ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка методики підтримки ігрового процесу блокчейн ігор на основі обробки зображень»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
(код, найменування спеціальності)
освітньо-професійної програми «Інженерія програмного забезпечення»
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Микита КОНОВАЛ
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-64
Микита КОНОВАЛ

Керівник: Оксана ЗОЛОТУХНІА
к.т.н., доцент

Рецензент: _____
Ім'я, ПРІЗВИЩЕ

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти Магістр
Спеціальність 121 Інженерія програмного забезпечення
Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного забезпечення
_____ Ірина ЗАМРІЙ
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Коновалу Микиті Михайловичу

1. Тема кваліфікаційної роботи: «Розробка методики підтримки ігрового процесу блокчейн ігор на основі обробки зображень»

керівник кваліфікаційної роботи Оксана ЗОЛОТУХІНА к.т.н., доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» жовтня 2023 р. №145.

2. Строк подання кваліфікаційної роботи «29» грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, документація бібліотеки OpenCv, комп'ютерна гра MIR M: Vanguard & Vagabond, інформація про курси криптовалют.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз науково-технічної літератури про блокчейн та методи обробки зображення.

2. Дослідження методів обробки зображення відповідності шаблонів та гістограми орієнтованих градієнтів.
3. Розробка критеріїв ефективності методики.
4. Розробка математичної моделі методу підтримки ігрового процесу.
5. Розробка програмного додатку підтримки ігрового процесу на основі методів обробки зображення.
6. Аналіз отриманих результатів на основі критеріїв ефективності.

5. Перелік графічного матеріалу: *презентація*
1. Методи обробки зображення для підтримки ігрового процесу.
2. Види зображень для обробки.
3. Блок-схема циклу видобутку ігрового ресурсу.
4. Математична модель методу обробки зображень.
5. Аналіз результатів впровадження методики.

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз методів та алгоритмів блокчейн-технологій	19.10-05.11.23	
2	Особливості використання блокчейн-технологій в комп'ютерних іграх	06.11-12.11.23	
3	Аналіз особливостей підтримки ігрового процесу в блокчейн іграх	13.11-19.11.23	
4	Аналіз технологій обробки зображень	20.11-26.11.23	
5	Визначення критеріїв ефективності	27.11-01.12.23	
6	Розробка моделі підтримки ігрового процесу	02.12-04.12.23	
8	Дослідження ефективності методики	09.12-15.12.23	
9	Оформлення роботи: вступ, висновки, реферат	16.12-20.12.23	
10	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти _____ Микита КОНОВАЛ
(підпис)

Керівник
кваліфікаційної роботи _____ Оксана ЗОЛОТУХІНА
(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 67 стор., 10 табл., 21 рис., 26 джерел.

Мета роботи – підвищення ефективності видобутку криптовалюти в комп'ютерній блокчейн-грі на основі методів обробки зображень.

Об'єкт дослідження – підтримка ігрового процесу блокчейн ігор.

Предмет дослідження – методи та засоби підтримки ігрового процесу блокчейн ігор на основі обробки зображень.

Короткий зміст роботи: У роботі проведено аналіз науково-технічною літератури за темою блокчейн та його алгоритмів PoW та PoS. Проаналізовано методи обробки зображення бібліотеку OpenCv та HOG. Був проведений порівняльний аналіз методів обробки зображення в результаті, було обрано оптимальний метод обробки зображення, який відноситься до бібліотеки OpenCv та має назву "Відповідність шаблонів", функції обраного методу були використані в реалізації методики. Розроблено методику підтримки ігрового процесу блокчейн ігор на основі методу обробки зображення, яка автоматизовано видобуває криптовалюту для користувача методики. Було проведено аналіз ефективності розроблювальної методики за встановленими критеріями: час, прибуток, оптимізація ресурсів. Результати впровадження підтверджують переваги методики над ручним видобутком криптовалюти.

КЛЮЧОВІ СЛОВА: PoW, PoS, ПІДТРИМКА ІГРОВОГО ПРОЦЕСУ, OpenCv, HOG, БЛОКЧЕЙН.

ABSTRACT

Text part of the master's qualification work: 67 pages, 21 pictures, 10 table, 26 sources.

The thesis used modern technologies that are actively developing, namely blockchain technologies, image processing methods. Blockchain technology is a technology that is actively used in the world, thanks to blockchain, PoW and PoS algorithms were created. Cryptocurrencies, smart contracts and blockchain games were created on its basis. Blockchain games are a new stage in the development of blockchain technologies, as they do not require computer resources and have a simplified cryptocurrency mining process. Image processing is a field that is used all over the world to improve people's lives, as evidenced by facial recognition technologies, action cameras, and more. Therefore, based on these data, the work is considered relevant and modern.

The initial chapter of the graduate work conducts an exhaustive analysis of scientific and technical literature integral to the development of the methodology. This encompasses a comprehensive elucidation of blockchain technology delineating the functionality of cryptocurrency. The primary focus of this graduate work revolves around blockchain gaming, with the development of a gaming process support methodology based on image processing methods. Specifically, the methods elucidates the utilization of image processing techniques such as match tamplate and histogram of oriented gradients (HOG).

The subsequent section details an algorithm founded on image processing, serving as the backbone for gameplay support. This algorithm showcases all intricacies of cryptocurrency mining processes within the developed application. Furthermore, mathematical models underpinning image processing, explicating the mechanics of the pattern matching method, are expounded upon.

The third section describes the graphical interface. Additionally, it includes the presentation of two diagrams: a case diagram outlining user-system interactions and a class diagram providing a formal depiction of the methodology's processes in terms of classes and methods.

The purpose of the work – improving the efficiency of cryptocurrency mining in a computer blockchain game based on image processing methods.

Object of research – support for the gameplay of blockchain games.

Subject of research – methods and means of supporting the gameplay of blockchain games based on image processing.

The practical significance of the results lies in the use of the developed methodology for the automated mining of the game resource, which is subsequently exchanged for NFT and cryptocurrency.

KEYWORDS: PoW, PoS, GAMING SUPPORT, OpenCv, HOG, BLOCKCHAIN.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

UTXO	-	Unspend Transaction Output
NFT	-	Non-Fungible Token
PoS	-	Proof-Of-Stake
PoW	-	Proof-Of-Work
HOG	-	Histogram Of Oriented Gradients

ВСТУП

Блокчейн технології – це нова галузь, яка активно розвивається і набуває високої популярності про що, свідчить висока ціна криптовалюти за даними МІНФІНУ. Через високу популярність криптовалюти на основі блокчейн технології утворюються новітні предметні галузі, такі як блокчейн ігри або смарт-контракти. Блокчейн ігри галузь, яка поєднала в собі галузі комп'ютерних ігор та крипто-видобутку. Ця галузь набула високої популярності, через простоту видобутку криптовалюти порівнюючи з першопочатковим методом видобутку криптовалюти. Блокчейн ігри популярні в Китаї про що, свідчить реліз багатьох представників цієї галузі: MIR4, Ni no Kuni:Cross Worlds, MIR M: Vanguard and Vagabond тощо. Остання гра цього списку була обрана в якості предметної області до якої буде впроваджено методику підтримки ігрового процесу.

Обробка зображення – це перспективна технологія, яка використовується в багатьох галузях, а саме: медицина, система безпеки, комп'ютерні ігри тощо. Кожна з цих галузей створена для спрощення життя людини наведу приклад такого спрощення на основі гри MIR M: Vanguard and Vagabond. Завдяки метод обробки зображення буде автоматизовано видобуток криптовалюти, яку в подальшому можна буде обміняти на реальні гроші.

Таким чином завдання на розробку методу підтримки ігрового процесу на основі обробки зображень є сучасним та актуальним.

Мета роботи – підвищення ефективності видобутку криптовалюти в комп'ютерній блокчейн-грі на основі методів обробки зображень.

Об'єкт дослідження – підтримка ігрового процесу блокчейн ігор.

Предмет дослідження – методи та засоби підтримки ігрового процесу блокчейн ігор на основі обробки зображень.

Практична значущість результатів полягає в використанні розробленої методики для автоматизованого видобутку ігрового ресурсу, який в подальшому

обмінюється на NFT та криптовалюту з можливістю застосовувати розроблену методику на інші блокчейн ігри подібного жанру.

Для досягнення мети вирішувались наступні завдання.

1. Аналіз науково-технічної літератури про блокчейн та методи обробки зображення.
2. Дослідження методів обробки зображення відповідності шаблонів та гістограми орієнтованих градієнтів.
3. Розробка критеріїв ефективності методики.
4. Розробка математичної та формальної моделі методу підтримки ігрового процесу.
5. Розробка програмного додатку підтримки ігрового процесу на основі методів обробки зображення.
6. Аналіз отриманих результатів на основі критеріїв ефективності.

1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ПІДТРИМКИ ІГРОВОГО ПРОЦЕСУ БЛОКЧЕЙН ІГОР

1.1 Аналіз методів та алгоритмів блокчейн-технологій

1.1.1 Загальний опис блокчейн технології

В кваліфікаційній роботі використовується дві основні технології: блокчейн та обробка зображень. В даному розділі буде розглянуто технологію блокчейн. Блокчейн[1] – це ланцюгова система блоків (див.рис.1.1) в яких зберігається інформація попередніх блоків, ланцюг постійно зростає при додані нових блоків. Блок в блокчейні – цей термін позначає сукупність комп'ютерних файлів, в яких зберігаються дані транзакцій. Для прикладу: кожен блок ланцюга криптовалюти Bitcoin містить власний унікальний хеш, хеш попереднього блоку та інформацію про блок.

- унікальний хеш блоку – хеш ідентифікує блок та весь його вміст він завжди унікальний;
- хеш попереднього блоку – в кожному блоці крім першого блоку genesis, записуються дані про хеш попереднього блоку дана система робить блокчейн безпечним;
- інформація про блок – це блок в якому зберігається інформація про відправника, одержувача та кількість відправленої валюти;
- genesis блок або перший блок – є першим блоком який був записаний в будь-якій блокчейн-мережі, інакше може бути також названий, як "блок 0" або "блок 1";

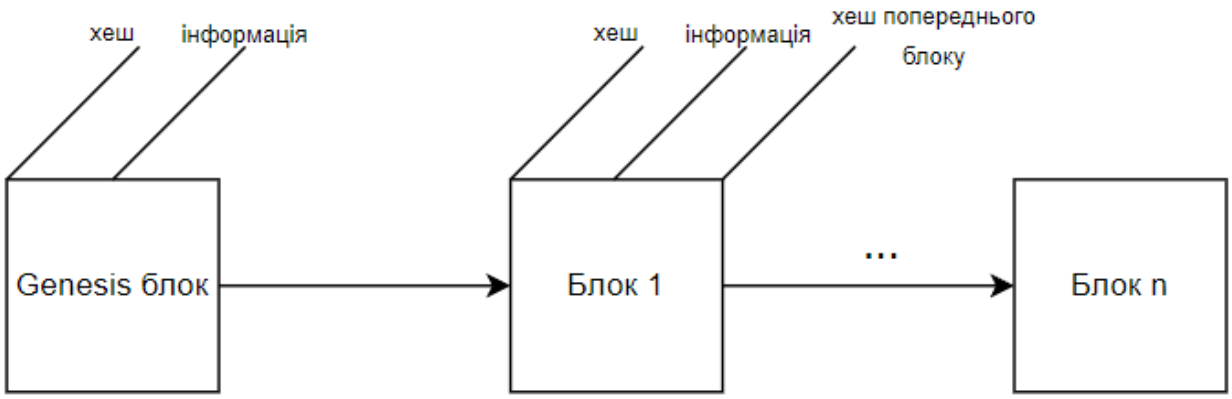


Рис. 1.1 Ланцюгова система блоків

Дані в системі зберігаються відкрито, але підробити блок з інформацією досить важко тому що, для підробки або редагування, треба змінити хеш кожного попереднього блоку перед блоком в якому проводяться зміни. Тобто, якщо змінити інформацію в блоці, то блоку надається інший хеш, який не буде збігатись з іншим блоком. Наразі комп'ютери мають достатню потужність, щоб підробити хеш у всіх попередніх блоків для подальшої валідації блокчейну. Тому для запобігання підробки окрім унікального хешу кожного блоку, блокчейн Bitcoin має механізм proof of work(PoW). Механізм PoW через свою повільну працю надає безпеку блокчейну тому що, у зловмисників немає можливості швидко змінювати хеш блоку, що призводить до не цілеспрямованих втрат ресурсів.

Блокчейн технологія набула великої популярності про що свідчить велика кількість криптовалют та ціна за одиницю криптовалюти, нижче наведена таблиця, яка відображає коштовність криптовалют на поточний час.

Таблиця 1.1.

