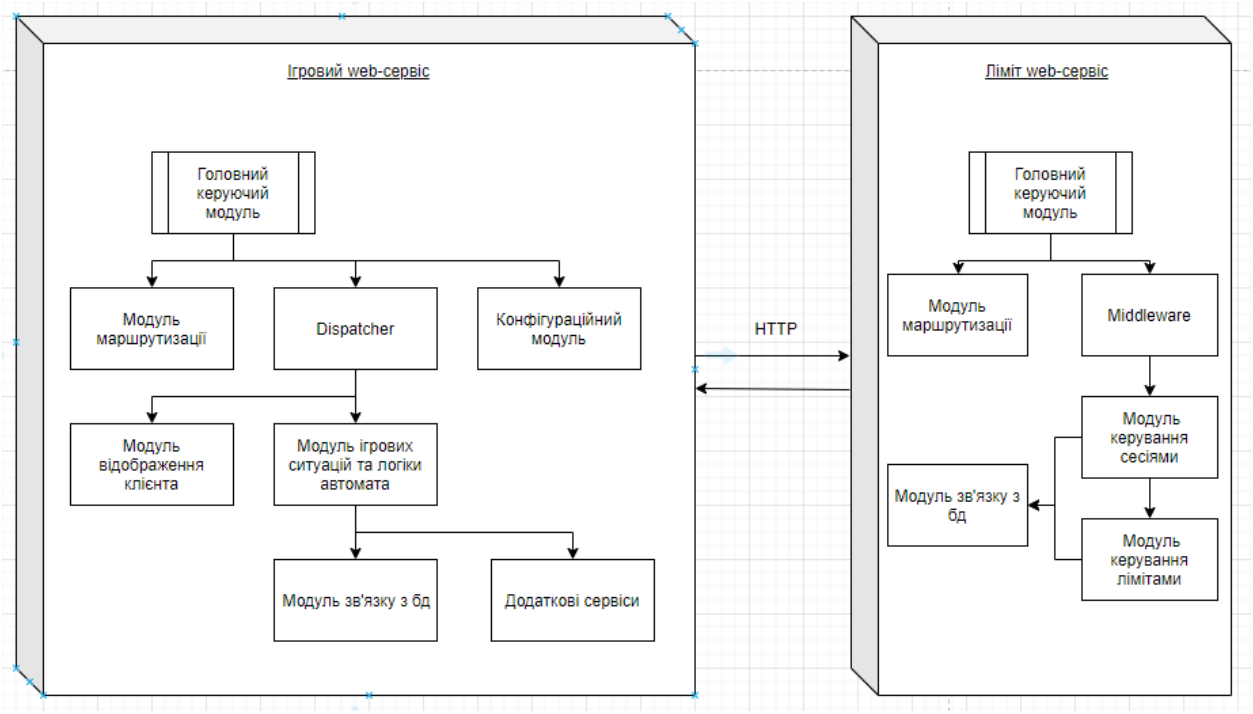

Рисунок 3.2 – Схема модулів віртуального ігрового автомата

Схема підключення модуля показує, що додаток матиме як необхідну функціональність, так і можливість легкого розширення. Поділ функцій також добре видно на схемі. Сервіс ігрової мережі відповідає за створення та керування різними

ігровими ситуаціями та слотами загалом, а онлайн-сервіс лімітів керує сесіями гравців та розрахунком ліміту наступного виграшу. Ці служби обмінюються інформацією через HTTP-запити. Оскільки орган запиту часто передає персональні дані та результати грошових операцій, потрібна аутентифікація HTTP.

Основна схема аутентифікації HTTP визначена в RFC 7617, який передає облікові дані у вигляді пар слот/пароль, закодованих base64. Код підтвердження запиту показано нижче:

```
$validateGameApiRequest = function ($ request, $ response, $next) {
    $authHeader = $request -> getHeader('X-AUTH');
    if ( ! $authHeader ) {
        return $response -> withJson ([
            'errors '=> [
                ['status' => '402', 'detail' => 'No auth header']
            ]
        ], 402);
    }
    list($authType, $authKey) = explode(' ', $authHeader[0]);
    if ( $authKey !=                                    = base64_encode(getenv('GAME_
API_USERNAME') . ':' . getenv('GAME_API_PASSWORD')) ) {
        return $ response->withJson([
            'errors'=>[
                ['status' => '402', 'detail' => 'Unauthorized']
            ]
        ], 402);
    }
    $data = $request->getParsedBody();
    if ( ! isset($data['token']) ) {
        return $response->withJson([
            'errors' => [
                ['status' => '432', 'detail' => 'No token provided']
            ]
        ], 432);
    }
}
```

Оскільки ідентифікатор машини та пароль надсилаються по мережі відкритим текстом, хоча вони кодуються base64, цей механізм є оборотним, тому також необхідно використовувати сертифікат SSL.

3.4 Програмна реалізація основних модулів віртуального ігрового автомата

Першим кроком після взаємодії користувача з ігровим автоматом, тобто після натискання кнопки «Spin the drum», є підрахунок максимально дозволеного виграшу. Для цього метод GET використовується для відправки HTTP-запиту до мережевого сервісу, який керує сеансами та обмеженнями:

```
 $guzzle = new Guzzle();
 request data = [
      'token' => $session['attributes' ]['token'],
      'spin_id' => $spin-›id,
      'amount' => $bet[ 'amount '],
 ];
 try {
  $api _request = $guzzle-›get ("($container['settings']['api_url']}balance/bet"
  [
'headers'=> [
             'X-AUTH' => 'Basic c2VjcmV0OnNlY3]ldA=='
             'Cache-Control'=> 'no-cache',
      'Content-Type' => 'application/json'
],
'json' => request data
  ]);
 $response _data = json_decode($api_request-›getBody(), TRUE) ['data'];
 } catch (\GuzzleHttp\ Exception\RequestException $e) {
  $container['starburst_log']-›error('', [
'request'=> json decode($e-›getRequest ()-›getBody(), TRUE),
'response'=> json decode ($e-›getResponse()-›getBody(), TRUE),
  ]);
 return \GameError::send(0) ;
 }
```

Ліміт виграшу розраховується за допомогою SQL-запиту:

```
$sal - "SELECT
       (SELECT IFNULL (SUM(amount), 0)
       FROM transactions
       WHERE created at > CURDATE()
       AND status = 1
       AND type = 0) AS 'total bets',
       (SELECT IFNULL (SUM(amount), 0)
       FROM transactions
       WHERE created _at > CURDATE()
       AND status = 1
       AND type = 1) AS 'total wins'
";
$stmt = $this-›container['db']-›prepare($sql);
$stmt-›execute();
$result = ($stmt-›fetch ());
```

База даних збирає інформацію про ставки та виграші користувача за поточний день. При використанні методів IFNULL і SUM значення підсумовується або за замовчуванням дорівнює нулю, якщо порожнє. Потім обчислюється і повертається максимально можливий наступний виграш:

```
       $next_max_win=$result['total_bets']*(1 - $limit-
>limit/100) - $result['total_wins'];
       return $next_max_win > 0 ? round($next_max_win, 2, PHP_ROUND_HALF_DOWN) : 0;
```

Другим кроком є алгоритм створення ігрового поля, генерування ігрових комбінацій та визначення суми виграшу.

```
$limit = $response_data['attributes']['max_next_win'];
$lowReelsPercents = false;
```

На основі значення ліміту запускається цикл, який триватиме до тих пір, поки не буде створена ситуація програшу в разі нульового максимального значення виграшу, або:

```
if ($limit > 0) {
 do {
```

генеруємо ігрове поле для кліїнта:

```
   $reels = $this->logic->generateReels($lowReelsPercents);
```

розраховуємо бонусний безоплатний раунд:

```
$overlaied = $this->logic->overlayReels($reels);
$respin = $overlaied['respin'];
```

та перевіряємо наявність на полі виграшної комбінації:

```
$matches = $this->logic->match($overlaied['reels']);
```

У випадку, коли виграшних комбінацій немає – виходимо з циклу і закінчуємо роботу алгоритму.

```
if (empty($matches)) {
  $i = 0;
  break;
}else{
```

Якщо все наявні виграшні комбінації уже присутні, тоді розраховуємо суму потенційного виграшу для даного раунду,

```
$win_coins = $this->logic->calculateWonCoins($matches);
$bet_results = $container['moneyManager']-
>countWinning($betlevel, $denomination, $win_coins);
```

І перевіряємо тип виграшної комбінації, якщо тип «середній» чи «низький», то закінчуємо роботу алгоритму, тим самим дозволяємо користувачеві виграти,

```
if ($this->determineWinType($bet_results['amount']) === 'low' ||
    $this->determineWinType($bet_results['amount']) === 'avarage') {
    $i = $bet_results['amount'];
} else {
```

Але якщо тип виграшу «високий», встановлюємо прапорець зниження коефіцієнтів для генерації «дорогих елементів», та проходимо цикл іще один раз

```
    $lowReelsPercents = true;
$i = $limit + 1;
    }
  }
 } while($i > $limit) ;
} elseif ($limit == 0) {
  $i = 1;
  while($i != 0) {
  $reels = $this-›logic-›generateReels();
  Soverlaied = $this-›logic-›overlayReels ($reels);
  $respin = $overlaied['respin'];
  $matches = $this-›logic-›match ($overlaied 'reels']);
  if (empty($matches)) f
$1 = 0;
  }
```

```
}
}
```

Після цього, наступним кроком є налаштування тіла відповіді клієнта та обробка результатів відповідно до базових параметрів, визначених у модулях конфігурації. У процесі цього етапу, система опрацьовує вхідні дані та встановлює необхідні значення для подальших кроків гри.

У випадку, якщо змінна $respin має позитивне значення, встановлюється певний знак для поточного раунду. Після цього може наступити бонусний раунд, проте обмеженням є можливість проведення лише трьох бонусних раундів. Сценарій гри під час бонусного раунду ідентичний звичайному раунду, за винятком того, що виграші можуть накопичуватися, роблячи гру ще цікавішою для гравців.

Усі отримані результати та важливі дані щодо гри зберігаються в базі даних, що дозволяє зберігати та відстежувати хід гри, а також накопичувати статистику для подальших аналізів і вдосконалення геймплею. Такий алгоритм дозволяє створювати захоплюючий та динамічний геймінговий досвід для користувачів.

```
$won _coins = $this-›logic-›calculateWonCoins($matches);
$freespin results = $container['moneyManager']->countWinning(
       $spin-›betlevel,
       $spin-›denomination,
       $won coins
);
$previous _wins = $freespin_mapper-›sumWinAmount($spin->id);
if (lis_null ($spin win)){
      $i = $spin win-›win_amount + previous wins + $freespin results['amount'];
} else {
      $i = $previous wins + $freespin _results['amount'];
}
if ($this->determineWinType ($freespin results[' amount']) !== 'high") {
$lowReelsPercents = true;
      $i =$limit + 1;
}
```

3.5 Тестування програмного забезпечення віртуального ігрового автомата

Головне вікно програмного забезпечення віртуального ігрового автомата складається з декількох частин: основне ігрове поле (матриця розміром 3x5); блок вибору розміру початкової ставки, блок визначення вартості ігрової монети та кнопки керування запуском барабанів ігрового поля (рис. 3.3).

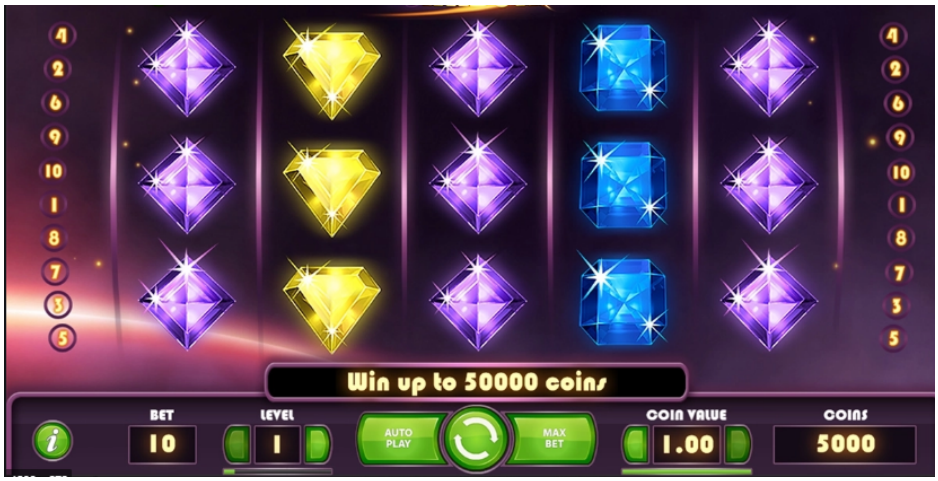

Рисунок 3.3 – Головне вікно програмного забезпечення віртуального ігрового автомата

В програмі наявна спеціальна кнопка, що відкриває доступ до сторінки з описом усіх функцій ігрового автомата та правил гри (рис. 3.4).

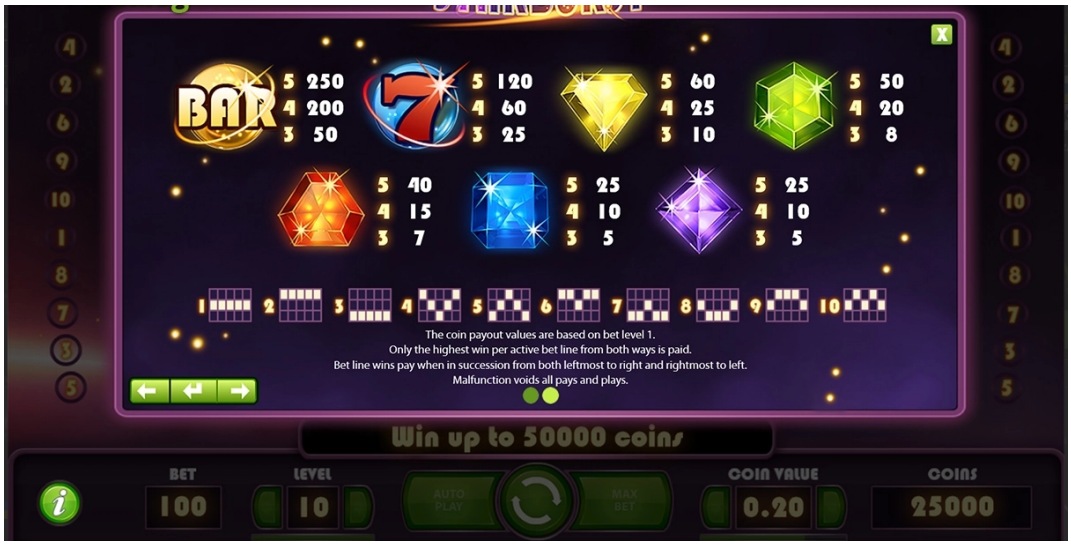

Рисунок 3.4 – Вікно з правилами гри ігрового автомата

Якщо гравець відкриє вікно з правилами гри (рис. 3.4), натиснувши на певну кнопку, то він побачить вікно із списком всіх ігрових елементів. Виграшна комбінація кожного з яких може бути в 3 або 4 або 5 елементів підряд. Кожна комбінація, звичайно, має свою вартість, що вимірюється в ігрових монетах. Інформація про цей список зберігається на сервері гри, і на основі цього списку формується ігрове поле для гравця (рис. 3.5). Кожний елемент має певну ймовірність появи, так як вага кожної комбінації буде різною.

В нижній частині вікна наявний список із десяти ліній, по яким можуть випадати виграшні комбінації. Даний віртуальний ігровий автомат може формувати такі комбінації як зліва, так і справа. У разі виграшу, гравцеві буде зарахована найдорожча комбінація (рис. 3.5). Список таких виграшних ліній тежзберігається на сервері.

Рисунок 3.5 – Виграшна комбінація віртуального ігрового автомата

При генерації ігрового поля з невеликою ймовірністю може генеруватись спеціальний елемент – wild, що при порівнянні елементів на виграшній лінії замінює собою будь який з них. За даними правилами, wild може генеруватись лише на 2, та/або 3, та/або 4 барабанах (рис. 3.6).

Рисунок 3.6 – Виграшна комбінація зі спеціальним елементом wild

У випадку, коли на всіх барабанах гри з'являються тільки символи Wild, гравець отримує унікальну можливість: додаткову преміальну прокрутку ігрових барабанів. Ця подія вносить екстра-елемент в гру, роблячи її ще захопливішою для гравців.

Під час преміального раунду залишається чинними всі основні правила гри, проте з певною варіацією. Попередньо згенерований барабан залишається незмінним протягом цього бонусного раунду, дозволяючи гравцеві зберігати певний стан гри, який може призвести до більших виграшів або захопливих можливостей.

Важливо відзначити, що кількість преміальних раундів обмежена - гравець може скористатися цією можливістю лише тричі під час гри. Це додає стратегічний елемент до геймплею, спонукаючи гравців обдумано використовувати преміальні раунди для максимізації свого виграшу в грі. Такий підхід створює враження гри з несподіваними поворотами і допомагає забезпечити емоційно насичений геймінговий досвід.

З метою тестування програмного забезпечення віртуального ігрового автомата було проведено реєстрацію для 5 гравців, при цьому кожний гравець провів по 10 ігрових раундів. Відповідно до проведених досліджень порівняємо розроблене програмне забезпечення з раніше розглянутими програмами аналогами за наступними критеріями: зручна реєстрація в системі, зацікавленість у грі, ймовірність виграшу ігровим автоматом. Порівняння наведено у таблиці 3.1.

Таблиця 3.1 – Порівняння програм для керування соціальною мережею

Назва програми	Зручна реєстрація	зацікавленість у грі	ймовірність виграшу ігровим автоматом
TwinSpin	+	+	91 %
Pyramid	-	+	93 %
Розроблений ігровий автомат	+	+	96,06 %

Таким чином в результаті тестування програмного забезпечення віртуального ігрового автомата відповідно до порівняльної таблиці, можна зробити висновок про те що розроблений ігровий автомат на 3,06% має більшу ймовірність виграшу автоматом у гравця, що свідчить про те що мету дослідження було досягнуто.

3.6 Висновок до розділу 3

У даному розділі було обгрунтовано вибір мови програмування та середовища розробки. В ході даного аналізу для розробки програмного забезпечення віртуального ігрового автомата було вибрано мову програмування PHP, Slim і Laravel Framework, в середовищі програмування Visual Studio Code. Для тестування програмного забезпечення було встановлено віртуальну машину Homestead. Також детально описано програмну реалізацію основних модулів віртуального ігрового автомата: генерація ігрового поля для нового гравця, пошук виграшної комбінації, HTTP аутентифікація нового гравця.