Коштовні криптовалюти на момент за даними Мінфіну[5, 22] на 01.12.2023

Криптовалюта	**Вартість за одиницю**
Bitcoin / BTC	37,325,00 $
Ethereum / ETH	2,040,10 $
HEX / HEX	0,997698 $
Tehther / USDT	1,001 $
DRONE / DRN	0.6591 $

1.1.2 Алгоритм консенсусу Proof-of-Work

Для роботи будь-якого блокчейну необхідно використання консенсусу для того щоб, можна було додавати нові блоки до блокчейну, що в свою чергу надає прибуток користувачу та був захист хеш-рейту. Proof of work(PoW)[2] – це алгоритм досягнення консенсусу у блокчейні. Головними його учасниками є користувачі, які добивають криптовалюту, тобто майнери. За допомогою ресурсів комп’ютерного заліза вони обробляють транзакції користувачів та додають їх у блоки. Останні потім включаються до блокчейну. PoW вважається найпершим алгоритмом на ньому працюють всі старі криптовалюти такі як: Bitcoin, Ethereum, Dogecoin, Litecoin та ін. Отже, PoW працює таким чином[2]:

- користувачі (майнери) вирішують складні обчислювальні завдання використовуючи комп’ютерне обладнання, щоб додати блок транзакцій до блокчейну;
- валідатори блокчейну розподіляють винагороду за участь у стейкінгу та контролюють цілісність блоків;
- валідатори блокчейну беруть дані із заголовка блоку як вхідні дані;
- наступним кроком валідатори блокчейну безперервно запускають введення даних через криптографічний хеш-функцію. валідатори забезпечують хешування незначних змін у вхідних даних за рахунок включення довільного числа;

Алгоритм PoW, який використовується в блокчейні, вимагає вищої обчислювальної потужності для визначення додавання даних у наступному блоці. Тому, для здобичі криптовалюти на алгоритмі PoW потрібні потужні ресурси комп’ютеру, такі комп’ютери дуже прив’язані до курсу криптовалюти для якої вони збираються, що може бути досить коштовно.

Отже, механізм PoW найперший в блокчейні, тому його методи можуть бути вже застарілими, але на момент 2009 року це було передовим рішенням. Алгоритм має свої переваги та недоліки[2]. Переваги:

- високий рівень безпеки транзакцій;
- доступність для кожного користувача, який підключиться до мережі блокчейну;

недоліки механізму:

- повільна швидкість обробки транзакцій. необхідність щоразу підбирати хеш блоку зменшує швидкість роботи блокчейна, але якщо подивитись з точки зору захисту системи, то цей недолік захищає блоки від підробки;
- високе споживання електроенергії ;

1.1.3 Алгоритм консенсусу Proof of Stake

Через високу не економічність використання механізму PoW був розроблений новий алгоритм консенсусу PoS(Proof of Stake або Підтвердження частки володіння), який виконує аналогічні функції, але повинен бути економічним до використання ресурсів. Proof-of-Stake – механізм консенсусу, за допомогою якого право на генерацію нових блоків, перевірку транзакцій і включення їх у блокчейн за певним алгоритмом розміщується між обчислювальними вузлами (пристроями користувачів або нодами). Відповідно до значної концепції Proof-of-Stake, право на управління блокчейном надається всім його користувачам відповідно до долі монет, якими вони володіють на поточний час. На поточний момент всі ново-створенні криптовалюти встановлюють на алгоритмі PoS тому що, даний механізм не передбачає купівлю коштовного комп'ютерного обладнання, накопичування криптовалюти здійснюється завдяки "стейкінгу". Стейкінг(частка) потрібен для блокування певної кількості криптовалюти у крипто-гаманці для отримання права безпосередньо або через посередників брати участь у підтримці працездатності блокчейна даного блоку та отримувати за це криптовалюту. Популярними блокчейнами, які використовують алгоритм консенсусу PoS являються Ethereum, Cardano, Solana, Tezos та Algorand.

Також, цей алгоритм використовує предметна область кваліфікаційної роботи, тобто комп'ютерна гра MIR M: Vanguard & Vagabond, яка використовує блокчейн технології, що надає унікальності ігровому процесу, блокчейн в свою чергу будується на механізмі консенсусу PoS, криптовалютою являються монети DRONE, DOGMA.

Отже, порівнюючи два механізми консенсусу PoS та PoW. Два механізми мають однакову ціль, тобто надання консенсусу блокчейну, але досягнення цих цілей мають значні відмінності. В механізмі PoW для досягнення консенсусу використовується метод майнингу де майнери(користувачі) добувають криптовалюту виконуючи складні математичні задачі. В свою чергу PoS валідатори отримують право на додавання блоку транзакцій і винагороду через стейкінг. Тобто вони зобов'язані

заблокувати в мережі певну кількість криптовалюти, щоб мати шанс бути обраним для оплачуваної роботи в блокчейні. Наразі програмісти прагнуть використовувати PoS механізм через його екологічність перед попередником.

1.2 Особливості використання блокчейн-технологій в комп'ютерних іграх

1.2.1 Загальний опис блокчейн в комп'ютерних іграх

Перші спроби гейміфікації блокчейн технологій були зроблені у 2018 році на криптовалюті Ethereum та мала назву "CryptoKitties". Наразі ця галузь ігрової індустрії дуже швидко розвивається та стала найбільш популярною у Китаю. Про це свідчить реліз багатьох представників цієї галузі, прикладом може стати Ni no Kuni: Cross Worlds, MIR 4, MIR M: Vanguard & Vagabond тощо. За форматом та жанром блокчейн-ігри не відрізняються від класичних ігор. Але різниця в тому що, у класичній грі користувач немає прав власності на ігрові об'єкти, всі права залишаються в розробників. У іграх з технологією блокчейн користувач реально володіє ігровими об'єктами, які були придбані, тому що користувачем придбається NFT. Гра на основі блокчейну здатна запропонувати справжнє право власності на цифровий ігровий актив, оскільки посилання на власника активу можна записати в незмінному блокчейні з точки зору смарт-контрактів. Смарт-контракти – це програми з відкритим кодом, написані на платформі Ethereumblockchain. Розумні контракти постійно розраховуються, оскільки транзакції цифрових ігрових активів перевіряються залученими гравцями. Смарт-контракти на блокчейні також можна використовувати для безпечної миттєвої передачі права власності на цифрові ігрові активи без залучення третіх сторін. Головною особливістю індустрії можна зазначити високий поріг входу, тобто в цю галузь може поглинутись користувач, який грав у класичні

комп’ютерні ігри, у випадку даного проекту це – комп’ютерні ігри з ізометричною перспективою.

Також, треба зазначити недоліки та переваги блокчейн-ігор. Головним недоліком являється нестабільність криптовалюти в таких проектах, тобто повне знецінення криптовалюти конкретного проекту з подальшою втратою ресурсів користувача: час, токени тощо. Але в блокчейн-ігри грають через їх переваги для користувача:

- децентралізованість індустрії – контроль над заробленими активами та можливість впливу на гру завдяки dao. dao – це децентралізована автономна організація, яка повністю або частково керує програма, голосування та фінанси обробляються через блокчейн;
- доступність індустрії для користувачів – блокчейн-ігри об’єднують в собі дві функції розваги та заробіток, що притягує багато нових користувачів;

Отже, для блокчейн-ігор добре продуманий ігровий дизайн має бути фундаментальною основою для тривалого утримання гравців.

1.2.2 Блокчейн технологія в комп’ютерній грі MIR M: Vanguard & Vagabond

MIR M: Vanguard & Vagabond – це масивна багатокористувацька онлайн-гра з ізометричною перспективою, що базується на жанрі MMORPG (масова онлайн-рольова гра). Гра пропонує гравцям величезний відкритий світ, де вони можуть взаємодіяти, розвивати ігрових персонажів та брати участь у багатьох видах внутрішніх подій. MIR M: VANGUARD & VAGABOND також має складну економічну систему та соціальні аспекти, що побудовані на технології блокчейн.

Економіка в комп'ютерній грі MIR M: Vanguard & Vagabond структурована таким чином: EXW та NFT.

NFT – це незамінні токени створених в середині гри “персонажів”, які можна обмінювати, купувати та продавати за допомогою WEMIX гаманця. WEMIX гаманець – це цифровий актив, який прив’язаний до USDC у співвідношенні один до одного.
В грі представлені такі токени “DRONE” та “DOGMA”:

DRONE(Darksteel Reserve Onward New Economy) – це токен, який являється частиною DAO. Токен можна отримати, обмінюючи видобутий ігровий ресурс темна сталь (darksteels) на спеціальну скриньку, яка потребує 110,000 одиниць ресурсу, скринька обмінюється на DRONE у співвідношенні 1:1.DOGMA – це маркер управління який визначає доступність основного вмісту в MIR M. Токен можна отримати, як винагороду поставивши блок DRONE або HYDRA у програмі стакінгу DIVINE, токени отримують у співвідношенні 1:1, тобто один блок один токен.

WEMIX гаманець – це стабільна монета, прив’язана до USD (американского долару) у співвідношенні 1:1.

Hydra[5] – це блокчейн із відкритим вихідним кодом PoS, який працює на моделі UTXO. Система спрямована на реалізацію критичних економічних функцій і використання перевірених технологій для передачі даних. Згідно з офіційним документом, блокчейн прагне фіксувати комісію за транзакції за допомогою унікального механізму спалювання створеної транзакційної економіки.

UTXO (Unspend Transaction Output)(див.рис.1.2.)[6] – це кількість криптовалюти, яка залишилась після виконання транзакції. Коли транзакція завершена, невитрачена вихідна транзакція повертається в базу даних як вхідні дані, які в подальшому можна буде використовувати в інших транзакціях.

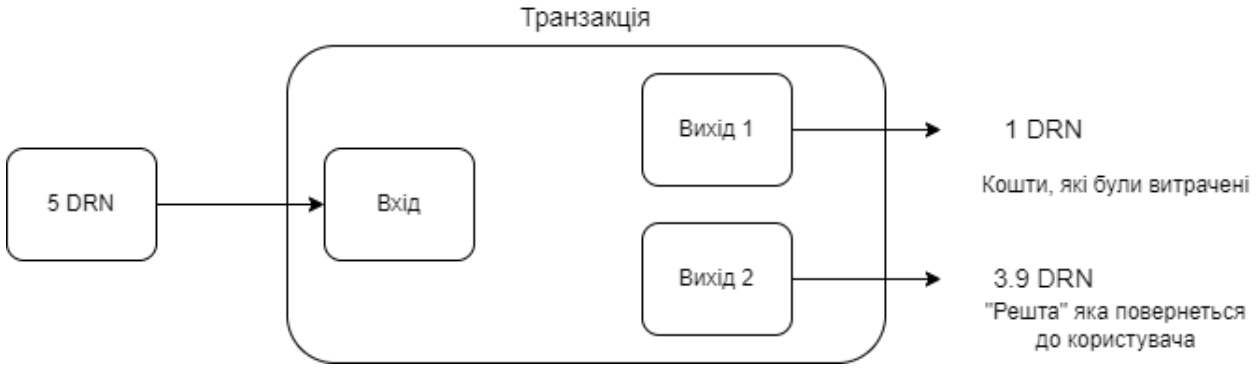


Рис. 1.2 Спрощена модель праці UTXO

Решту криптовалюти, яка залишилось можна використати таким чином:

- Надіслати решту криптовалюти на свій рахунок.
- Використати решту як комісію за транзакцію. Для того, щоб інші користувачі додали транзакцію до блокування і вона була підтвердженою.

EXW аналог NFT з відзнакою в тому, що обмін або продаж здійснюється ігрових предметів, цей спосіб заробітку токенів для розроблювальної методики не підходить за такими критеріями:

- Потрібне постійне втручання користувача.
- Нестабільність заробітку, предмети мають зазначений грою шанс на випадіння.

1.3 Аналіз технологій обробки зображень

1.3.1 Бібліотека OpenCv

OpenCv(Open Source Computer Vision) – це бібліотека з відкритим кодом та ліцензією BSD (має альтернативну назву “copycenter”, що означає можливість використання бібліотеки в будь-яких цілях), яка включає в себе багато різних

алгоритмі комп'ютерного зору. Перша версія бібліотеки розроблювалась для мови програмування C ,але з релізом другої версії бібліотека перейшла на C++, також підтримуються підключення до інших мов програмування Python, Java, Ruby тощо. Бібліотека OpenCv має модульну структуру, який включає в себе динамічні та статичні модулі.

core – модуль, що визначає базові структури даних, включаючи щільні багатовимірні матриці.

imgproc – модуль обробки зображень, що включає лінійне та нелінійне фільтрування зображень, геометричні трансформації зображень.

- фільтрація зображень – додаткові методи фільтрації зображень, а саме: медіанний фільтр, фільтр гауса. фільтри допомагають виправляти дефекти на зображенні;
- виявлення та видалення контурів – функція пошуку границь та видалення дефективних деталей на зображенні;

video – модуль аналізу відео, що включає видалення фону та алгоритми відстеження об'єктів.

Ключові функції модулю:

- обробка кадрів – функція модулю за допомогою, якої проводяться такі операції над кадрами в відео: зміна розміру, обрізка, фільтрація, зміна кольорів, виявлення об'єктів, відстеження руху, видалення шуму;
- кодеки та формати – бібліотека підтримує різні формати файлів, що дозволяє працювати з різними файлами;

calib3d – модуль, який включає в себе калібрування камери, ректифікації зображень, загалом, використовується для роботи з тривимірними даними. Ключові функції модулю:

- калібрування камери – використовується для визначення параметрів камери, а саме: матриця, коефіцієнти об’єктива тощо. важлива функція для виправлення дефектів при зйомці з камери;
- ректифікація зображень – вирівнює два зображення, щоб вони були однаковими по вертикалі та горизонталі;
- вирішення завдань стереозору – функція, яка шукає відповідності між двома зображеннями, розраховує глибину стерео зображення;
- тривимірна реконструкція – використовується для перетворення отриманого зображення з камер відеоспостереження до моделі тривимірних координат;

features2d – надає функціонал для виявлення особливостей ключових точок на зображеннях, описування знайдених точко та пошук відповідності між ними.