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота «Інформаційної технології віртуального ігрового автомата» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;

2) розраховано витрати на здійснення науково-технічної розробки;

3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія віртуального ігрового автомата» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в таблиці 4.1 [15].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	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 працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато 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, ніж в	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
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 на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження таблиці 4.1

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати 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	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	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
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 4.2.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	4	5
2. Ринкові переваги (наявність аналогів)	2	2	3
3. Ринкові переваги (ціна продукту)	4	4	3
4. Ринкові переваги (технічні властивості)	3	3	4
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	4	3	3
7. Ринкові перспективи (конкуренція)	3	2	2
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	4	5	5
11. Практична здійсненність (термін реалізації)	5	4	5
12. Практична здійсненність (розробка документів)	5	5	4
Сума балів	44	42	44
Середньоарифметична сума балів $СБ_с$	43,3		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в таблиці 4.3 [19].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія віртуального ігрового автомата» становить 43,3

бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [19]:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i , \tag{4.1}$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу *і-го* технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має виконуватись умова $\sum_{i=1}^{k} \alpha_i = 1$;

β_i – відносне значення *і-го* технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ні}}{I_{аі}} , \tag{4.2}$$

де $I_{ні}$ та $I_{на}$ – чисельні значення конкретного *і*-го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ні}}; \quad (4.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Тактова частота процесора не нижче	Ггц	3,2	1,2	2,67	0,25
Оперативна пам'ять не менше	Мб	8	4	2	0,3
Об'єм диску	Гб	50	20	2,5	0,15
Рівень візуалізації	бал	8	9	1,13	0,1
Швидкість мережевого каналу	Мб/с	100	50	2	0,2

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення складе:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 2{,}67 \cdot 0{,}25 + 2 \cdot 0{,}3 + 2{,}5 \cdot 0{,}15 + 1{,}13 \cdot 0{,}1 + 2 \cdot 0{,}2 = 2{,}16.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,16 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія віртуального ігрового автомата», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [19]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \tag{4.4}$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=21 дні.

$$З_о = 21000{,}00 \cdot 58 / 21 = 58000{,}00 \text{ } грн.$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник науково-дослідної роботи з розробки і дослідження інформаційної технології віртуального ігрового автомата	21000,00	1000,00	58	58000,00
Програміст (інженер-програміст 1-ї категорії)	18600,00	885,71	58	51371,43
Консультант (аналітик математичного забезпечення процесу)	20000,00	952,38	10	9523,81
Технік 1-ї категорії	10000,00	476,19	48	22857,14
Всього				141752,38

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія віртуального ігрового автомата» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i, \tag{4.5}$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

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

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $M_м$=6700,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [19];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 6700{,}00 \cdot 1{,}10 \cdot 1{,}35 / (21 \cdot 8) = 59{,}22$ *грн.*

$З_{р1} = 59{,}22 \cdot 8{,}00 = 473{,}79$ *грн.*

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (4.7)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 12%.

$З_{дод} = (141752{,}38 + 3544{,}35) \cdot 12 / 100\% = 17435{,}61$ *грн.*

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Організація робочих місць учасників проекту з розробки і дослідження інформаційної технології віртуального ігрового автомата	8,00	2	1,10	59,22	473,79
Установка обчислювального обладнання	6,00	3	1,35	72,68	436,10
Інсталяція програмного забезпечення проекту	5,40	5	1,70	91,53	494,24
Підготовка бази даних віртуальної системи	14,32	2	1,10	59,22	848,08
Проведення обчислювального експерименту	7,00	4	1,50	80,76	565,31
Тестування віртуальної системи	10,00	3	1,35	72,68	726,83
Всього					3544,35

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (4.8)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (141752,38 + 3544,35 + 17435,61) · 22 / 100% = 35801,11 грн.

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія віртуального ігрового автомата».

Витрати на матеріали ($М$), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj}, \quad (4.9)$$

де $Н_j$ – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ *= 1,1 ... 1,15*);

$В_j$ – маса відходів j-го найменування, кг;

$Ц_{вj}$ – вартість відходів j-го найменування, грн/кг.

$М_1$ = 3,0 · 202,00 · 1,05 - 0 · 0 = 636,30 грн.

Проведені розрахунки зведемо до таблиці 4.7.

Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Інформаційна технологія віртуального ігрового автомата», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j \quad (4.10)$$

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір	202,00	3,0	0	0	636,30
Папір для записів	111,00	3,0	0	0	349,65
Органайзер офісний	310,00	3,0	0	0	976,50
Канцелярське приладдя (набір офісного працівника)	195,00	2,0	0	0	409,50
Картридж для принтера	2110,00	1,0	0	0	2215,50
Диск оптичний	25,00	6,0	0	0	157,50
USB-пам'ять Kingston 128 GB	299,00	2,0	0	0	627,90
Тека для паперів	84,00	4,0	0	0	352,80
Всього					5725,65

де H_j – кількість комплектуючих j-го виду, шт.;

$Ц_j$ – покупна ціна комплектуючих j-го виду, грн;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$).

$K_в = 1 \cdot 2239,00 \cdot 1,05 = 2350,95$ *грн.*

Проведені розрахунки зведемо до таблиці 4.8.

Таблиця 4.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
ДЖОЙСТИК LOGITECH EXTREME 3D PRO (942-000031)	1	2239,00	2350,95
Всього			2350,95

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (4.11)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.i}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань устаткування.

$B_{спец} = 10335,00 \cdot 1 \cdot 1,05 = 10851,75$ *грн.*

Отримані результати зведемо до таблиці 4.9.

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
МАРШРУТИЗАТОР MIKROTIK RB4011IGS+5HACQ2HND-IN	1	10335,00	10851,75
Всього			10851,75

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.12)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань програмних засобів.

$B_{прг} = 300,00 \cdot 1 \cdot 1,1 = 330,00$ грн.

Отримані результати зведемо до таблиці 4.10.

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Вільне та відкрите програмне забезпечення для створення та конфігурування віртуального середовища розробки Vagrant	1	300,00	330,00
Середовище програмування Visual Studio Code	1	7560,00	8316,00
Всього			8646,00

Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12} , \qquad (4.13)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$Т_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$А_{обл} = (43599,00 \cdot 2) / (3 \cdot 12) = 2422,17 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.11.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер AMD Ryzen5000/RAM8Mb/SSD500 Gb	43599,00	3	2	2422,17
Робоче місце розробника ПЗ	9600,00	5	2	320,00
Пристрої виводу (візуалізації) інформації	14800,00	4	2	616,67
Оргтехніка	8450,00	4	2	352,08
Приміщення лабораторії розробки та дослідження інформаційної технології	355000,00	20	2	2958,33
ОС Windows 11	6120,00	3	2	340,00
Прикладний пакет Microsoft Office 2021	5700,00	3	2	316,67
Обладнання передачі цифрових даних даних	9600,00	5	2	320,00
Всього				7645,92

Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (4.14)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e = 7,50$ грн;

$K_{вni}$ – коефіцієнт, що враховує використання потужності, $K_{вni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$B_e = 0,35 \cdot 400,0 \cdot 7,50 \cdot 0,95 / 0,97 = 1050,00$ *грн.*

Проведені розрахунки зведемо до таблиці 4.12.

Таблиця 4.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер AMD Ryzen5000/RAM8Mb/SSD500Gb	0,35	400,0	1050,00
Робоче місце розробника ПЗ	0,08	400,0	240,00
Пристрої виводу (візуалізації) інформації	0,12	8,0	7,20
Оргтехніка	0,32	3,5	8,40
Обладнання передачі цифрових даних даних	0,15	380,0	427,50
Всього			1733,10

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія віртуального ігрового автомата» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \quad (4.15)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св} = 0\%$.

$B_{св} = (141752,38 + 3544,35) \cdot 0 / 100\% = 0,00$ грн.

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_о + З_р) \cdot \frac{Н_{сп}}{100\%}, \quad (4.16)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп} = 30\%$.

$B_{сп} = (141752,38 + 3544,35) \cdot 30 / 100\% = 43589,02$ грн.

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

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

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.17)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 60%.

$I_в$ = (141752,38 + 3544,35) · 60 / 100% = 87178,04 грн.

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

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

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.18)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 120%.

$В_{нзв}$ = (141752,38 + 3544,35) · 120 / 100% = 174356,07 грн.

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія віртуального ігрового автомата» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + B_{спец} + B_{прг} + A_{обл} + B_е + B_{св} + B_{сп} + I_в + B_{нзв}. \quad (4.19)$$

$B_{заг}$= 141752,38 +3544,35 +17435,61 +35801,11 +5725,65 +2350,95 +10851,75 + 8646,00 + 7645,92 +1733,10 +0,00 +43589,02 +87178,04 +174356,07 = 540609,95 грн.

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

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.20)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,9.

ЗВ = 540609,95 / 0,9 = 600677,72 грн.

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

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

Результати дослідження проведені за темою «Інформаційна технологія віртуального ігрового автомата» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, приймемо 45000 осіб;

$Ц_б$ – вартість доступу до програмного продукту у році до впровадження результатів розробки, приймемо 250,00 грн;

$\pm\Delta Ц_о$ – зміна вартості доступу до програмного продукту від впровадження результатів науково-технічної розробки, приймемо 88,88 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [19]:

$$\Delta П_i = (\pm\Delta Ц_о \cdot N + Ц_о \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.21)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо ρ =40%;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1$ =(88,88·45000,00+338,88·5000)·0,83·0,4·(1-0,18/100%)=1550066,50 грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2$ =(88,88·45000,00+338,88·13000)·0,83·0,4·(1-0,18/100%)=2288109,14 грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3$ =(88,88·45000,00+338,88·23000)·0,83·0,4·(1-0,18/100%)=3210662,44 грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4$ =(88,88·45000,00+338,88·30000)·0,83·0,4·(1-0,18/100%)=3856449,75 грн.

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

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.22)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=0,15;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$ПП$ =1550066,50/(1+0,15)^1+2288109,14/(1+0,15)^2+3210662,44/(1+0,15)^3+
+3856449,75/(1+0,15)^4=1347883,91+1730139,24+2111062,67+2204937,66=7394023,48 грн.

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \quad (4.23)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 600677,72 грн.

$PV = k_{інв} \cdot ЗВ = 2 \cdot 600677{,}72 = 1201355{,}44$ грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \tag{4.24}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 7394023,48 грн;

PV – теперішня вартість початкових інвестицій, 1201355,44 грн.

$E_{абс} = ПП - PV = 7394023{,}48 - 1201355{,}44 = 6192668{,}04$ грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{4.25}$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 6192668,04 грн;

PV – теперішня вартість початкових інвестицій, 1201355,44 грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+6192668{,}04/1201355{,}44)^{1/4} = 0{,}58.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \tag{4.26}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d =0,1;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,4.

$\tau_{мін}$= 0,1+0,4 = 0,5 < 0,58 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія віртуального ігрового автомата» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \tag{4.27}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$= 1 / 0,58 = 1,74 р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновки до розділу 4

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія віртуального ігрового автомата» становить 43,3 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,16 рази.

Також термін окупності становить 1,74 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія віртуального ігрового автомата».

ВИСНОВКИ

Під час виконання магістерської кваліфікаційної роботи було розроблено інформаційну технологію віртуального ігрового автомата.

В роботі проаналізовано предметну область, розглянуто: основні поняття інформаційної технології віртуального ігрового автомата; процеси та етапи, які будуть використовуватись при організації технології віртуального ігрового автомата. Проаналізувавши об’єкт дослідження визначено: ключові вимоги до інформаційної технології, вхідні та вихідні дані для програмних засобів та вимоги до програмно-апаратного забезпечення системи. У результаті аналізу систем аналогів визначено їхні характеристики, основні критерії, переваги та недоліки, область використання, розглянуто основі проблеми, які виникають при розробці подібних систем та технології за допомогою яких, можна їх усунути. Визначено особливості розробленої інформаційної технології, а саме використання сервіс-орієнтованї архітектури, визначена математична модель управління віртуальним ігровим автоматом. Сформульовано структуру програмного продукту та розроблено основні алгоритми для функціонування віртуального ігрового автомата. Розглянуто і проаналізовано основні підходи сервіс-орієнтованої архітектури проектування основних функціональних модулів. Проаналізовано переваги та недоліки найпопулярніших підходів проектування Web-сервісів, обрано та застосовано REST підхід, перевагами якого є масштабовність системи, що дозволяє їй еволюціонувати з новими вимогами, а також не має прив’язки до одного якось одного формату передачі данних (HTML, XML, JSON). Розглянуто і проаналізовано методи реалізації процесу генерації ігрового поля та виграшних комбінацій.

Обгрунтовано вибір мови програмування та середовища розробки. В ході даного аналізу для розробки програмного забезпечення віртуального ігрового автомата було вибрано мову програмування PHP, Slim і Laravel Framework, в середовищі програмування Visual Studio Code. Для тестування програмного забезпечення було встановлено віртуальну машину Homestead. Було доведено, що

розроблений ігровий автомат на 3,06% має більшу ймовірність виграшу автоматом у гравця, ніж його аналоги.

В розділі економічної частини проведено оцінювання комерційного потенціалу розробки інформаційної технології оптичного розпізнавання тексту, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, які склали 600677,72 грн.

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія віртуального ігрового автомата» становить 43,3 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,16 рази.

Також термін окупності становить 1,74 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже всі поставлені задачі було виконано, мету роботи досягнуто.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Д.А. Крамаренко, В.С. Озеранський. Розробка інформаційної технології віртуального ігрового автомату. Матеріали Всеукраїнської науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)» – [Електронний ресурс]. – https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/viewFile/19182/15903

2. Офіційний сайт Construct2 – [Електронний ресурс]. – Режим доступу: http://www.scirra.com.

3. Офіційний сайт easeljs – [Електронний ресурс]. – Режим доступу: http://createjs.com/easeljs.

4. Портал розробників Html5 ігор – [Електронний ресурс]. – Режим доступу https://html5gameengine.com/

5. Paul Dubois – MySql Cookbook Solutions for database developers and administrators. – O'Reilly.

6. Eloquent ORM. [Електронний ресурс]. – Режим доступу: https://eloquentbyexample.com/

7. Integrated development environment. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Інтегроване середовище розробки.

8. Laravel for php artisans. [Електронний ресурс] – Режим доступу: https://www.laravel.com.

9. Сервісно-орієнтована архітектура. [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Сервісно-орієнтована_архітектура.

10. Порівняння SOAP і REST. [Електронний ресурс] - Режим доступу: https://medium.com/@nanotexnolagiya/сравнение-soap-и-rest-с-json-2019-779fef6eba9b.

11. В.М.Антоненко, С.Д.Мамченко, Ю.В.Рогушина – Сучасні інформаційні системи і технології: управління знаннями.

12. Драчук С.В. Керування обліковими записами клієнтів в онлайн іграх / С.В. Драчук, С.І. Перевозніков // Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2020)»: Тез. доп. – Вінниця, 2020.

13. PHP, MySql та інші технології. [Електронний ресурс] – Режим доступу: http://www.php.su/.

14. Інформаційні бази [Електронний ресурс]. URL: https://its.1c/db/metod8dev/content/1591/hdoc. /(дата звернення: 2023р.).

15. Робота з базою даних [Електронний ресурс]. URL: https://v8.1c/platforma/rabota-s-bazoy-dannykh/(дата звернення: 2023р.).

16. Алгоритм і система роботи онлайн-слотів. [Електронний ресурс] – Режим доступу: https://www.softgamings.com/ru/blog/algoritm-slotov/.

17. П. Кравець, Р. Киркало – Системи прийняття рішень з нечіткою логікою.

18. Інструкція для слот автоматів, історія, типи, принципи роботи. [Електронний ресурс] – Режим доступу: http://mukachevo.net/ua/news/view/

19. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

20. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

ДОДАТКИ

Додаток А (обов'язковий)

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія віртуального ігрового автомата

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

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

Показники звіту подібності Unicheck

Оригінальність 90,32% Схожість 9,68%

Аналіз звіту подібності (відмітити потрібне):

✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

□ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

□ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

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

Автор роботи _____ Крамаренко Д.А.