Ключові функції модулю:

- детектори ключових точок – знаходить “ключові точки” для подальшого аналізу, наприклад виявлення кутів об’єкта або текстурні зміни на зображенні;
- опис ключових точок – функція, яка працює з даними, які були про аналізовані попереднім методом. описи надають інформацію про границі зображення;
- методи пошуку відповідності – функція, пошуку відповідності на зображенні використовуючи інформацію їх опису;

objdetect – містить інструменти для роботи з виявленням об’єктів та екземплярів заданих класів, наприклад: людина, дерево, автомобіль тощо.

ключові функції модулю:

- hog(histogram of oriented gradients) – функція аналізу градієнти яскравості зображення, наприклад: пошук конкретних об’єктів на зображенні людей, тварин тощо;
- обробка відео та прямих трансляцій – функція обробки зображення у реальному часі;

highgui – це набір інструментів для створення віконного інтерфейсу, відображення зображень та відео, через взаємодію користувача через обладнання введення та виведення зображень.

- робота з веб-камерою – дозволяє використовувати відео дані з веб-камери для подальшої обробки зображення;

gpu – дозволяє використовувати потужності відеокарти комп’ютеру для прискорення обчислень обробки зображень. Використовується для прискорення процесів обробки зображень за методами перелічених вище.

Для того, щоб не втрачати якість інформації або пошук на зображеннях був детальним, тобто меншою кількістю похибок в обробці зображення. Якщо зберігається менші 8 (16) біт результату, це призведе до візуальних артефактів, та може впливати на подальший аналіз зображення. Тому, бібліотекою використовується метод “Saturated arithmetic”(арифметика насиченості) цей метод обробки значень, обмежує зображення за певними типами даних, 8 або 16-канальними. Наприклад, щоб зберегти результат операції в 8-бітному зображенні, ви обираєте найближче значення з діапазону 0..255.

Таблиця 1.2.

Обмеження по типам даних з якими працює бібліотека

Тип даних	**Опис**	**Синтаксис в мові програмуванні**
8-bit	Беззнакові цілі	uchar
8-bit	Зі знаком цілі	schar

Продовження Таблиці 1.2

Обмеження по типам даних з якими працює бібліотека

Тип даних	**Опис**	**Синтаксис в мові програмуванні**
16-bit	Беззнакові цілі	ushort
16-bit	Зі знаком цілі	short
32-bit	Зі знаком цілі	int
32-bit	З плаваючою крапкою	float
64-bit	З плаваючою крапкою	double

Отже, було зазначено багатофункціональність бібліотеки OpenCv, яка дозволяє великою кількістю способів обробляти зображення. Для дипломної роботи був обраний метод відповідність шаблонів(Match tamplate).

1.3.2 Відповідність шаблонів

Метод відповідності шаблонів – це техніка обробки зображень, яка використовується для пошуку збіжності по шаблону. В даній роботі доречно використовувати цей метод, через статичне зображення, яке буде в якості вихідного зображення. Елемент, який порівнюється може мати не великі розміри, тому в методі використовується “маска”. Маска – виділяє об'єкт, який треба знайти методу для більшої точності. Метод працює з темними та білими кольорами, що надає змогу працювати з зображеннями низької якості та отримувати точний результат збігу вихідного зображення та шаблону.

Метод працює таким чином, потрібні два основним компоненти для обробки: вихідне зображення та зображення шаблону.

Вихідне зображення – це зображення, в якому ми хочемо знайти збіг із зображенням шаблону, цей скріншот робить програма завдяки функції мови

програмування Python MSS, яка в свою чергу використовує стандартний функціонал операційної системи Windows 10. Python MSS(Multiple Screen Shots) – це бібліотека, яка використовується для отримування швидких знімків екрану. Для розробки методики використовується, для надання вихідних скріншотів за якими відбувається пошук потрібної піктограми або об’єкту.

Зображення шаблону – зображення, яке порівнюватиметься з вихідним скріншотом, цей шаблон завантажується в директорію, яку власноручно робить розробник в залежності від цілей.

Пошук потрібного елементу матриці(M1 – вихідне зображення) виконується завдяки переміщенням пікселів по горизонталі та вертикалі, починаючи з верхнього лівого кутка матриці та до моменту поки не буде знайдено елемент співпадіння. Завдяки функції TM_CCORR_NORMED утворюється третя матриця(M3) зображення, яке представлене у негативі, тобто темні елементи зображення вважаються -1, а яскраві елементи зображення +1. Це дозволяє більш робити більш точну обробку зображення, а також працювати з картинками низької якості.

Отже, елементом співпадіння становиться найяскравіший сегмент зображення.

1.3.3 Функції відповідності шаблонів

Метод відповідність шаблонів, являється загальним методом, для роботи якого використовуються функції.

TM_SQDIFF(Squared Difference) – це функція, яка обчислює суму квадратів різниць між піксельними значеннями шаблону та відповідних областей цільового зображення. Особливість полягає в тому, що мінімальне значення вказує на найяскравішу область, яка найбільше схожа на шаблон.

TM_SQDIFF_NORMED(Normalized Squared Difference) – метод обчислює суму квадратів різниць, але вони нормалізуються за допомогою величини шаблону, що зменшує вплив розміру шаблону на результат порівняння.

TM_CCORR(Cross Correlation) – функція використовує кореляції між шаблоном та вихідним зображенням, функція знаходить області, де найбільша відповідність з шаблоном.

TM_CCORR_NORMED(Normalized Cross Correlation) – нормалізований варіант функції “TM_CCORR”, що зменшує вплив абсолютних розмірів.

TM_CCOEFF(Correlation Coefficient) – ця функція використовує кореляційний коефіцієнт між шаблоном та окремими областями зображення. Розраховує відхилення від середнього значення, тому позитивні значення показують високу відповідність з шаблоном.

Для розробки додатку було обрано функцію TM_CCOEFF_NORMED тому що, перевага функції в праці з зображеннями низької якості. Гра MIR M: Vanguard & Vagabond вважається мобільної, але портована на персональні комп’ютери, но текстури та об’єкти не змінювали з мобільної версії, тому на комп’ютерах якість зображення низка.

Як працює метод з функцією TM_CCOEFF_NORMED.

Як було зазначено для праці методу відповідності шаблонів потрібно два зображення: вихідне зображення та шаблон за яким буде порівнюватись. Щоб визначити відповідну область, виконується порівняння зображення шаблону з вихідним зображенням, пересунувши його.

Рис. 1.3 Вихідне зображення, яке робить додаток

Визначимо вихідний шаблон, як I та шаблон, як T, а матрицю результатів як R.

Рис. 1.4 Шаблон “піктограми ремонту”

Для кожного розташування T понад I зберігається метрика в матриці результатів R. Кожне розташування (x,y) в R містить показник відповідності.

$$R(x,y) = \frac{\sum_{x',y'}\left(T'(x',y') * I'(x+x',y+y')\right)}{\sqrt{\sum_{x',y'} T'(x',y')^2 * \sum_{x',y'} I'\,(x+x',y+y')^2}}, \qquad (1.1)$$

Де, R(x,y) – це коефіцієнт кореляції для позиції (x,y) між шаблоном T та вихідним зображенням I на позиції (x,y),

x та y – координати пікселя на зображенні,

x' та y'– координати пікселя на шаблоні,

$\mathrm{T}(x', y')$ – яскравість пікселя на шаблоні на координатній осі,

$\mathrm{T}(x + x', y + y')$ – яскравість пікселя на вихідному зображенні.

Для кожного розташування T над I обчислена метрика результату зберігається в матриці результатів, яка зберігає дані у негативних та позитивних значеннях R.

Рис. 1.5 Матриця результатів

Найяскравіші місця вказують на найбільші збіги. Як зазначено на зображенні, позначена червоним колом піктограма, являється місцем із найвищим значенням, тому це розташування вважається відповідним.

Отже, як було зазначено вище, функція добре виконує поставлене завдання. А саме:

- знаходить маленькі об’єкти;
- працює з об’єктами низької якості;

- зберігає дані у вигляді масива даних , який обробляється у негативні та позитивних значеннях, що сприяє швидкій обробці зображення.

1.3.4 Гістограма орієнтованих градієнтів(HOG)

Гістограма орієнтованих градієнтів (HOG – Histogram of Oriented Gradients) – це метод в комп'ютерному зору, який використовується для знаходження об'єктів на зображеннях. Метод дозволяє описати форму та текстуру об'єктів, використовуючи градієнти яскравості в зазначених областях зображення. Дескриптор функції – це спосіб опису зображення за певними критеріями: форма, градієнт, текстури зображення, вилучає “корисну” інформацію про зображення. Корисною інформацією вважаються певні класифікації об’єкту, а саме: колір, форма тощо. Наведу приклад: потрібно витягнути корисну інформацію з об’єкту, а саме монітор комп’ютера, аналізуються всі класи об’єкту (колір, форма, розмір) в цьому випадку корисною інформацією буде: форма та розмір, колір в цьому випадку не потрібен.

Розрахунок методу гістограми орієнтованих градієнтів не можливий без кроку попередньої обробки зображення, вище було зазначено, яка інформація обробляється та, які класи об’єкту використовуються для обробки, але це не може бути можливим, якщо не підготувати зображення. Отже розрахунок методу HOG можна поділити на такі етапи:

1) Попередня обробка зображення. Для коректної та ефективної у часі обробки зображення треба видозмінити зображення до потрібного розширення, наведу формати розширення зображення з якими метод ефективніше всього проводить обробку у вигляді таблиці:

Таблиця 1.3.

Формати розширення зображення

Формат	**Опис переваг форматів**
64 x 64	Малі формати швидко оброблюються та потребують ресурсів комп’ютерного обладнання.
128x128	Оптимальний формат зображення, який забезпечує баланс між якістю та швидкістю.
256x256	Оптимальний формат зображення, який забезпечує баланс між якістю та швидкістю.
512x512	Висока якість обробки зображення

Зазначу, що співвідношення розмірів об’єкту повинні бути фіксовані співвідношення сторін, тобто формати з нефіксованими розширеннями, наприклад 121x220, недопустимі. Також, використання великих форматів, а саме вище 512x512 можуть значно знизити швидкість обробки та збільшити вимоги до комп’ютерного заліза.

2) Обчислення градієнтів зображення. Надалі виконується процес визначення наявних змін яскравості пікселів на зображенні та зміни їх напрямків. Етапи обчислення градієнтів:

   Перетворення кольорового зображення у градації сірого для спрощення обчислень. Наступним кроком треба знайти горизонталі та вертикалі градієнтів, який обчислюється методом Соболя.

   Метод Соболя – метод використовує два ядра, одне для визначення горизонтальних градієнтів, інше - для вертикальних. Результатом обчислення градієнта в кожній точці зображення стає комбінація горизонтальних та вертикальних градієнтів.

Формула розрахунку горизонтального похідної:

$$G_X = \begin{pmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{pmatrix} * A, \tag{1.2}$$

Формула розрахунку вертикальної похідної:

$$G_y = \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} * A. \tag{1.3}$$

Де,

* виконує операцію згортки у двомірну обробку сигналу.

Обчислення величини та напрямків. На основі обчислених горизонтальних та вертикальних градієнтів визначаються величини та напрямки змін яскравості в кожній точці зображення.

Координату x тут визначено як зростаючу «праворуч», а координату y – як зростаючу «донизу». У кожній точці зображення отримані наближення градієнта можливо об'єднувати, щоб отримувати величину градієнта, використовуючи наведену нижче формулу:

$$G = \sqrt{G_x^2 + G_y^2}, \tag{1.4}$$

Використовуючи отримані дані, розраховується напрямок градієнта:

$$\Theta = arctan\left(G_x^2, G_y^2\right). \tag{1.5}$$

Де,

$G_{x,y}$ – два зображення, які містять наближення горизонтальної та вертикальної похідної,

Θ – це результат обчислення кута напряму градієнту на кожній точці зображення,

arctan2 – це функція, яка повертає арктангенс відношення двох вказаних чисел. У випадку обробки зображень, ця функція використовується для обчислення кута напряму градієнту за його компонентами.

В результаті розрахунків утворилося абсолютне значення по X-градієнту, абсолютне значення Y-градієнту та величина градієнту.

Метод Лапласа використовується для виявлення точок яскравості на зображенні. Наведений метод являється одним із методів пошуку об'єкту завдяки аналізу границь зображення. Для обробки зображення використовуються фільтри ядра, застосовуються такі види обробки ядра:1D, 2D та 3D.

Матриця обробки 1D фільтра ядра:

$$D_x^{\rightarrow 2} = (1 \quad -2 \quad 1), \tag{1.6}$$

Матриця обробки 2D фільтра ядра:

$$D_{xy}^2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{1.7}$$

Матриця обробки 3D фільтр ядра D_{xyz}^3:

Розрахунок матриць

$$\text{по } x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \text{по } y = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -6 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \text{по } z = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{1.8}$$

Фільтри ядра застосовується до зображення для обчислення другої просторової похідної яскравості в кожній точці зображення. Результатом цієї операції допомагає виявляти зміни яскравості, такі як границі та контури на зображенні, оскільки другі похідні вказують на зміни яскравості вздовж обох осей x та y.