Керівник роботи _____ Озеранський В.С.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку _____ Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

```
<?php
namespace Games\Starburst\Actions;
use Games\Starburst\Logic;
use Models\Starburst\SpinMapper;
use Models\Starburst\Spin as StarburstSpin;
use Models\Starburst\BetMapper;
use Models\Starburst\Bet;
use Models\Starburst\WinMapper;
use Models\Starburst\Win;
use GuzzleHttp\Client as Guzzle;
class Spin
{
    private $logic;
    private $bl_standart = '0,1,2,3,4,5,6,7,8,9';
    function __construct()
    {
        $this->logic = new Logic;
    }
    public function run($container, $query_params, $session)
    {
        if (empty($query_params['wantsreels']) || $query_params['wantsre
els'] != 'true') {
            return \GameError::send(0);
        }
        if (empty($query_params['wantsfreerounds']) || $query_params['wa
ntsfreerounds'] != 'true') {
            return \GameError::send(0);
        }
        if ( empty($query_params['bet_denomination']) || empty($query_pa
rams['bet_betlevel']) || empty($query_params['bet_betlines']) ) {
            return \GameError::send(0);
        }
        $balance = $session['relationships']['balance'];
        if ($balance['attributes']['amount'] <= 0) {
            return \GameError::send(10);
        }
        $betlevel = $container['moneyManager']-
>setBetLevel($query_params['bet_betlevel']);
        if ($betlevel === FALSE) {
            return \GameError::send(0);
        }
        $denomination = $container['moneyManager']-
>setDenomination($query_params['bet_denomination']);
        if ($denomination === FALSE) {
            return \GameError::send(0);
        }
        $bet = $container['moneyManager']-
>getBet($betlevel, $denomination);
        if ($bet['amount'] > $balance['attributes']['amount']) {
            return \GameError::send(10);
        }
        $spin_response = include($container['game_template'][$session['r
elationships']['game']['attributes']['name']]['spin']);
        $spin_response['gamesoundurl'] = $container['settings']['server_
url'].'/starburst/game';
        $spin_response['jackpotcurrency'] = $balance['attributes']['curr
ency'];
        $spin_response['jackpotcurrencyiso'] = $balance['attributes']['c
urrency'];
        $spin_response['playercurrency'] = $balance['attributes']['curre
ncy'];
```

```
                $spin_response['playercurrencyiso'] = $balance['attributes']['cu
rrency'];
                $spin_state = [
                    'uid'           => $session['relationships']['user']['id'],
                    'sessid'        => $session['attributes']['token'],
                    'betlevel'      => $betlevel,
                    'denomination'  => $denomination,
                    'freespins_left'=> 0,
                    'params'        => json_encode([]),
                    'type'          => 'spin'
                ];
                $spin_mapper = new SpinMapper($container['db']);
                $spin = $spin_mapper-
>save(StarburstSpin::fromState($spin_state));
                //TODO: maybe check $spin on is_null
                $bet_state = [
                    'sid'   => $spin->id,
                    'bet'   => $bet['amount'],
                ];
                $bet_mapper = new BetMapper($container['db']);
                $neon_bet = $bet_mapper->save(Bet::fromState($bet_state));
                //TODO: maybe check $neon_bet on is_null
                $guzzle = new Guzzle();
                $request_data = [
                    'token'     => $session['attributes']['token'],
                    'spin_id'   => $spin->id,
                    'amount'    => $bet['amount'],
                ];
                try {
                    $api_request = $guzzle-
>post("{$container['settings']['api_url']}balance/bet", [
                        'headers' => [
                            'X-AUTH'        => 'Basic c2VjcmV0OnNlY3JldA==',
                            'Cache-Control' => 'no-cache',
                            'Content-Type'  => 'application/json'
                        ],
                        'json' => $request_data
                    ]);
                    $response_data = json_decode($api_request-
>getBody(), TRUE)['data'];
                } catch (\GuzzleHttp\Exception\RequestException  $e) {
                    $container['starburst_log']->error('', [
                        'request'   => json_decode($e->getRequest()-
>getBody(), TRUE),
                        'response'  => json_decode($e->getResponse()-
>getBody(), TRUE),
                    ]);
                    return \GameError::send(0);
                }
                $new_balance = $container['moneyManager']-
>convert($response_data['attributes']['amount'], $denomination);
                /**
                 * if $limit = 0 - generate reels until spin would be lost
                 * if $limit = {number} - generate random reels, dermine if limi
t low, avarage or high
                 * if hight try to regenerate reels with lower symbol params
                 * but win cannot be larger than $limit
                 */
                $limit = $response_data['attributes']['max_next_win'];
                $lowReelsPercents = false;
                if ($limit > 0) {
                    do {
                        $reels = $this->logic->generateReels($lowReelsPercents);
                        $overlaied = $this->logic->overlayReels($reels);
                        $respin = $overlaied['respin'];
                        $matches = $this->logic->match($overlaied['reels']);
                        if (empty($matches)) {
                            $i = 0;
                            break;
```

```
                    }else{
                        $win_coins = $this->logic-
>calculateWonCoins($matches);
                        $bet_results = $container['moneyManager']-
>countWinning($betlevel, $denomination, $win_coins);
                        if ($this-
>determineWinType($bet_results['amount']) === 'low' ||
                            $this-
>determineWinType($bet_results['amount']) === 'avarage') {
                            $i = $bet_results['amount'];
                        } else {
                            $lowReelsPercents = true;
                            $i = $limit + 1;
                        }
                    }
                } while($i > $limit);
            } elseif ($limit == 0) {
                $i = 1;
                while($i != 0) {
                    $reels = $this->logic->generateReels();
                    $overlaied = $this->logic->overlayReels($reels);
                    $respin = $overlaied['respin'];
                    $matches = $this->logic->match($overlaied['reels']);
                    if (empty($matches)) {
                        $i = 0;
                    }
                }
            }
            /**
             * end limits
             */
            $positions = [];
            foreach ($overlaied['reels'] as $col => $reel) {
                /**
                 * overlay
                 */
                $position = rand(1,999);
                while (in_array($position, $positions)) {
                    $position = rand(1,999);
                }
                $spin_response['rs.i0.r.i'.$col.'.pos'] = $position;
                array_push($positions, $position);
                if (in_array('SYM1', $reel)) {
                    $i = 0;
                    $overlay_position = $spin_response['rs.i0.r.i'.$col.'.po
s'];
                    $spin_response['rs.i0.r.i'.$col.'.hold'] = 'true';
                    foreach ($reel as $row => $sym) {
                        if ($sym == 'SYM1') {
                            $spin_response['rs.i0.r.i'.$col.'.overlay.i'.$i.
'.pos'] = $overlay_position;
                            $spin_response['rs.i0.r.i'.$col.'.overlay.i'.$i.
'.row'] = $row;
                            $spin_response['rs.i0.r.i'.$col.'.overlay.i'.$i.
'.with'] = 'SYM1';
                            $overlay_position++;
                            $i++;
                        }
                    }
                }
                $spin_response['rs.i0.r.i'.$col.'.syms'] = implode(',', $ree
l);
            }
            if (empty($matches)) {
                $spin->params = json_encode([
                    'reels'         => $overlaied['reels'],
                    'combinations'  => [
                        'matches'   => $matches,
                        'info'      => []
```

```
                    ]
                ]);
                $spin->balance = $new_balance['amount'];
                $spin->status = 1;
                if ($respin) {
                    $spin->type = 'freespin';
                    $spin->freespins_left = 1;
                    $spin->status = 0;
                    $spin_response['current.rs.i0'] = 'wildOnReel_'.$overlai
ed['wilds'];
                    $spin_response['gameover'] = 'false';
                    $spin_response['next.rs'] = 'wildOnReel_'.$overlaied['wi
lds'];
                    $spin_response['nextaction'] = 'respin';
                }
                $spin_response['credit'] = $new_balance['cents'];
                $spin->response = json_encode($spin_response);
                $spin = $spin_mapper->save($spin);
                return $spin_response;
            }else{
                $win_state = [
                    'sid'       => $spin->id,
                    'win_amount'=> $bet_results['amount']
                ];
                $win_mapper = new WinMapper($container['db']);
                $win = $win_mapper->save(Win::fromState($win_state));
                if (!$respin) {
                    $request_data = [
                        'token'     => $session['attributes']['token'],
                        'spin_id'   => $spin->id,
                        'amount'    => $bet_results['amount'],
                    ];
                    try {
                        $api_request = $guzzle-
>post("{$container['settings']['api_url']}balance/win", [
                            'headers' => [
                                'X-
AUTH'        => 'Basic c2VjcmV0OnNlY3JldA==',
                                'Cache-Control' => 'no-cache',
                                'Content-Type'  => 'application/json'
                            ],
                            'json' => $request_data
                        ]);
                        $response_data = json_decode($api_request-
>getBody(), TRUE)['data'];
                        } catch (\GuzzleHttp\Exception\RequestException  $e)
 {
                            $container['starburst_log']->error('', [
                            'request'   => json_decode($e->getRequest()-
>getBody(), TRUE),
                            'response'  => json_decode($e->getResponse()-
>getBody(), TRUE),
                        ]);
                        return \GameError::send(0);
                        }
                        $new_balance = $container['moneyManager']-
>convert($response_data['attributes']['amount'], $denomination);
                    }
                // information for back office
                $info = [];
                $info['show_lines'] = TRUE;
                foreach ($matches as $line => $symbols) {
                    $info['lines'][$line+1]['sym'] = $this->logic-
>getWonSymbol($symbols);
                    $info['lines'][$line+1]['gain'] = $bet_results['symbols'
][$line]['amount'];
                    foreach ($symbols as $sym) {
                        $info['lines'][$line+1]['matches'][] = $sym;
                    }
```

```
                    }
                    //
                    $spin->balance = $new_balance['amount'];
                    $spin->params = json_encode([
                        'reels'         => $overlaied['reels'],
                        'combinations'  => [
                            'matches'   => $matches,
                            'gain'      => $bet_results['symbols'],
                            'info'      => $info
                        ]
                    ]);
                    $spin->status = 1;
                    if ($respin) {
                        $spin->status = 0;
                        $spin->type = 'freespin';
                        $spin->freespins_left = 1;

                        $spin_response['current.rs.i0'] = 'wildOnReel_'.$overlai
ed['wilds'];
                        $spin_response['gameover'] = 'false';
                        $spin_response['next.rs'] = 'wildOnReel_'.$overlaied['wi
lds'];
                        $spin_response['nextaction'] = 'respin';
                    }
                    $spin_response['credit'] = $new_balance['cents'];
                    $spin_response['totalwin.coins'] = $bet_results['coins'];
                    $spin_response['totalwin.cents'] = $bet_results['cents'];
                    $spin_response['game.win.coins'] = $bet_results['coins'];
                    $spin_response['game.win.cents'] = $bet_results['cents'];
                    $spin_response['game.win.amount'] = $bet_results['amount'];
                    $positions = $this->logic-
>defineWonPositions($matches, $bet_results['symbols']);
                    foreach ($positions as $value) {
                        $spin_response = array_merge($spin_response, $value);
                    }
                    $spin->response = json_encode($spin_response);
                    $spin = $spin_mapper->save($spin);
                    return $spin_response;
                }
            }
        }
```

Лістинг коду Respin.php

```
<?php
namespace Games\Starburst\Actions;
use \Games\Starburst\Logic;
use \Models\Starburst\Spin as NeonSpin;
use \Models\Starburst\SpinMapper;
use \Models\Starburst\Freespin;
use \Models\Starburst\FreespinMapper;
use \Models\Starburst\Win;
use \Models\Starburst\WinMapper;
use GuzzleHttp\Client as Guzzle;
class Respin
{
    private $logic;
    function __construct()
    {
        $this->logic = new Logic;
    }
    public function run($container, $query_params, $session)
    {
        if (empty($query_params['wantsreels']) || $query_params['wantsre
els'] != 'true') {
            return \GameError::send(0);
        }
        if (empty($query_params['wantsfreerounds']) || $query_params['wa
ntsfreerounds'] != 'true') {
            return \GameError::send(0);
        }
```

```
                $spin_mapper = new SpinMapper($container['db']);
                $spin = $spin_mapper-
>lastByUid($session['relationships']['user']['id']);
                if (is_null($spin)) {
                    return \GameError::send(0);
                }
                if ($spin->freespins_left <= 0) {
                    return \GameError::send(0);
                }
                $win_mapper = new WinMapper($container['db']);
                $spin_win = $win_mapper->findBySpinId($spin->id);
                $freespin_mapper = new FreespinMapper($container['db']);
                $freespin = $freespin_mapper->lastBySpinId($spin->id);
                $previous = is_null($freespin) ? $spin : $freespin;
                $previous_params = json_decode($previous->params, TRUE);
                if (empty($previous_params['reels'])) {
                    $container['starburst_log']-
>error('There is no freespin some params value', [
                        'request'   => $previous
                    ]);
                    return \GameError::send(0);
                }
                $balance = $session['relationships']['balance'];
                $freespin_response = include($container['game_template'][$sessio
n['relationships']['game']['attributes']['name']]['spin']);
                /**
                 * Request to get winnig a limit
                 */
                $guzzle = new Guzzle();
                $request_data = [
                    'token'     => $session['attributes']['token']
                ];
                try {
                    $api_request = $guzzle-
>post("{$container['settings']['api_url']}operator/limit", [
                        'headers' => [
                            'X-AUTH'        => 'Basic c2VjcmV0OnNlY3JldA==',
                            'Cache-Control' => 'no-cache',
                            'Content-Type'  => 'application/json'
                        ],
                        'json' => $request_data
                    ]);
                    $response_data = json_decode($api_request-
>getBody(), TRUE)['data'];
                    $limit = $response_data['attributes']['max_next_win'];
                } catch (\GuzzleHttp\Exception\RequestException  $e) {
                    $container['starburst_log']->error('', [
                        'request'   => json_decode($e->getRequest()-
>getBody(), TRUE),
                        'response'  => json_decode($e->getResponse()-
>getBody(), TRUE),
                    ]);
                    return \GameError::send(0);
                }
                /** end limit request */
                if ($limit > 0) {
                    do {
                        $reels = $this->logic-
>generateReels('freespin', $previous_params['reels']);
                        $overlaied = $this->logic->overlayReels($reels);
                        $respin = FALSE;
                        if ($overlaied['respin'] === TRUE && $spin-
>freespins_left < 3) {
                            $respin = TRUE;
                        }
                        $matches = $this->logic->match($overlaied['reels']);
                        if (empty($matches)) {
                            $i = 0;
                            break;
```

```
                    } else {
                        // maybe rewrite just for convert win_sum instead of
 counting bet results;
                        $won_coins = $this->logic-
>calculateWonCoins($matches);
                        $freespin_results = $container['moneyManager']-
>countWinning(
                            $spin->betlevel,
                            $spin->denomination,
                            $won_coins
                        );
                        $previous_wins = $freespin_mapper-
>sumWinAmount($spin->id);
                        if (!is_null($spin_win)) {
                            $i = $spin_win-
>win_amount + $previous_wins + $freespin_results['amount'];
                        } else {
                            $i = $previous_wins + $freespin_results['amount'
];
                        }
                        if ($this-
>determineWinType($freespin_results['amount']) !== 'high') {
                            $lowReelsPercents = true;
                            $i = $limit + 1;
                        }
                    }
                } while ($i > $limit);
            } elseif ($limit == 0) {
                $i = 1;
                while($i != 0) {
                    $reels = $this->logic-
>generateReels('freespin', $previous_params['reels']);
                    $overlaied = $this->logic->overlayReels($reels);
                    $respin = FALSE;
                    if ($overlaied['respin'] === TRUE && $spin-
>freespins_left < 3) {
                        $respin = TRUE;
                    }
                    $matches = $this->logic->match($overlaied['reels']);
                    if (empty($matches)) {
                        $i = 0;
                    }
                }
            }
            $freespin_response['clientaction'] = 'respin';
            $freespin_response['last.rs'] = 'wildOnReel_'.$this->logic-
>overlayReels($previous_params['reels'])['wilds'];
            $freespin_response['previous.rs.i0'] = $freespin_response['last.
rs'];
            $freespin_response['gamesoundurl'] = $container['settings']['ser
ver_url'].'/starburst/game';
            $freespin_response['playercurrency'] = $balance['attributes']['c
urrency'];
            $freespin_response['playercurrencyiso'] = $balance['attributes']
['currency'];
            $freespin_response['jackpotcurrency'] = $balance['attributes']['
currency'];
            $freespin_response['jackpotcurrencyiso'] = $balance['attributes'
]['currency'];
            $positions = [];
            foreach ($overlaied['reels'] as $col => $reel) {
                /**
                 * overlay
                 */
                $position = rand(1,999);
                while (in_array($position, $positions)) {
                    $position = rand(1,999);
                }
                $freespin_response['rs.i0.r.i'.$col.'.pos'] = $position;
```

```
                    array_push($positions, $position);
                    if (in_array('SYM1', $reel)) {
                        $i = 0;
                        $overlay_position = $freespin_response['rs.i0.r.i'.$col.
'.pos']
                        $freespin_response['rs.i0.r.i'.$col.'.hold'] = 'true';
                        foreach ($reel as $row => $sym) {
                            if ($sym == 'SYM1') {
                                $freespin_response['rs.i0.r.i'.$col.'.overlay.i'
.$i.'.pos'] = $overlay_position;
                                $freespin_response['rs.i0.r.i'.$col.'.overlay.i'
.$i.'.row'] = $row;
                                $freespin_response['rs.i0.r.i'.$col.'.overlay.i'
.$i.'.with'] = 'SYM1';
                                $overlay_position++;
                                $i++;
                            }
                        }
                    }
                    $freespin_response['rs.i0.r.i'.$col.'.syms'] = implode(',',
$reel);
                }
                $converted_balance = $container['moneyManager']-
>convert($balance['attributes']['amount'], $spin->denomination);
                if (empty($matches)) {
                    $freespin_state = [
                        'sid'   => $spin->id,
                        'uid'   => $session['relationships']['user']['id'],
                        'sessid'=> $session['attributes']['token'],
                        'params'=> json_encode([
                            'reels'         => $overlaied['reels'],
                            'combinations'  => [
                                'matches'   => $matches,
                                'info'      => []
                            ]
                        ]),
                    ];
                    $previous_wins = $freespin_mapper->sumWinAmount($spin->id);
                    if ($respin === FALSE) {
                        if ($previous_wins > 0 || !is_null($spin_win)) {
                            $win_state = [
                                'sid'        => $spin->id,
                                'win_amount' => !is_null($spin_win) ? ($previous
_wins + $spin_win->win_amount) : $previous_wins
                            ];
                            if (!is_null($spin_win)) {
                                $spin_win-
>win_amount = $win_state['win_amount'];
                                $win = $win_mapper->save($spin_win);
                            } else {
                                $win = $win_mapper-
>save(Win::fromState($win_state));
                            }
                        }
                        $request_data = [
                            'token'     => $session['attributes']['token'],
                            'spin_id'   => $spin->id,
                            'amount'    => !is_null($spin_win) ? ($spin_win-
>win_amount) : $previous_wins,
                        ];
                            try {
                                $api_request = $guzzle-
>post("{$container['settings']['api_url']}balance/win", [
                                'headers' => [
                                    'X-
AUTH'        => 'Basic c2VjcmV0OnNlY3JldA==',
                                    'Cache-Control' => 'no-cache',
                                    'Content-Type'  => 'application/json'
```

```
                            ],
                            'json' => $request_data
                        ]);
                        $response_data = json_decode($api_request-
>getBody(), TRUE)['data'];
                        } catch (\GuzzleHttp\Exception\RequestException  $e)
 {
                            $container['starburst_log']->error('', [
                            'request'   => json_decode($e->getRequest()-
>getBody(), TRUE),
                            'response'  => json_decode($e->getResponse()-
>getBody(), TRUE),
                        ]);
                        return \GameError::send(0);
                        }
                       $new_balance = $container['moneyManager']-
>convert($response_data['attributes']['amount'], $denomination);
                    $freespin_response['credit'] = $new_balance['cents'];
                } else {
                    $freespin_response['credit'] = $converted_balance['cents
'];
                    $freespin_response['current.rs.i0'] = 'wildOnReel_'.$ove
rlaied['wilds'];
                    $freespin_response['gameover'] = 'false';
                    $freespin_response['next.rs'] = 'wildOnReel_'.$overlaied
['wilds'];
                    $freespin_response['nextaction'] = 'respin';
                }
                $spin->freespins_left = $respin === TRUE ? $spin-
>freespins_left + 1 : $spin->freespins_left;
                $win_amount = !is_null($spin_win) ? ($previous_wins + $spin_
win->win_amount) : $previous_wins;
                if ($respin === FALSE) {
                    /**
                     * if there no more spins
                     */
                    $spin->balance = $new_balance['amount'];
                    $spin->status = 1;
                    $win_amount = !is_null($spin_win) ? $spin_win-
>win_amount : $previous_wins;
                }
                $converted_wins = $container['moneyManager']-
>convert($win_amount, $spin->denomination);
                $freespin_response['game.win.amount'] = $converted_wins['amo
unt'];
                $freespin_response['game.win.cents'] = $converted_wins['cent
s'];
                $freespin_response['game.win.coins'] = $converted_wins['coin
s'];
                $freespin_response['totalwin.cents'] = $converted_wins['cent
s'];
                $freespin_response['totalwin.coins'] = $converted_wins['coin
s'];
                $freespin_state['response'] = json_encode($freespin_response
);
                $freespin_new = $freespin_mapper-
>save(Freespin::fromState($freespin_state));
                $spin_update = $spin_mapper->save($spin);
                return $freespin_response;
            }
            /* Won freespin */
            // information for back office
            $info = [];
            $info['show_lines'] = TRUE;
            foreach ($matches as $line => $symbols) {
                $info['lines'][$line+1]['sym'] = $this->logic-
>getWonSymbol($symbols);
                $info['lines'][$line+1]['gain'] = $freespin_results['symbols
'][$line]['amount'];
```

```
                    foreach ($symbols as $sym) {
                        $info['lines'][$line+1]['matches'][] = $sym;
                    }
                }
                //
                $freespin_state = [
                    'sid'   => $spin->id,
                    'uid'   => $session['relationships']['user']['id'],
                    'sessid'=> $session['attributes']['token'],
                    'win_amount'=> $freespin_results['amount'],
                    'params'=> json_encode([
                        'reels'         => $overlaied['reels'],
                        'combinations'  => [
                            'matches'=>$matches,
                            'gain'=>$freespin_results['symbols'],
                            'info'=>$info
                        ]
                    ])
                ];
                $freespin_new = $freespin_mapper-
>save(Freespin::fromState($freespin_state));
                $previous_wins = $freespin_mapper->sumWinAmount($spin->id);
                if ($respin === FALSE) {
                    /**
                     * if there no more spins
                     */
                    if ($previous_wins > 0 || !is_null($spin_win)) {
                        $win_state = [
                            'sid'        => $spin->id,
                            'win_amount' => !is_null($spin_win) ? ($previous_win
s + $spin_win->win_amount) : $previous_wins
                        ];
                        if (!is_null($spin_win)) {
                            $spin_win->win_amount = $win_state['win_amount'];
                            $win = $win_mapper->save($spin_win);
                        } else {
                            $win = $win_mapper-
>save(Win::fromState($win_state));
                        }
                    }
                    $request_data = [
                        'token'     => $session['attributes']['token'],
                        'spin_id'   => $spin->id,
                        'amount'    => !is_null($spin_win) ? ($spin_win-
>win_amount) : $previous_wins,
                    ];
                    try {
                        $api_request = $guzzle-
>post("{$container['settings']['api_url']}balance/win", [
                            'headers' => [
                                'X-AUTH'        => 'Basic c2VjcmV0OnNlY3JldA==',
                                'Cache-Control' => 'no-cache',
                                'Content-Type'  => 'application/json'
                            ],
                            'json' => $request_data
                        ]);
                        $response_data = json_decode($api_request-
>getBody(), TRUE)['data'];
                    } catch (\GuzzleHttp\Exception\RequestException  $e) {
                        $container['starburst_log']->error('', [
                            'request'   => json_decode($e->getRequest()-
>getBody(), TRUE),
                            'response'  => json_decode($e->getResponse()-
>getBody(), TRUE),
                        ]);
                        return \GameError::send(0);
                    }
                    $new_balance = $container['moneyManager']-
>convert($response_data['attributes']['amount'], $denomination);
```

```
                $freespin_response['credit'] = $new_balance['cents'];
            } else {
                $freespin_response['credit'] = $converted_balance['cents'];
                $freespin_response['current.rs.i0'] = 'wildOnReel_'.$overlai
ed['wilds'];
                $freespin_response['gameover'] = 'false';
                $freespin_response['next.rs'] = 'wildOnReel_'.$overlaied['wi
lds'];
                $freespin_response['nextaction'] = 'respin';
            }
            $spin->freespins_left = $respin === TRUE ? $spin-
>freespins_left + 1 : $spin->freespins_left;
            $win_amount = !is_null($spin_win) ? ($previous_wins + $spin_win-
>win_amount) : $previous_wins;
            if ($respin === FALSE) {
                /**
                 * if there no more spins
                 */
                $spin->balance = $new_balance['amount'];
                $spin->status = 1;
                $win_amount = !is_null($spin_win) ? $spin_win-
>win_amount : $previous_wins;
            }
            $spin_update = $spin_mapper->save($spin);
            $positions = $this->logic-
>defineWonPositions($matches, $freespin_results['symbols'], 'wildOnReel_'.$over
laied['wilds']);
            $converted_wins = $container['moneyManager']-
>convert($win_amount, $spin->denomination);
            $freespin_response['game.win.amount'] = $converted_wins['amount'
];
            $freespin_response['game.win.cents'] = $converted_wins['cents'];
            $freespin_response['game.win.coins'] = $converted_wins['coins'];
            $freespin_response['totalwin.cents'] = $converted_wins['cents'];
            $freespin_response['totalwin.coins'] = $converted_wins['coins'];
            foreach ($positions as $value) {
                $freespin_response = array_merge($freespin_response, $value)
;
            }
            $freespin_new->response = json_encode($freespin_response);
            $freespin_update = $freespin_mapper->save($freespin_new);
            return $freespin_response;
        }
    }
```