Отже, порівнюючи метод відповідності та метод гістограми орієнтованих градієнтів, перший метод ефективний для точного визначення зазначених об'єктів на зображенні, а метод градієнтів використовується для загального виявлення об'єктів без використання конкретних шаблонів. Метод відповідності шаблонів має можливість працювати з різним розширенням зображення та втрачаючи якості та швидкості обробки, тоді як метод градієнтів робить акцент на роботі з різними масштабами. Головною перевагою методу є можливість працювати з маленькими об’єктами, що являється ключовою перевагою в розробці методики підтримки ігрового процесу блокчейн ігор на основі обробки зображень. Метод градієнтів має більшу здатність адаптуватися до різних умов освітлення та орієнтації об'єктів на зображенні в порівнянні з методом відповідності шаблонів. Головною перевагою методу є можливість працювати без шаблону потрібного об’єкту, але метод значно програє в аналізі об’єктів низької якості та маленькими за розмірами.

1.4 Аналіз особливостей підтримки ігрового процесу в блокчейн іграх

Для початку треба встановити, яким чином проводиться підтримка ігрового процесу в блокчейн іграх, в яких конкретно блокчейн іграх ця підтримка доречна. Підтримка в блокчейн іграх – це алгоритм прописаних дій, які автоматизують головну частину ігрового процесу, як правило це однотипні дії, які треба повторювати від разу в разу. Підтримку ігрового процесу можна прирівняти до бота. Ботів можна розділити на певні категорії:

Чат-бот – додаток, які розроблені для автоматичної комунікації з іншими користувачами через чат-інтерфейс. Алгоритм виконує прості дії, наприклад надання інформації тощо.

Інтернет-бот – додаток, який автоматично може сканувати веб-сторінки для збору певної інформації, індексація даних для пошукових систем, виконування певних дій на веб-сторінках.

Боти соціальних мереж – додаток, який використовується для автоматичного керування групами в соціальних мережах. В соціальній мережі Telegram боти використовуються, як заміна менеджерів, наприклад, запис у перукарню здійснюється завдяки ботам, це суттєво збільшує ефективність часу з боку користувача, та економію коштів з боку перукарні.

Ігрові боти – алгоритм, який автоматизує головну частину ігрового процесу, як правило це однотипні дії, які треба повторювати від разу в разу. У випадку з предметною областю проекту це видобуток ресурсів та подальший обмін на криптовалюту.

Розроблювальну методику можна класифікувати саме, як ігровий бот, але також, зазначу, що розроблювальна методика саме автоматизує процес гри тому що, багато дій, які залежать від користувача, наприклад:

- запуск додатку;
- початкове налаштування додатку;
- налаштування персонажу;
- перезапуск комп’ютерної гри за потребою;

Автоматизація – у контексті комп’ютерних ігор термін означає, що користувачем будуть використанні боти або певні скрипти, які виконують запрограмовані алгоритми дій, наприклад видобуток певних ігрових ресурсів, але частково, втручання людини на певних етапах роботи бота необхідне. Тобто, автоматизацією можна назвати дії, які повністю не може виконати програма.

Автоматично – у контексті комп'ютерних ігор термін означає, що програма виконується без втручання користувача, тобто автоматична розсилка повідомлень в ігровий чат гри з ціллю, щось продати.

Але автоматизація в може бути впроваджена на не всі блокчейн ігри, через специфіку ігрової механіки для якої розробляється методика. Для порівняння були обрані блокчейн ігри з різними ігровими механіками та жанрами.

Опис гри для представлення, яким чином відбується видобуток криптовалюти.

CryptoKitties – це онлайн гра у картковому стилі на основі блокчейну Ethereum, де користувачі можуть купувати, продавати, розмножувати та колекціонувати цифрових котиків, які представлені у вигляді невзаємозамінних токенів (NFT). Головною ідеєю гри є створення унікальних цифрових котиків, які мають свої унікальні характеристики, які можуть бути успадковані їх нащадками. Кожен котик має унікальний генеративний код, який визначає його зовнішність і характеристики.

Користувачі можуть розмножувати своїх котів та створювати нових котів, які також будуть мати нові характеристики, успадковані від батьків. Котики можуть бути продані або куплені за криптовалюту Ethereum, що робить їх унікальними цифровими активами з власними цінностями.

Ni No Kuni: Cross Worlds – це онлайн гра з ізометричною перспективою на основі блокчейну, де користувачі можуть отримувати два типи невзаємозамінних токенів (NFT): теритові жетони та жетони астеріта. Теритові жетони можна отримати просто граючи в гру та проходження сюжету або виконувати специфічні місії. Жетони астеріта можна отримати гаючи змагання проти інших гравців, після успішного раунду користувачам надається ігрова валюта астеріт, який можна обміняти на криптовалюту.

MIR4 – це онлайн гра з ізометричною перспективою на основі блокчейну Hydra, в грі представленні жетони, які можна заробляти шляхом видобутку ігрових предметів або прокачувати ігрових персонажів та продавати. Ігрові предмети в залежності від цінності можна продати на торговому порталі гри, де інші користувачі придбають за

криптовалюту предмети. В MIR4 популярним є прокачка та продаж персонажу, користувачам, яким не хочуть втрачати багато часу.

MIR M: Vanguard and Vagabond – це онлайн гра з ізометричною перспективою на основі блокчейну. Головний спосіб видобутку криптовалюти це обмін ігрових ресурсів на криптовалюту DRONE, процес видобутку здійснюється в спеціальній локації, де з'являється специфічна руда. Також, криптовалюту можна отримувати добуваючи з монстрів предмети та виставляти на продаж до торгової площадки, де інші користувачі придбають предмет.

Нижче наведено таблицю, яка виділить переваги впровадження до блокчейн ігор розроблювальну методику.

Таблиця 1.4

Сумісність методики та блокчейн ігор зазначених в списку

Назва гри	**Особливості впровадження методу**
CryptoKitties	Через специфіку роботи розроблювального методу в цю гру не можливо впровадити підтримку ігрового процесу тому що, розроблювальна методика спеціалізується на іграх з ізометричною перспективою. Також, гра не адаптована під багаторазово повторення дій, за якими можна розробити певний алгоритм.
Ni No Kuni	Впровадження методики частково можливе, а саме впровадження можливе тільки, якщо користувач видобуває криптовалюту за рахунок проходження місії з монстрами. Методика підтримує тільки підтримку проти штучного інтелекту або AI (Artificial intelligence).
MIR4	Метод ефективно буде працювати на процесі прокачування персонажу для подальшої продажі.

Продовження таблиці 1.4

Сумісність методики та блокчейн ігор зазначених в списку

Назва гри	**Особливості впровадження методу**
MIR M	Метод ефективно буде працювати на видобутку ігрових ресурсів з спеціалізованих локацій.

Слід зазначити, що за використання ботів в блокчейн іграх можна отримати ігрову блокіровку персонального профілю користувача. Але через особливість ігрової механіки гри MIR M: Vanguard and Vagabond, а саме прокладення авто-маршруту при натисканні певних функцій ігрового персонажу. Методика розроблялась на основі цих даних, тому отримати ігрову блокіровку майже не можливо тому що, алгоритм робить все чітко, як би це робив звичайний користувач.

Отже, боти підтримки в блокчейн іграх представляють собою інтелектуальні програмні агенти, які надають допомогу гравцям у різних аспектах гри. Вони призначені для вирішення запитань, надання інформації, вирішення технічних проблем, надання порад щодо до ігрових механік та підвищення ефективності виконання процесів у грі. Боти можуть надавати підтримку користувачам, які не можуть проводити весь вільний час в грі, але просуватись є бажання просуватись далі по грі.

Боти, також забезпечують інформацією та оновленнями користувачів щодо новин та подій у грі, надають допомогу в торгівлі та перегляді ринку в разі присутності у грі цифрових активів на блокчейні, таких як NFT. Крім того, вони можуть вести комунікацію з гравцями через чат-інтерфейс, відповідати на запитання та встановлювати зв'язок з розробниками гри. Все це робить ботів підтримки цінними інструментами для полегшення досвіду гравців у світі блокчейн ігор, надаючи доступ до необхідної інформації, вирішуючи проблеми та забезпечуючи ефективну взаємодію з ігровим середовищем.

2 РОЗРОБКА МЕТОДИКИ ПІДТРИМКИ ІГРОВОГО ПРОЦЕСУ БЛОКЧЕЙН ІГОР НА ОСНОВІ ОБРОБКИ ЗОБРАЖЕНЬ

2.1 Розробка моделі підтримки ігрового процесу блокчейн ігор

Для реалізації ідеї методики потрібно зазначити, який метод обробки зображення доречно використовувати. Було створено список графічних особливостей предметної області для якої розробляється метод підтримки ігрового процесу.

Список графічних особливостей гри MIR M: Vanguard and Vagabond:
Маленький головний екран гри – розроблювальна методика передбачає використання чотирьох відкритих вікон гри, тому кожне ігрове вікно відкривається в розширенні 980 x 520. Тобто, екран має стандартне розширення 1980 x 1080 розділяємо його на чотири частини(див.рис.2.1), отримаємо маленькі вікна.

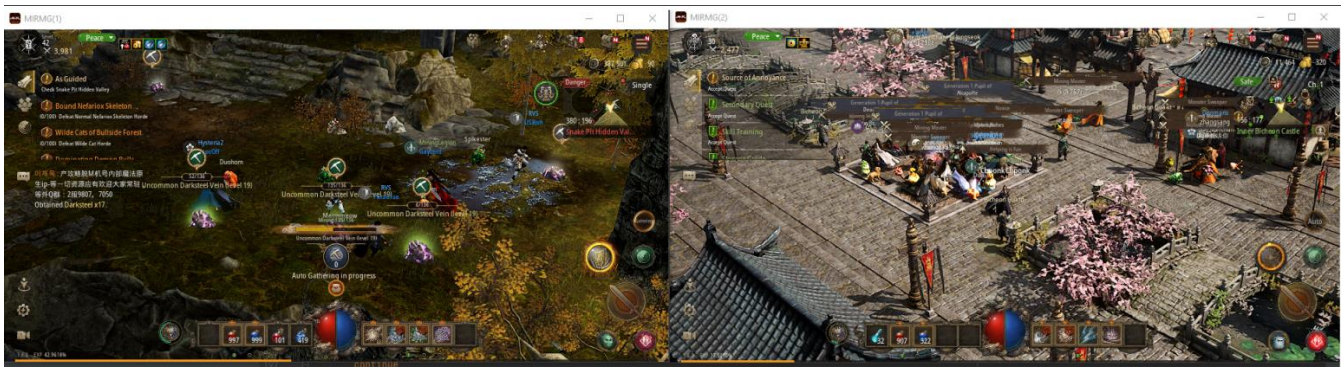


Рис. 2.1 Головні екрани гри

Погана якість текстур зображення – гра була реалізована для смартфонів, також була офіційно портована і на комп’ютер, але якість текстур залишився для рівня обладнання смартфонів. Також великим фактором заниження якості текстур об’єктів, являється розширення головного екрану 920 x 520.

Низька контрастність кольорів – спеціалізована локація для видобутку руди має темні кольори, через це деякі методи обробки зображень можуть давати низьку якість обробки.

Зазначені графічні характеристики комп'ютерної гри відображають унікальні параметри, які мають великий вплив на ефективність методів обробки зображень в цих умовах. Отже, важливо вибрати метод, який відповідає специфіці графіки та здатний забезпечити високоякісний аналіз у цьому контексті.

В ході розробки методики підтримки ігрового процесу блокчейн ігор, було проаналізовано два методи обробки зображення, а саме: метод відповідності шаблонів та метод гістограма орієнтованих градієнтів(HOG).

Таблиця 2.1

Порівняльна таблиця методу гістограми орієнтованих градієнтів

Переваги	**Недоліки**
Ефективний для розпізнавання складних об'єктів, оскільки враховує текстурні та структурні особливості.	Використання більшої кількості обчислювальних ресурсів для розрахунків.
Менш чутливий до змін освітлення об’єктів	Може бути чутливим до фонового середовища, що може вплинути на точність зіставлення об’єкту.
Багатогранність методу, включаючи розпізнавання обличчя, метод відстеження об'єктів.	

Отже, за отриманими даними можна зазначити, що метод гістограми орієнтованих градієнтів добре обробляє зображення з чіткої текстурною властивістю, а саме: зображення з фігурами людей, тварин, вуличні сцени тощо. Цей метод не підходить для обробки предметної області проекту, через “брудний” фон гри.

Таблиця 2.2

Порівняльна таблиця методу відповідності шаблонів

Переваги	**Недоліки**
Відносно легко реалізувати, оскільки використовує простий підхід з порівнянням піксельних значень.	Метод може бути чутливим до змін освітлення та перспективи об'єкта на зображенні.
Добре працює для виявлення невеликих об'єктів у зображенні.	Не ефективний для об'єктів, де змінюється ракурс камери об’єкту може змінюватись.
Якісно обробляє зображення, якщо є багато варіацій одного об’єкту	

Результатом, порівняння обох методів для обробки зображення, суттєву перевагу отримав метод відповідності шаблонів, через свою особливість працювати з зображеннями низької якості, зображеннями маленьких розмірів тощо.

Була розроблена математична модель методу відповідності шаблонів. Модель відображає як порівнюється шаблон з вихідним зображенням.

Нехай I – це вихідне зображення, T – це шаблон, за яким буде проводитись його пошук на вихідному зображенні. Розмір T – w x h, розмір вихідного зображення I – W x H.

Розрахунок математичної операції кореляції між вихідним зображенням I та шаблоном T:

$$R(u,v) = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} I(x+u, y+v) * T(x,y)\,. \qquad (2.1)$$

Де,

$R(u,v)$ – представляє значення кореляції між шаблоном T та пікселями вихідного зображення I.

Операція сумування проводиться по всім пікселям шаблону T – шаблону і відповідним пікселям вихідного зображення I, де (u, v) – це переміщення точок по вихідному зображенню I(див. рис. 2.2.). Нижче зображено матрицю вихідного зображення, на рисунку можна побачити, що пікселі спочатку йдуть по ряду, потім змішаються на один стовпець і далі прораховуються.

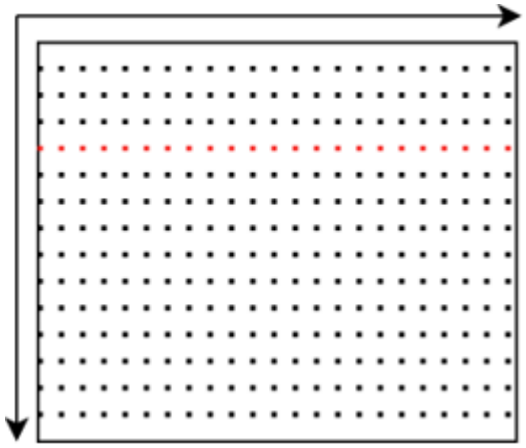


Рис.2.2 Вихідна матриця

Червоним відображено, пікселі, які результуюча кореляційна матриця зробить яскравими.

Після розрахунку значень кореляції $R(u, v)$ наступним кроком визначається місцеположення шаблону T на вихідному зображенні I за допомогою функції пошуку максимуму minMaxLock.

Математична модель визначає процес відповідності шаблонів, де розрахунки кореляції використовуються для знаходження найбільшого співпадіння між вихідним зображенням та шаблоном.

Нижче представлено формальну модель роботи методу, в якому відображена структура методу відповідності шаблонів.

Формальна структура пошуку потрібного об’єкту на зображенні:

$$\text{Матриця об}'\text{єкту} = \text{Метод}(\text{Вихідне зобр.}, \text{Шаблон}, \text{Функція кореляції}), \quad (2.2)$$

Формальна структура пошуку точки максимуму на матриці кореляції:

$$\text{Максимальне значення} = \text{функція максимуму}(\text{Матриця об}'\text{єкту}). \quad (2.3)$$

Також, слід зазначати додаткові методи та функції, які були використані при розробці методики підтримки ігрового процесу блокчейн ігор.

Бібліотека NumpPy – інструмент у мові програмуванні Python для роботи з масивами, математичними функціями та обробки чисельних даних. В бібліотеці можна зазначити такі особливості:

- масиви – масиви дозволяють виконувати операції над багатовимірними масивами даних;
- векторизація операцій – використовується для розрахунку математичних операцій на масивах даних без необхідності застосовувати цикли, що робить програмний код оптимізованим;
- математичні функції – бібліотека математичних функцій, а саме тригонометричні, логарифмічні тощо;
- робота з лінійною алгеброю – функція для лінійної алгебри, включаючи операції над матрицями, розв’язування систем лінійних рівнянь і знаходження власних значень та векторів;

В розробці методу підтримки ігрового процесу використовується для зберігання розмірів зображення кожного відкритого екрану гри.

Функція пошуку максимуму та мінімуму minMaxLock – використовується у бібліотеці OpenCv для знаходження мінімального та максимального значень матриці. Функція складається за такою структурою:

- “matrix” – це вхідна матриця, для якої ви шукаєте мінімальне та максимальне значення;
- “min_val” – мінімальне значення в матриці;
- “max_val” – максимальне значення в матриці;
- “min_loc” – координати (x, y) мінімального значення у матриці;
- “max_loc” – координати (x, y) максимального значення у матриці;

Функція MSS – бібліотека, яка використовується для знімків екрану комп’ютеру шляхом захоплення дисплея. Це дозволяє робити зображення екрану дуже швидко та зберігати їх для подальшої обробки.

Функція Grab – функція, яка використовується сумісно з функцією MSS, Grab використовується для отримування знімків екрану.

Також застосовувалися методи графічного представлення розробленого додатку, а саме UML-діаграми:

Діаграма класів – це графічний спосіб відображення структури класів і зв'язків між ними в об'єктно-орієнтованій системі. Діаграма відображає класи програми, їх атрибути (змінні) і методи (функції). Також зв'язки між класами, а саме:

- залежності – визначає, залежні об’єкти між собою, наприклад під класи, які спадкуються від головного класу;
- асоціації – відображає пов’язаність класів між собою;
- агрегації – визначає певну частину іншого класу;
- композиції – клас, який складається з інших класів;

Діаграма варіантів використання – діаграма, яка представляє візуальне представлення про то, як користуватись додатком звичайному користувачу.

2.2 Розробка методу підтримки ігрового процесу блокчейн ігор

Для розроблювальної методики підтримки ігрового процесу, було розроблено формальний алгоритм дій персонажу коли працює методика. В алгоритмі поетапно розписана кожна дія персонажу в різній ситуації. Також для повного розуміння з якими даними працює розроблювальна методика було розроблено таблицю з необхідними шаблонами.

Узагальнений алгоритм дій персонажу під час роботи додатку.

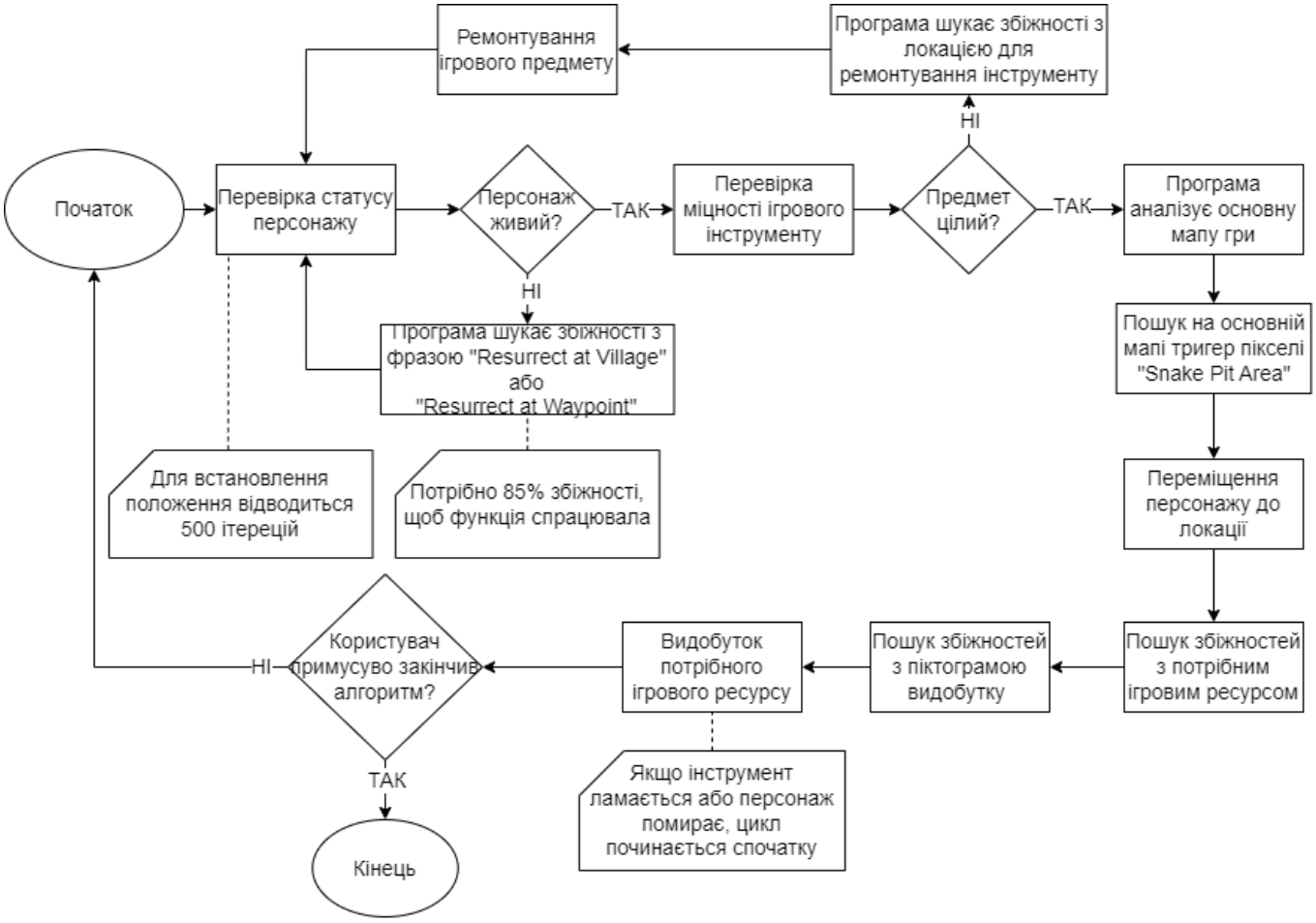


Рис. 2.3 Формальний алгоритм роботи додатку

Перший крок алгоритму.

1. Програма перевіряє статус персонажу – це потрібно для того, щоб виключити затримку алгоритму.

Другий крок.

2. Програма перевіряє живий персонаж чи ні – перевірка проводиться завжди коли персонаж нічого не робить 20 секунд. Якщо персонаж мертвий, то програма шукає збіжності з піктограмою “відродження”, після знаходження збіжності, персонаж відроджується та алгоритм починається спочатку.

Третій крок.

3. Програма перевіряє ігровий інструмент – перевірка проводиться завжди коли персонаж нічого не робить 20 секунд. Якщо інструменти зламані, то на головному екрані буде відображена піктограма “починка ігрового інструмента”. Після ремонтування ігрового інструменту, алгоритм починається з етапу перевірки статусу ігрового персонажу.

Четвертий крок .

4. Програма шукає потрібну локацію для видобутку руди – порівнюється на міні мапі гри шаблон піктограми “локація для видобутку руди”, після знаходження персонаж за допомогою автоматичної навігації(автоматична навігація – це особливість комп’ютерної гри) переходить до локації.

П’яти крок.

5. Програма шукає потрібну руду для подальшого видобутку – програма порівнює шаблони з піктограмою “руда для видобутку”.

Шостий крок.

6. Програма шукає піктограму видобутку руди – додаток робить співвідношення шалону піктограми з вихідним зображенням, після успішного співвідношення персонаж починає видобуток руди. Слід зазначити, що особливістю гри є функція авто-видобутку руди, тобто коли додаток перевірив статус персонажу

та ігрових інструментів, в моменті коли персонаж йде до потрібної локації алгоритм 170 секунд(стільки часу потрібно персонажу, щоб перейти до потрібної локації) знаходиться в стані очікування. Після виходу програму зі стану очікування, персонаж видобуває руду до моменту смерті або поломки ігрового інструменту.

Шостий крок.

7. Цикл працює до моменту смерті персонажу або несправності інструменту, яким добувається руда.

Тобто, алгоритм зациклений. Також, алгоритм можна виключити примусово натиснувши гарячу клавішу “F4”. Зазначу, що якщо персонаж не буде нічого робити протягом 15 секунд, то алгоритм починається спочатку.

Для більш детального пояснення алгоритму, було розроблено таблицю з потрібними для роботи методики піктограмами. Піктограми – це скріншоти з комп'ютерної гри до якої розробляється методика підтримки ігрового процесу, чим більше зроблено шаблонів для порівняння тим більше ефективності співвідношення вихідного зображення до шаблону.

Таблиця 2.3

Потрібні для обробки шаблони

Назва	**Опис**
Скріншот піктограми “локація для видобутку руди”	Робиться для того, щоб встановити місце видобутку руди
Скріншот піктограми “руда”	Робиться для видобутку руди
Скріншот піктограми “ ремонтування ігрового інструмента”	Робиться для того, щоб методика працювала довгий проміжок часу
Скріншот піктограми “відродження персонажу”	Робиться для перевірки статусу персонажу

Продовження таблиці 2.3

Потрібні для обробки шаблони

Назва	**Опис**
Скріншот піктограми “видобуток руди”	Робиться для того, щоб персонаж почав видобутку руди

Це ключові шаблони, які використовуються для співвідношення з вихідним зображенням. Зазначу, що чим більше буде зроблено різних варіацій шаблонів, то більш якісне буде розрахована подібність. Було виділено такі критерії за якими потрібно робити варіації шаблонів:

- різне розширення – кожен шаблон повинен бути представлений дуже докладно, розширення одна з ключових класифікацій;
- різний масштаб зображення – додавання шаблонів в різних масштабах сприяє, більш детальної обробки зображення;

Отже, розроблений алгоритм надає формальний опис роботи методики, який коротко відображає головне ідею розроблювального методу.