Лістинг коду Logic.php

```
<?php
namespace Games\Starburst;
class Logic
{
    public $lines;
    private $weights;
    private $wild = 'SYM1';
    function __construct()
    {
        $this->weights = [
            'SYM3'  =>  [3 => 50, 4 => 200, 5 => 250],
            'SYM4'  =>  [3 => 25, 4 => 60, 5 => 120],
            'SYM5'  =>  [3 => 10, 4 => 25, 5 => 60],
            'SYM6'  =>  [3 => 8, 4 => 20, 5 => 50],
            'SYM7'  =>  [3 => 7, 4 => 15, 5 => 40],
            'SYM8'  =>  [3 => 5, 4 => 10, 5 => 25],
            'SYM9'  =>  [3 => 5, 4 => 10, 5 => 25],
        ];
        $this->symbols = [        // percents
            'SYM3'  =>  9,
            'SYM4'  =>  11,
            'SYM5'  =>  12,
            'SYM6'  =>  14,
            'SYM7'  =>  16,
```

```
                'SYM8'  =>  19,
                'SYM9'  =>  19,
            ];
            $this->lines = [
                0    =>  [1,1,1,1,1],
                1    =>  [0,0,0,0,0],
                2    =>  [2,2,2,2,2],
                3    =>  [0,1,2,1,0],
                4    =>  [2,1,0,1,2],
                5    =>  [0,0,1,0,0],
                6    =>  [2,2,1,2,2],
                7    =>  [1,2,2,2,1],
                8    =>  [1,0,0,0,1],
                9    =>  [1,0,1,0,1],
            ];
        }
        ### gamelogic ###

        public function generateReels($method = null, $previous_reels = [])
        {
            $reels = [];
            /* Array with symbols, considering percents for each to be gener
ated in a reel */
            $symbols = [];
            $s = 1;
            foreach ($this->symbols as $sym => $p) {
                if (count($symbols) <= 100) {
                    for ($p; $p != 0; $p--) {
                        $symbols[$s] = $sym;
                        $s++;
                    }
                }
            }
            /** end */
```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВІРТУАЛЬНОГО ІГРОВОГО АВТОМАТА

Виконав: студент 2 курсу, групи 3КН-22м
спеціальності 122 – Комп'ютерні науки
(шифр і назва напряму підготовки, спеціальності)
Крамаренко Д.А.
(прізвище та ініціали)
Керівник: к.т.н., доцент каф. КН
Озеранський В.С.
(прізвище та ініціали)
«07» 12 2023 р.