2.3 Підготовка шаблонів для співставлення

Для того, щоб методика працювала потрібно провести попередню підготовку, а саме зробити шаблони за якими буде порівнюватись вихідне зображення.

Шаблони класифікувати за певними типами:

Шаблони руди – до шаблонів руди відносяться всі зображення, які пов’язані з видобутком ігрового ресурсу.

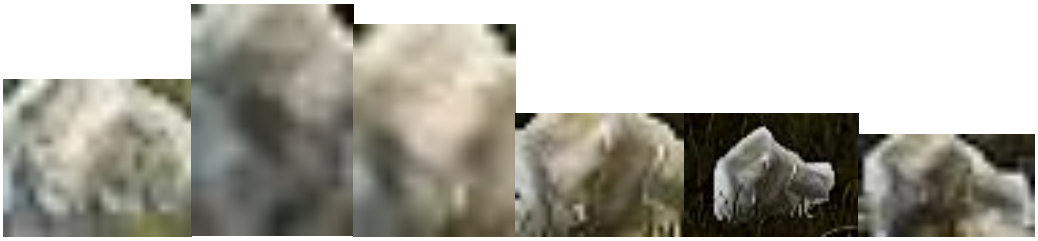


Рис.2.4 Шаблони руди

Для проекту було зроблено шість варіацій руди, це зроблено для того, щоб методика мала змогу аналізувати всі місця де генерується ігровий ресурс тому що, кожна жила має своє місце генерації.

Шаблони персонажа – всі, які пов’язані з персонажем на пряму, а саме його статус в випадку цього проекту це живи чи мертвий.

В грі MIR M: Vanguard & Vagabond є дві варіації відродження персонажу перша(див.рис.2.5.), яка відроджує персонажа в місті де можна відремонтувати ігровий інструмент, тобто це місто можна назвати головним хабом.

Рис.2.5 Відродження персонажу в місті

Друга піктограма відродження персонажу(див.рис.2.6.), виконує функцію відродження персонажу не в головному місті, а як другорядний хаб, тобто це місто де не велика безпечна ігрова зона, де нема ключових для роботи методики місць для ремонтування ігрового інструменту.

Рис.2.6 Відродження персонажу не в головному місті

Шаблони дії – шаблони, які пов’язані з діями, які потрібно виконати персонажу коли з’являється потреба, а саме ремонт ігрового інструменту та видобуток руди. В цьому випадку методика проводить обробку зображення за такими піктограмами “ремонтування ігрового інструменту"(див.рис. 2.7.) та “видобуток руди”(див.рис. 2.8.).

Наведена нижче піктограма, з’являється на головному екрані гри, після того як міцність інструменту знизиться до 25%, але програма запуститься тільки після повної поломки інструменту. Це зроблено для того щоб, можна було більше видобути руди для обміну.

Рис.2.7 Піктограма ремонтування ігрового інструменту

Наведена нижче піктограма, з’являється на головному екрані гри, після того як персонаж підходить до потрібної для видобутку руди. Методика починає роботу з перших двох кроків алгоритму(див.рис.2.3.), після того як персонаж вирушить до локації для видобутку руди, методика переходить у стан очікування на 170 секунд (час було зазначено досвідченим методом). Персонаж приходить до місця видобутку руди, на головному екрані відображається піктограма видобутку руди, програма робить обробку зображення після успішного зіставлення зображень, персонаж починає видобуток руди.

Рис.2.8 Піктограма видобутку руди

Шаблони локації – шаблони, піктограми яких відображаються на міні-мапі гри(див.рис.2.9.), яка знаходить у верхній праві частині інтерфейсу гри.

Рис.2.9 Скріншот головно екрану гри з міні-мапою.

Червоним колом акцентовано увагу на міні-мапі, зараз можна порівняти наскільки шаблон буде маленьким. На цьому зображенні визначено один з двох прикладів шаблону, нижче наведено всі види шаблонів піктограм локації для видобутку руди.

Зображення на Рисунку 2.5. демонструє один із конкретних прикладів шаблону, який використовується в системі. Додатково, нижче наведено інші варіанти шаблонів, які також можуть використовуватись для позначення піктограм локацій для видобутку руди. Виділення уваги на міні-мапі дозволяє зорієнтуватися у розмірах шаблонів у контексті візуального представлення на карті чи екрані.

Рис.2.10 Вигляд піктограми коли персонаж на локації

Рис.2.11 Вигляд піктограми коли персонаж за межами локації

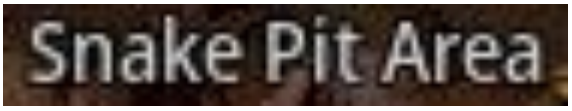


Рис.2.8 Масштабований варіант піктограми коли персонаж за межами локації

Зазначу, що для всіх шаблонів визначено 85% співпадіння за шаблонами, цей параметр було визначено дослідницьким методом, тобто при такої якості текстур та розширенні відкритих ігрових додатків встановлений параметр оптимальний.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДИКИ ПІДТРИМКИ ІГРВОГО ПРОЦЕСУ БЛОКЧЕЙН ІГОР НА ОСНОВІ ЛБРОБКИ ЗОБРАЖЕНЬ

3.1 Екранні форми додатку

У створеному програмному додатку відсутній повний графічний інтерфейс через те, що реалізація методики передбачає роботу без використання інтерфейсу користувача. Проте, для полегшення налаштувань області, що підлягає захопленню знімків за допомогою функції MSS, були впроваджені дві екрані-форми.

Екранні форми виконуються роль зручного визначення конфігурації області екрану, яка буде захоплюватись під час роботи програми. Вони дозволяють користувачеві вказати параметри захоплення, такі як розмір та розташування області на екрані. Це полегшує налаштування функціоналу, спрощуючи процес визначення конкретних параметрів захоплення знімків.

Незважаючи на відсутність повноцінного графічного інтерфейсу, впровадження цих екранів-форм спрямоване на підвищення зручності користування та налаштування програми, забезпечуючи необхідну функціональність для визначення області захоплення екрану для подальшої обробки.

Нижче представленні екранні форми розробленого додатку:

Перша форма(див.рис.3.1.) – робить скріншоти головного екрану на якому виконуються такі дії: видобуток руди, ремонтування інструментів, перевірка персонажу на живий чи мертвий.

Рис.3.1 Екранна форма головного екрану гри

Друга форма(див.рис. 3.2.) – робить скріншоти міні-мапи на якій виконується пошук ключових локацій, а саме: місця зародження руди та локацію для видобутку руди.

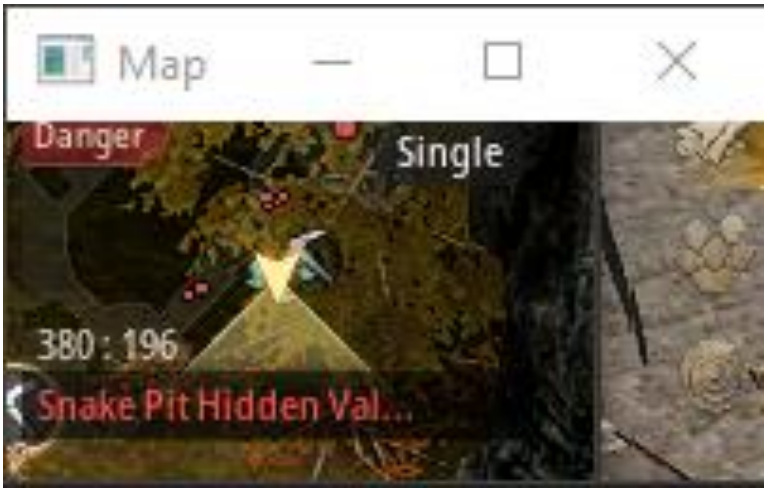


Рис.3.2 Екранна форма міні-мапи гри

Скріншоти зберігаються завдяки функції grab() з бібліотеки MSS є потужним інструментом для захоплення знімків екрана. Вона витягує інформацію про пікселі знімка, представляючи його у формі бінарних даних, де кожен піксель кодується за його кольором та іншими параметрами.

Отримані бінарні дані можуть бути перетворені у зображення за допомогою різних інструментів, таких як модуль PIL. Це може здійснюватися шляхом перетворення бінарних даних у відповідний формат зображення, що дозволяє подальшу обробку чи відображення.

Перетворення зображення у масив NumPy створює зручну форму представлення, де кожен піксель відображається числовими значеннями. Це дозволяє виконувати різноманітні операції обробки зображень, такі як зміни кольору, розміру чи застосування фільтрів, використовуючи функціональність NumPy для роботи з числовими даними зображення. Такий підхід дає можливість швидко обробляти зображення методом відповідності шаблонів.

3.2 UML представлення додатку

Було розроблено графічне відображення взаємодії типового користувача з розроблювальною методикою.

Діаграма варіантів використання – діаграма створена з ціллю відобразити весь процес використання розробленої методики підтримки ігрового процесу, від запуску додатку до обміну токенів на криптовалюту.

На Рисунку 3.3 зображена система, яка відображає процес роботи всієї системи загалом, приклад наведений для комп'ютерної гри MIR M: Vanguard And Vagabond.

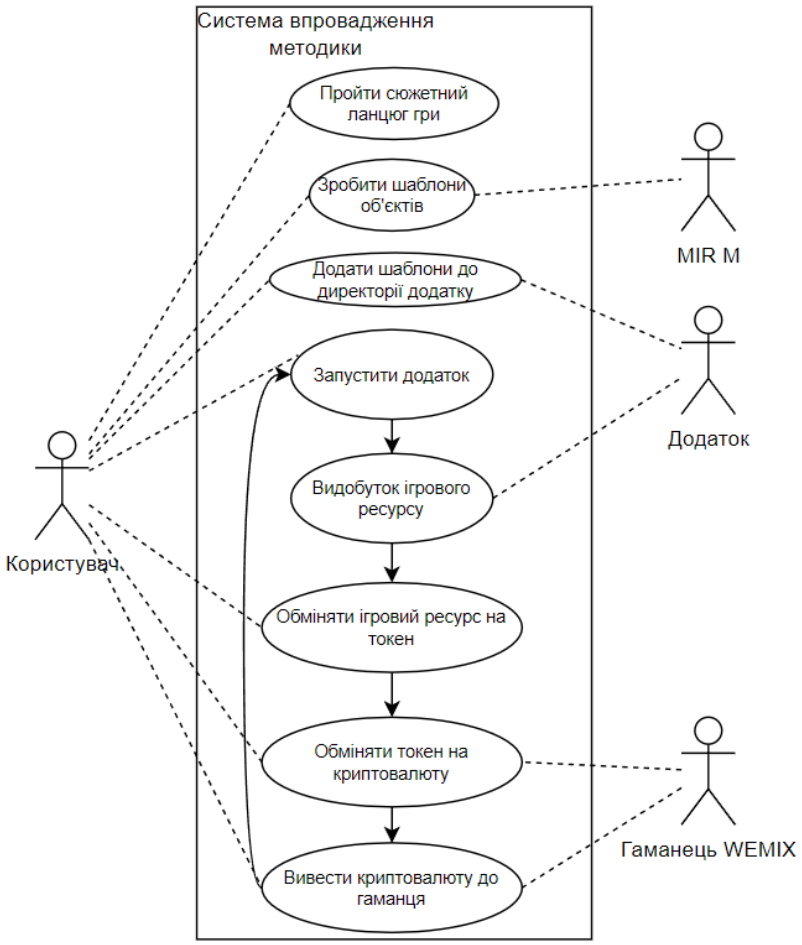


Рис.3.3 Система впровадження методики

WEMIX – крипто гаманець, який обмінює криптовалюти та виводить їх до гаманця. На діаграмі зазначені дії, які потребують одноразового втручання користувача та багаторазові.

Одноразові втручання користувача:

Пройти сюжетний ланцюг – необхідний пункт, якщо його не виконати гра не надасть доступу до спеціальної локації для видобутку руди.

Зробити шаблони об’єктів – шаблони об’єктів, які будуть оброблятися методикою.

Завантажити шаблони до директорії додатку – шаблони звантажуються до директорії, які в подальшому будуть співвідноситись до вихідних зображень додатку. Обов’язково треба називати шаблони таким чином:

- repairLogo – шаблон ремонту;
- locationLogo – локація для видобутку ігрової руди відображення піктограми, якщо персонаж знаходиться в межах локації;
- miningLogo – шаблон дії видобутку руди;
- snakePitArea1 – локація для видобутку руди відображення піктограми, якщо персонаж знаходиться за межами локації;
- resLogo – шаблон відродження персонажу після смерті;
- minedStoneLogo – ігровий ресурс, який видобувається, кожна піктограма має різний масштаб та розширення зображення;
- minedStoneLogo2 – ігровий ресурс, який видобувається, кожна піктограма має різний масштаб та розширення зображення;
- minedStoneLogo3 – ігровий ресурс, який видобувається, кожна піктограма має різний масштаб та розширення зображення;
- minedStoneLogo4 – ігровий ресурс, який видобувається, кожна піктограма має різний масштаб та розширення зображення;

- minedStoneLogo5 – ігровий ресурс, який видобувається, кожна піктограма має різний масштаб та розширення зображення;

Якщо завантажити скріншоти зображень під іншою назвою, додаток не буде працювати. В такому випадку, треба буде змінювати структуру змінних в яких зберігаються зображення.