Вінниця ВНТУ – 2023 рік

Рисунок В.1 – Програми аналоги ігрових автоматів

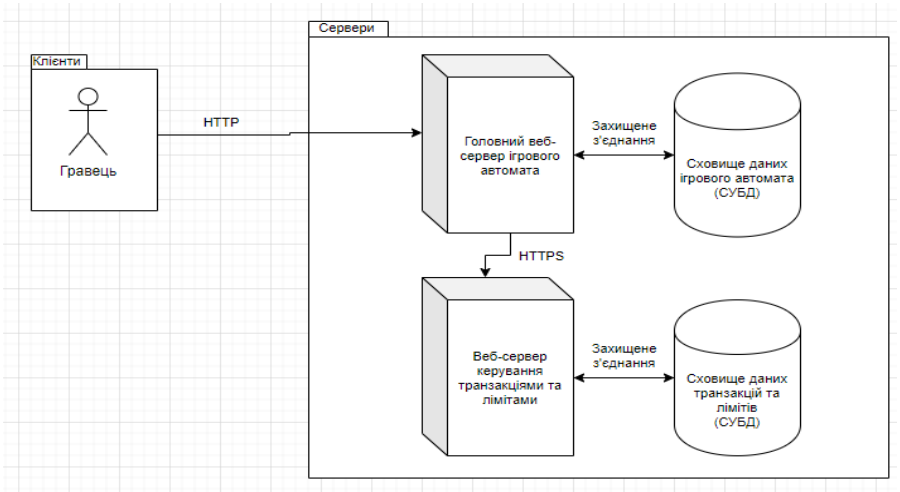

Рисунок В.2 – Структура віртуального ігрового автомата

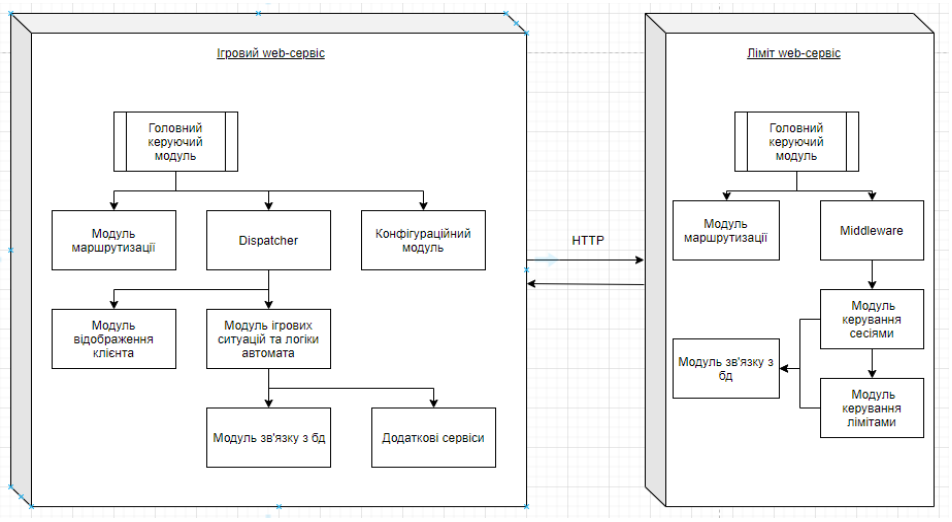

Рисунок В.3 – Модулі віртуального ігрового автомата

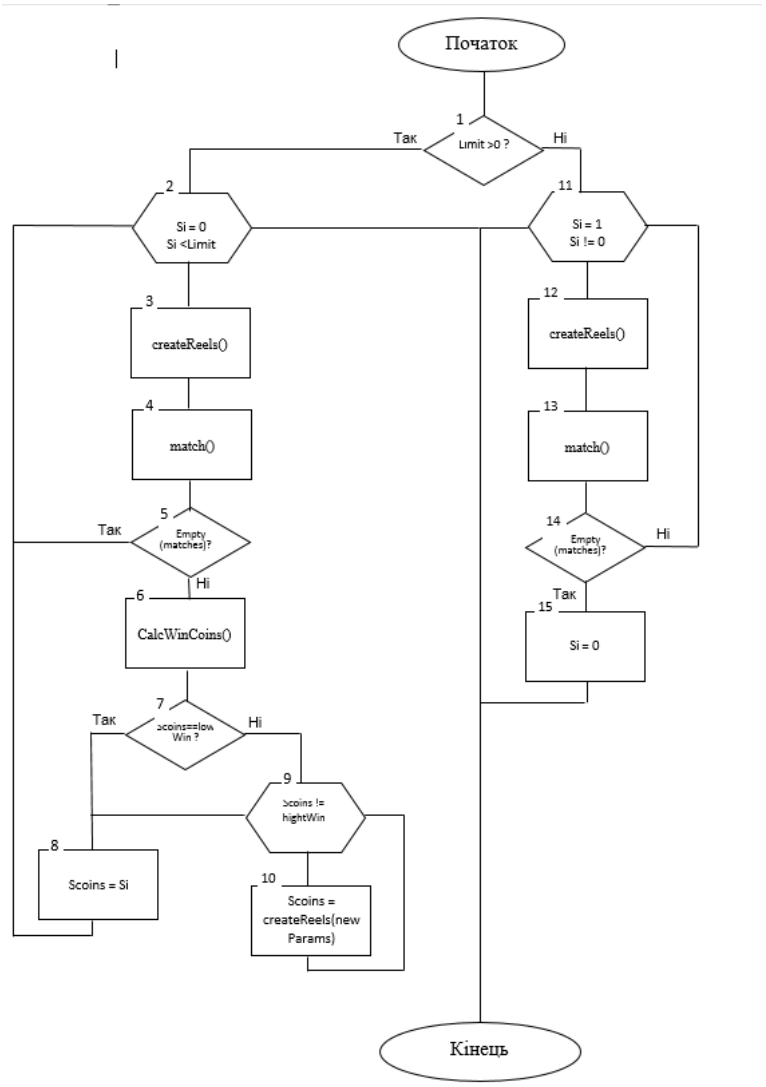

Рисунок В.4 – Алгоритм генерації виграшної комбінації віртуального ігрового автомата

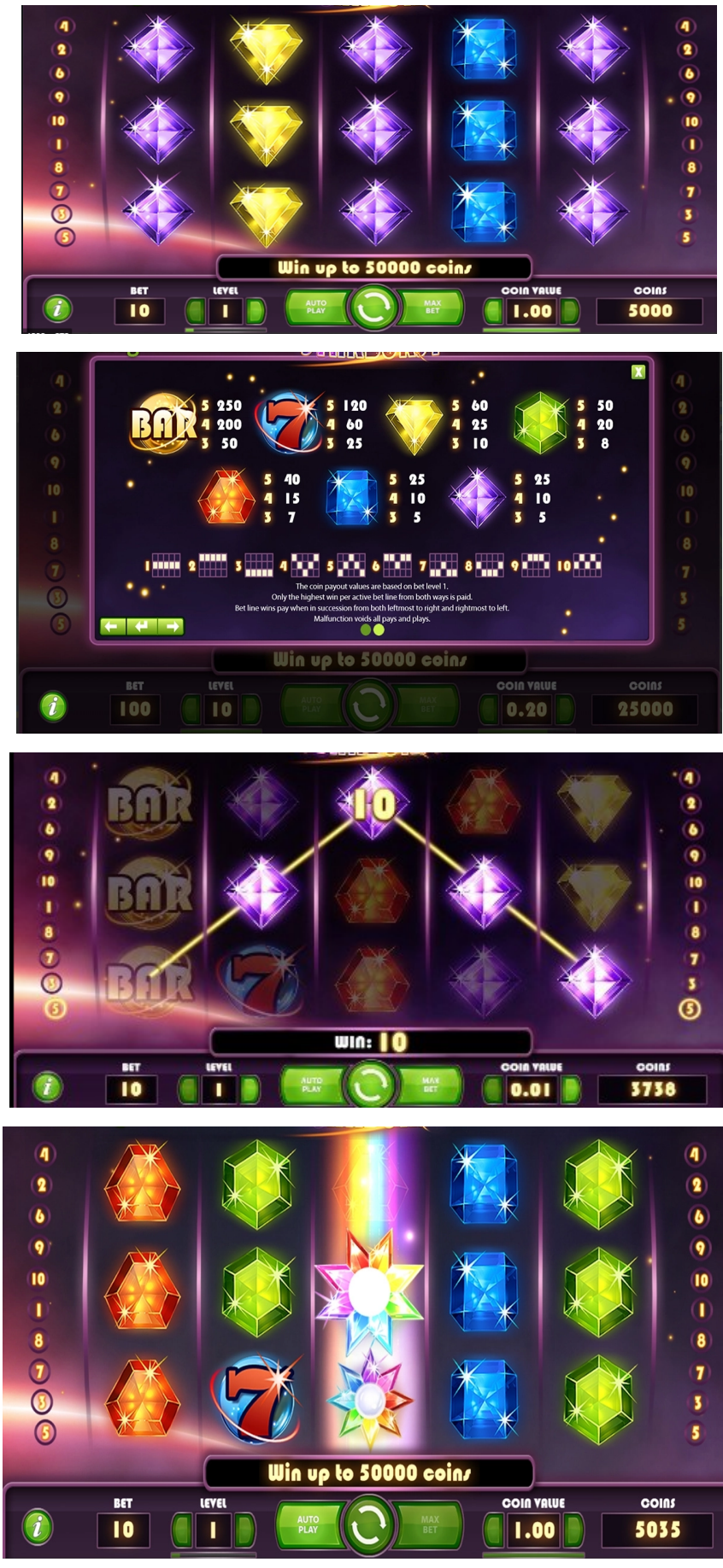

Рисунок В.5 – Інтерфейс програмного забезпечення віртуального ігрового автомата

Додаток Г (довідниковий)

Інструкція користувача

Вікно програмного забезпечення віртуального ігрового автомата складається з декількох блоків: основного ігрового поля (матриця 3x5); блоку вибору розміру ставки, визначення ваги ігрової монети та кнопки запуску барабанів ігрового поля (рис. Г.1).

Рисунок Г.1 – Вікно програмного модуля віртуального ігрового автомата

Також присутня спеціальна кнопка яка відкриває сторінку з описом різних функцій ігрового автомата та з правилами гри (рис. Г.2).

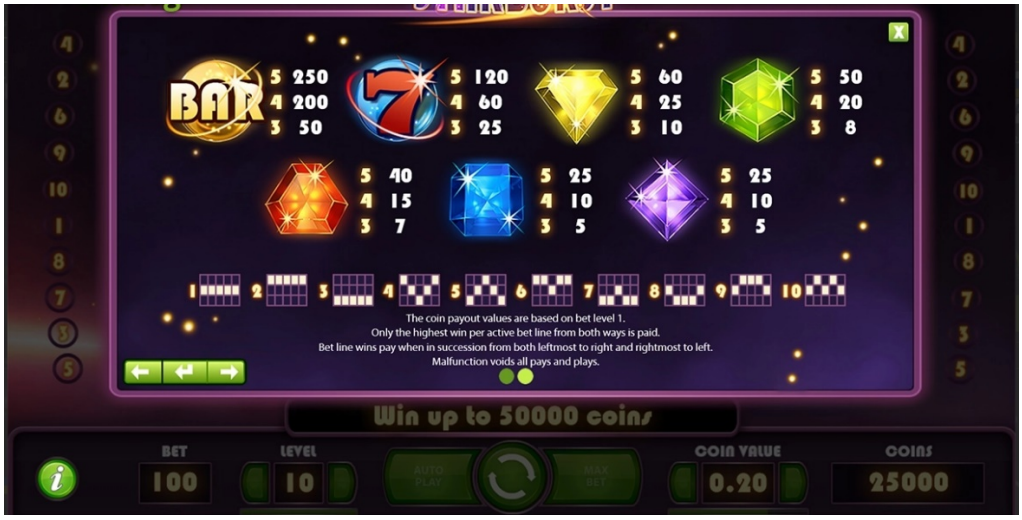

Рисунок Г.2 – Вікно з правилами віртуального ігрового автомата

Якщо відкрити вікно з правилами гри, натиснувши на відповідну кнопку, користувач побачить вікно зі списком ігрових символів по центру екрана. Виграшна комбінація кожного з них може бути з 3, 4 або 5 символів підряд. Кожна комбінація, відповідно, має свою вагу, яка вимірюється в ігрових монетах. Інформація про даний список зберігається на сервері, і при роботі алгоритму генерації виграшної комбінації, на основі цього списку формується ігрове поле. Символ має свій шанс на випадіння у відсотковому співвідношенні, так як ціна за комбінацію кожного з них, різна.

В нижній частині екрану видно список із 10 ліній по яким можуть генеруватись виграшні комбінації. Представлений ігровий автомат може співставляти комбінації як зліва на право так і справа на ліво. У разі виграшу, користувачеві буде зараховано найдорожча комбінація (рис. Г.3). Список виграшних ліній також зберігається на сервері.

Рисунок Г.3 – Приклад виграшної комбінації

Також при генерації ігрового поля з невеликим шансом може згенеруватись спеціальний символ – wild, який при порівнянні символів на виграшній лінії заміняє будь який з них. За правилами, wild може генеруватись лише на 2, 3 та 4 барабанах (рис. Г.4).

Рисунок Г.4 – Вікно віртуального ігрового автомата зі спеціальним символом wild

Якщо на всьому барабані згенерувались лише wild символи, користувач отримує додаткову бонусну прокрутку барабанів. Для бонусного раунда діють всі основні правила, лише попередньо згенерований барабан (який складається з бонусних символів), залишається на місці. Максимальна кількість бонусних раундів – 3.

Додаток Д (довідниковий)

Довідка про впровадження

МАЛЕ НАУКОВО-ВИРОБНИЧЕ ПІДПРИЄМСТВО "ТОВ "ІТІ"
Україна, 21021, м.Вінниця, вул. Келецька, 56

№ ___ від "__" _____ 20__ р.

ДОВІДКА

Дана Крамаренко Дмитру Антоновичу в тому, що результати, одержані ним в процесі виконання магістерської кваліфікаційної роботи, а саме алгоритми та програмні засоби управління ігровим віртуальним автоматом, планується використати в розробках ТОВ «ІТІ».

Зам. директора ТОВ «ІТІ» Бодяк В.М.