Втручання користувача, які потрібно робити через певний інтервал:

- запустити додаток;
- обмінювати ігровий ресурс на токен;
- обмінювати токен на криптовалюту;
- виводити криптовалюту до гаманця;

Діаграма класів – діаграма відображає, як працює розроблена методика працює внутрішньо на рівні методів та класів.

На Рисунку 3.4 визначена діаграма класів, відображає роботи методів в додатку.

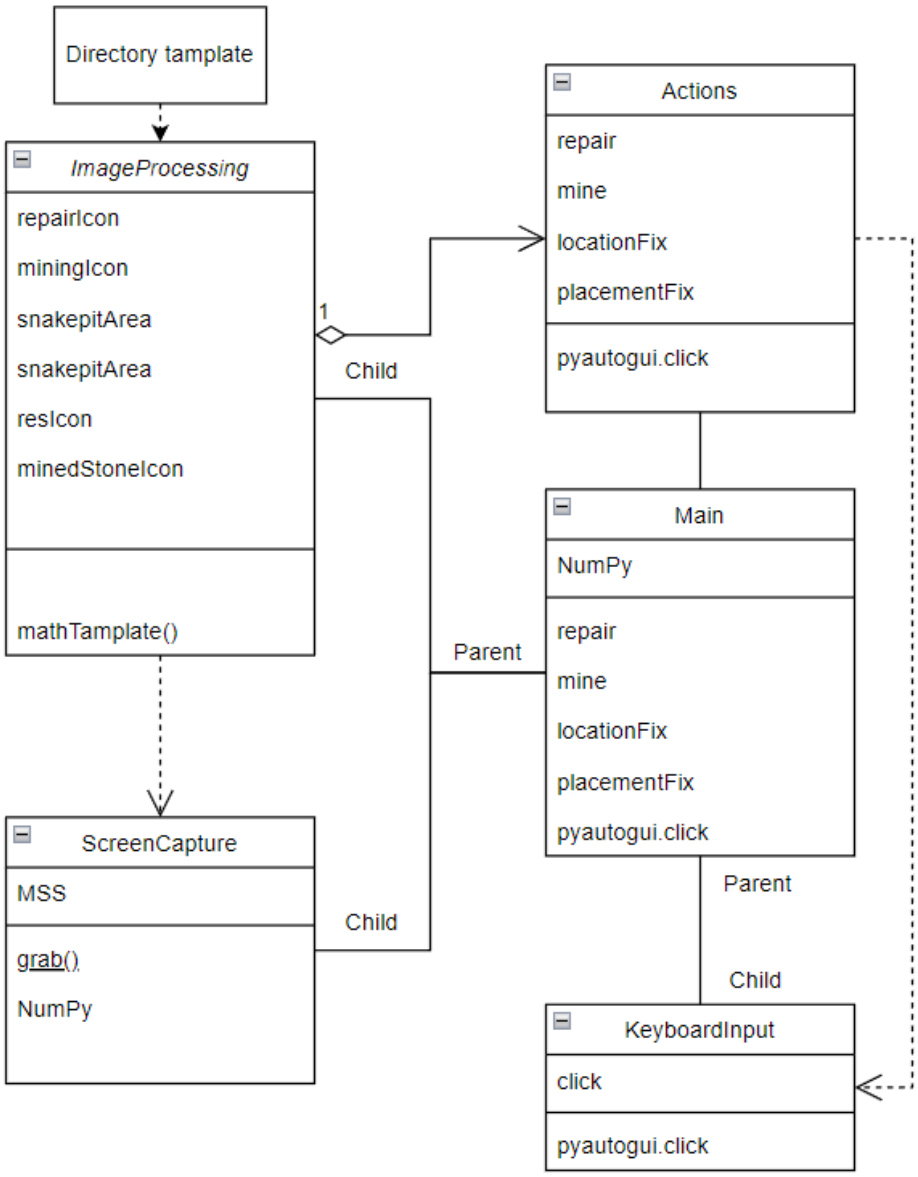


Рис.3.4 Діаграма класів

Програмний код можна розділили на п’ять ключовий класів.

ImageProcessing – має методи для обробки зображень: пошук певних іконок, шаблонів, а також виявлення елементів ігрового інтерфейсу. Даний клас з’єднаний зв’язком агрегації, що відображає перевагу класу да іншим.

Actions – класифікується, як головні дії додатку, виконує всі дії, які були зазначені в програмі, а саме: ремонт ігрового інструменту, видобуток ігрового

ресурсу, пошук локації для видобутку, пошук руди для видобутку. З’єднаний з класом ImageProcessing.

Main – включає основний цикл програми, отримання зображень з екрану і спостереження за подіями користувача. Має методи для ремонту, видобування та інших дій в грі. Головний клас в якому, розташовані всі методи програми, з’єднаний з кожним класом зв’язком асоціації.

ScreenCapture – відповідає за захоплення екрану та взаємодію з ним для отримання зображень, завдяки бібліотеці MSS та методу grab(). З’єднаний зв’язком залежності з класом ImageProcessing, надає вихідні скріншоти до методу обробки зображення.

KeyboardInput – визначає введення з клавіатури для управління програмою наприклад, натискання клавіш для активації функцій. З’єднана зв’язком залежності з класом Actions, виступає в ролі методу, який виступає в ролі рук додатку, виконує всі функція, які потребують взаємодію з клавішами клавіатури.

Дана діаграма є узагальненою, через те що, назви методів і класів були зведені до демонстраційного виду. Але розроблена модель чітко відображає поведінку додатку, та логічний зв’язок між класами, за яким функціонує алгоритм.

3.3 Визначення критеріїв ефективності процесу підтримки ігрового процесу

Ефективність в розробці методики – це ключовий фактор, який визначає успіх та користь для користувачів. Протягом розробки методики підтримки ігрового процесу блокчейн ігор, було зазначено такі критерії ефективності:

- час – критерій, за яким визначається скільки розроблювальна методика економить часу користувачу на видобуток криптовалюти;
- прибуток – критерій, за яким визначається наскільки розроблювальна методика збільшила видобуток криптовалюту;

- надійність – критерій, за яким визначається кількість часу без перервної праці методики. наведу фактори, які можуть заважати праці:
    - зовнішні фактори: раптове відключення світла, перезавантаження комп’ютеру, раптовий обрив інтернету;
    - внутрішні фактори: персонаж в грі завис і більше не функціонує, персонажа постійно вбивають в грі і метод не може нормально видобувати ресурс, гра перевантажилась та закрилась;
- оптимізація ресурсів – критерій за яким визначається навантаження на комп’ютерне обладнання та електроенергію;
- зручність у використанні – критерій, який визначає зручність використання комп’ютеру під час роботи методики;

Ці критерії являються оптимальними у розроблювальної методики тому що, кожен з них може надати потрібну статистичну інформацію, якщо вона потрібна. Але ключовими критеріями за якими оцінюється методика являються: час, прибуток, оптимізація ресурсів комп’ютера. Обґрунтовуючи вибір, зазначимо такі деталі про критерії.

Критерій часу – ключовий критерій тому що, головне завдання методики, щоб вона надавала видобувати криптовалюту користувачем, які не можуть цілодобово сидіти вдома та займатись видобутком ресурсу. За цим критерієм буде визначатись скільки часу в конкретних цифрах економить методика користувачам.

Прибуток – критерій, який доводить доцільність розроблювальної методики, тобто кількість прибутку за певний проміжок часу. Цей критерій дуже залежить від комп’ютерної гри та ціни на криптовалюту, щоб обґрунтувати вибір предметної області наявність цього критерія обов’язкова.

Оптимізація ресурсів комп’ютера – важливий критерій при розробці методики тому що, включає в себе дані про витрати електроенергії та порівнює їх з прибутком праці методики. Критерій обґрунтовує доцільність використання загалом.

Зазначимо, як працює заробіток криптовалюти в предметній області до якої розробляється методика, це надасть пояснення до даних, які використовуються при оцінці ефективності за різними критеріями. В грі MIR M: Vanguard And Vagabond, дуже простий алгоритм заробітку криптовалюти: спочатку користувачу потрібно пройти початкову сюжетну лінію для того, щоб відкрився доступ до спеціальної локації де випадково генерується руда. Після видобутку 110000 одиниць руди, користувачу потрібно обміняти її на спеціальний NFT предмет, цей предмет обмінюється на криптовалюту DRONE (див. таб.3.1).

Таблиця 3.1 демонструє результати порівняльного аналізу за критерієм часу. Переваги застосування методики полягають в наступному: автоматизовані засоби забезпечують цілодобовий видобуток криптовалюти без простоїв; у порівнянні з ручним процесом час видобутку на добу та кількість видобутих одиниць збільшились в 2,4 рази.

Таблиця 3.1

Оцінювання методики за критерієм часу

Показник	**Ручний видобуток криптовалюти**	**Видобуток з використанням методики**
Час на видобуток однієї жили	15 хвилин	15 хвилин
Кількість руди за одну жилу	1300 одиниць	1300 одиниць
Кількість руди за годину видобутку	5200 одиниць	5200 одиниць
Час роботи в день	10 годин	24 години
Час простою	14 годин	Немає простою
Результат праці за добу	52000 одиниць	124800 одиниць

В таблиці 3.2 представлено оцінку ефективності з точки зору критерію грошового еквіваленту. Одиницею виміру є токен – невзаємозамінний токен, який в подальшому обмінюється на криптовалюту DRONE. Розрахунки проводились при наступних початкових даних гри: кількість руди за одну жилу становить 1300 одиниць, кількість руди, яку можна отримати за годину видобутку становить 5200 одиниць, кількість руди, потрібної на обмін до NFT предмету, становить 110000 одиниць. Застосування розробленої методики для автоматизованого видобутку забезпечує гарантоване накопичення 1 токену за добу, чого не вдається досягнути при ручному видобутку. З урахуванням середньої вартості токену станом на грудень 2023 року при автоматизованому видобутку можна заробити в 2,88 рази більше, ніж при ручному. Представленні дані рахувались згідно роботі одного відкритого вікна гри. Методика може працювати з двома відкритими вікнами цієї гри, тому при відповідному технічному оснащенні (2 монітори) максимальний прибуток може становити до 1440 грн. на місяць.

Таблиця 3.2

Оцінювання методики за критерієм прибутку

Критерій	**Ручний видобуток криптовалюти**	**Видобуток з використанням методики**
Кількість руди за одну жилу	1300 одиниць	1300 одиниць
Кількість руди за годину видобутку	5200 одиниць	5200 одиниць
Кількість руди, потрібної на обмін до NFT предмету	110000 одиниць	110000 одиниць
Кількість токенів за добу	За добу не накопичується потрібна кількість	1 токен

Продовження таблиці 3.2

Оцінювання методики за критерієм прибутку

Критерій	**Ручний видобуток криптовалюти**	**Видобуток з використанням методики**
Накопичена сума за одну добу, яку можна обміняти	-	24 грн.
Накопичена сума за місяць	249,6 грн.*	720 грн. + 4 токени в пасивному режимі накопичення

* Розрахунки проведені за умови 5 денного робочого тижня людини, 22 робочих дні на місяць.

Результати прибутку за певний проміжок часу в таблиці 3.2 наведені без урахувань витрат на електроенергію. Оскільки при видобутку криптовалюти використовуються доволі потужні комп'ютерні системи, то витрати на електроенергію повинні обов'язково враховуватись при розрахунку прибутку на одиницю часу. Таблиця 3.3 містить розрахунки витрат електроенергії на видобуток криптовалюти. Витрати електроенергії за годину з використанням розробленої методики збільшуються за рахунок роботи додатку, що задіює додаткові ресурси як процесора, так і відеокарти, що призводить до збільшення витрат.

Таблиця 3.3

Визначення показників прибутку з урахуванням витрат на електроенергію

Критерій	**Ручний видобуток криптовалюти**	**Видобуток з використанням методики**
Витрати електроенергії за годину	0,2 kW h	0,24 kW h
Витрати електроенергії за робочий день	2 kW h	5,76 kW h
Витрати електроенергії за місяць	60 kW h	172,8 kW h

Продовження таблиці 3.3

Визначення показників прибутку з урахуванням витрат на електроенергію

Критерій	**Ручний видобуток криптовалюти**	**Видобуток з використанням методики**
Вартість витрат на електроенергію за годину, грн.	0,528 грн.	0,634 грн.
Вартість витрат на електроенергію за робочий день, грн.	5,28 грн.	15,206 грн.
Вартість витрат на електроенергію в середньому за місяць, грн.	116,16 грн.*	456,19 грн.

* Розрахунки проведені за умови 5 денного робочого тижня, 22 робочих дні на місяць.

При розрахунках бралась вартість електроенергії для населення 2,64 грн. за kW. Слід зазначити, що значення в таблиці 3.2 враховують показники витрат для апаратного забезпечення, визначеного в розділі 1. При запуску додатку на ноутбуці можна зменшити показник споживання електроенергії приблизно на половину відповідно до попередніх оцінок. Таким чином, з урахуванням витрат на електроенергію із застосуванням розробленої методики можна отримати на місяць прибуток в 263,81 грн. проти 133,44 грн. при ручному видобутку, що в 1,98 рази більше. При використанні методики в режимі 2 екранів прибуток на основі автоматизованого видобутку становить у 3,96 разів більше, ніж при ручному підході.

ВИСНОВКИ

1. Проаналізовано предметну область блокчейн ігор, особливості ігрового процесу та методи і засоби підтримки ігрового процесу в такому класі ігор. Визначено види зображень для обробки, які застосовано при розробці методики підтримки ігрового процесу блокчейн ігор.

2. Розглянуто методи обробки зображень, які можуть бути використані для підтримки ігрового процесу в блокчейн іграх, та виконано їх порівняльний аналіз. Для побудови методики використано метод зіставлення шаблонів.

3. Розроблено алгоритм обробки зображень блокчейн гри, в якому можна зазначити такі особливості роботи: автоматизація процесів, циклічність роботи, оптимізація часу.

4. Для перевірки працездатності методики розроблено додаток, який реалізує методику підтримки ігрового процесу в комп'ютерній грі MIR M: Vanguard and Vagabond та надає можливість автоматизованого видобутку криптовалюти.

5. Проведено аналіз результатів впровадження методики. Використання методики підтримки ігрового процесу блокчейн ігор на основі обробки зображень дозволило збільшити ефективність видобутку криптовалюти наступним чином: автоматизовані засоби забезпечують цілодобовий видобуток криптовалюти без простоїв; у порівнянні з ручним процесом час видобутку на добу та кількість видобутих одиниць збільшились в 2,4 рази. При цьому за місяць накопичується до 4 повних токенів в пасивному режимі, які можна обміняти в майбутньому без застосування видобутку в ці дні. З урахуванням витрат на електроенергію із застосуванням розробленої методики можна очікувати зростання прибутку в грошовому еквіваленті в 1,98 рази більше, ніж при ручному видобутку при одноекранному використанні методики і в 3,96 разів більше при двоекранному режимі.

ПЕРЕЛІК ПОСИЛАНЬ

1. EXBASE.IO [Електронний ресурс] // EXBASE.IO - create crypto wallet online | Cryptocurrency Wallet. – Режим доступу: https://exbase.io/uk/wiki/shho-take-blok.
2. INCRYPTED [Електронний ресурс] // Що таке Proof-of-Work (POW)?. – Режим доступу: https://incrypted.com/what-is-proof-of-work.
3. COINBASE [Електронний ресурс] // Графіки та ціни криптовалюти HYDRA. – Режим доступу: https://www.coinbase.com/ru/price/hydra.
4. MIR4 [Електронний ресурс] // WeMade. – 2021. – Режим доступу до ресурсу: https://store.steampowered.com/app/1623660/MIR4/.
5. Мінфін - все про фінанси: новини, курси валют, банки [Електронний ресурс] // МінфінМедіа. – Режим доступу: https://minfin.com.ua/currency/crypto.
6. What is Unspent Transaction Output (UTXO)? - GeeksforGeeks [Electronic resource] // GeeksforGeeks. – Mode of access: https://www.geeksforgeeks.org/what-is-unspent-transaction-output-utxo/.
7. Pfeiffer A. Blockchain Technologies and Games: A Proper Match? [Electronic resource] / Alexander Pfeiffer, Simone Kriglstein, Thomas Wernbacher // FDG '20:
8. International Conference on the Foundations of Digital Games, Bugibba Malta. – New York, NY, USA, 2020. – Mode of access: https://doi.org/10.1145/3402942.3402996.
9. Towards Understanding Player Behavior in Blockchain Games: A Case Study of Aavegotchi [Електронний ресурс] / Yu Jiang [та ін.] // FDG22: 17th International Conference on the Foundations of Digital Games, Athens Greece. – New York, NY, USA, 2022. – Режим доступу: https://doi.org/10.1145/3555858.3555883.
10. Hands-on Image Processing in Python [Електронний ресурс] / Sandipan Dey // 1st Publisher: Packt Publishing Limited, 2018
11. Suatap C. Development of Convolutional Neural Networks for Analyzing Game Icon and Screenshot Images [Електронний ресурс] / Chayanin Suatap, Karn Patanukhom

// International Journal of Pattern Recognition and Artificial Intelligence. – 2022. – Режим доступу: https://doi.org/10.1142/s0218001422540234.

12. PRIAMITSYN V. Y. CRYPTOCURRENCIES: ESSENCE AND PROSPECTS OF LEGAL REGULATION [Електронний ресурс] / V. Yu PRIAMITSYN, Ye S. ZOLOTAROVA // Law and Society. – 2023. – № 1. – С. 221–227. – Режим доступу: https://doi.org/10.32842/2078-3736/2023.1.32.
13. Experiences of Blockchain Technology Users [Електронний ресурс] / Carl John Cuagdan [та ін.] // 4th Northern Philippines Business Research Conference hosted by Northwestern University – 2023.
14. Introduction to Blockchain Technology [Електронний ресурс] / Suyel Namasudra // In book: Blockchain and its Applications in Industry 4.0 (pp.1-28) – 2023.
15. Chen J.-T. Blockchain and the Feature of Game Development [Електронний ресурс] / Jiun-Ting Chen // Lecture Notes in Electrical Engineering. – Singapore, 2020. – С. 1797–1802. – Режим доступу: https://doi.org/10.1007/978-981-15-3250-4_239 (дата звернення: 14.12.2023). – Назва з екрана.
16. Guillen G. Digital Image Processing with Python and OpenCV [Електронний ресурс] / Guillermo Guillen // Sensor Projects with Raspberry Pi. – Berkeley, CA, 2019. – С. 97–140. – Режим доступу: https://doi.org/10.1007/978-1-4842-5299-4_5 (дата звернення: 14.12.2023). – Назва з екрана.
17. Gao S. An empirical study on the adoption of blockchain-based games from users' perspectives [Електронний ресурс] / Shang Gao, Ying Li // The Electronic Library. – 2021. – Т. 39, № 4. – С. 596–614. – Режим доступу: https://doi.org/10.1108/el-01-2021-0009 (дата звернення: 14.12.2023). – Назва з екрана.
18. Коновал М.М., Золотухіна О.А. "Розробка методики підтримки ігрового процесу блокчейн ігор на основі обробки зображень" // Науково-практична конференція «Проблеми комп'ютерної інженерії», 10 грудня 2023 року, Державний університет інформаційно-комунікаційних технологій, Київ, Україна (с.137).

19. OpenCV: Operations [Електронний ресурс] // OpenCV documentation. – Режим доступу: https://docs.opencv.org/3.4/d2/de8/group__core__array.html#gab473bf2eb6d14ff97e89b355dac20707.
20. OpenCV Template Matching (cv2.matchTemplate) - PyImageSearch [Електронний ресурс] // PyImageSearch. – Режим доступу: https://pyimagesearch.com/2021/03/22/opencv-template-matching-cv2-matchtemplate/.
21. Histogram of Oriented Gradients explained using OpenCV [Електронний ресурс] // LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow with examples and tutorials. – Режим доступу: https://learnopencv.com/histogram-of-oriented-gradients/.
22. DRONE policy [Electronic resource] // Privacy Policy. – Mode of access: https://cache.wemixnetwork.com/wemixnetwork/whitepaper/drone/index.html.
23. TOKENS [Electronic resource] // MIR M: VANGUARD AND VAGABOND. – Mode of access: https://mirmglobal.com/tokens.
24. DOGMA - policy [Electronic resource] // Privacy Policy. – Mode of access: https://cache.wemixnetwork.com/wemixnetwork/whitepaper/dogma/index.html.
25. XDRACO coin [Electronic resource] // xdraco. – Mode of access: https://www.xdraco.com/coin/price.
26. OpenCV: Перекладен українською [Electronic resource] // Документація бібліотеки OpenCv. – Mode of access: https://ac2epsilon.github.io/TRANS/OPENCV/OpenCVStructures.html.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Магістерська робота

«Розробка методики підтримки ігрового процесу блокчейн ігор на основі обробки зображень»

Виконав: студент групи ПДМ-64 Коновал Микита Михайлович

Керівник: к.т.н., доц., доцент кафедри ІПЗ Золотухіна Оксана Анатоліївна

Київ - 2024

МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: підвищення ефективності видобутку криптовалюти в комп'ютерній блокчейн-грі на основі методів обробки зображень.

Об'єкт дослідження: підтримка ігрового процесу блокчейн ігор.

Предмет дослідження: методи та засоби підтримки ігрового процесу блокчейн ігор на основі обробки зображень.

МЕТОДИ ОБРОБКИ ЗОБРАЖЕНЬ ДЛЯ ПІДТРИМКИ ІГРОВОГО ПРОЦЕСУ БЛОКЧЕЙН ІГОР

Метод	Особливості застосування
Гістограма орієнтованих градієнтів	Обробка зображень на основі опису форми та текстури об'єктів.
Відповідність шаблонів	Чутливість методики до розмірів і освітлення об'єктів.

4

КРИПТОВАЛЮТИ ТА БЛОКЧЕЙН ІГРИ

Вартість популярних криптовалют за даними Мінфіну на 01.12.2023

Криптовалюта	Вартість за одиницю
Bitcoin / BTC	37,325,00 $
Ethereum / ETH	2,040,10 $
HEX / HEX	0,997698 $
Tehther / USDT	1,001 $
DRONE / DRN	0.6591 $

Блокчейн ігри:

- CryptoKitties
- Ni no Kuni: Cross Worlds
- MIR4
- MIR M: Vanguard and Vagabond

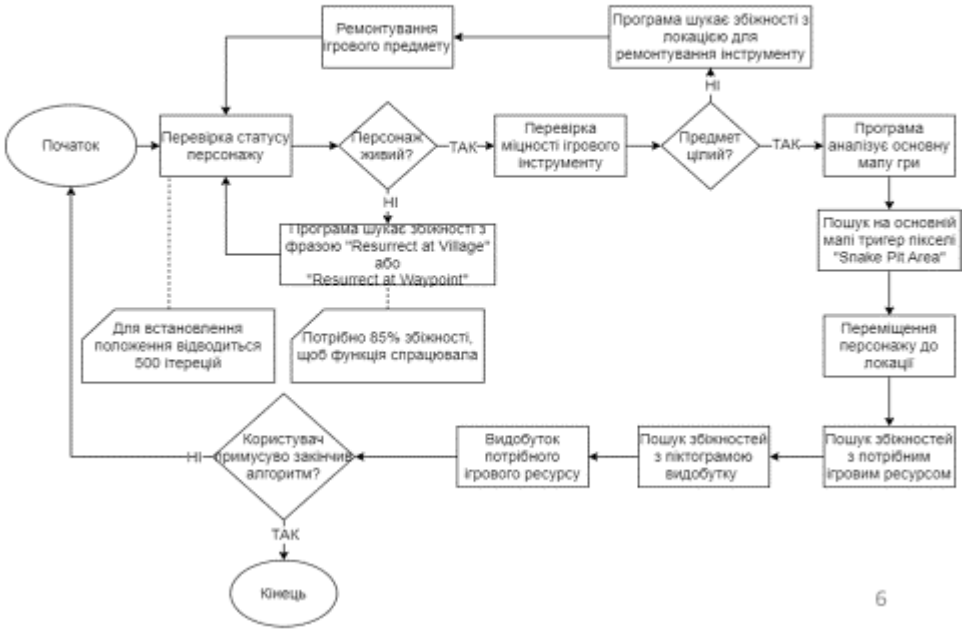
3

ВИДИ ЗОБРАЖЕНЬ ДЛЯ ОБРОБКИ

Назва	Опис
Скріншот піктограми “локація руди”	Робиться для того, щоб встановити місце видобутку руди
Скріншот піктограми “руда”	Робиться для видобутку руди
Скріншот піктограми “ремонтування”	Робиться для того, щоб методика працювала довгий проміжок часу
Скріншот піктограми “відродження персонажу”	Робиться для перевірки статусу персонажу
Скріншот піктограми “видобуток руди”	Робиться для того, щоб персонаж почав видобутку руди

5

БЛОК-СХЕМА ЦИКЛУ ВИДОБУТКУ ІГРОВОГО РЕСУРСУ



6

МАТЕМАТИЧНА МОДЕЛЬ МЕТОДУ ОБРОБКИ ЗОБРАЖЕНЬ TM_CCOEFF_NORMED

Розрахунок математичної операції кореляції між вихідним зображенням I та шаблоном T:

$$R(u,v) = \sum_{x=0}^{w-1}\sum_{y=0}^{h-1} I(x+u, y+v) * T(x,y) \,. (1)$$

Де,

I – вихідне зображення,

T – шаблон порівняння,

$R(u,v)$ – представляє значення кореляції між шаблоном T та пікселями вихідного зображення I.

7

Метод обробки зображень на прикладі "Ремонтування інструменту"

Вихідний скріншот – W*H.

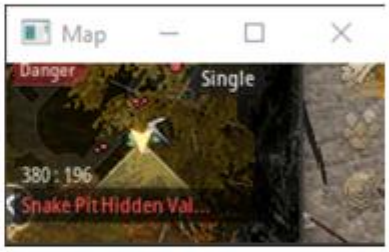
Шаблон скріншот – w*h.

Піктограма, яку ми шукаємо

8

ПРАКТИЧНА РЕАЛІЗАЦІЯ

Екранна форма головного екрану гри



Екранна форма міні-мапи гри

9

РЕЗУЛЬТАТИ ВПРОВАДЖЕННЯ МЕТОДИКИ

Параметри експерименту

Показник	Значення
Час на видобуток однієї жили	15 хв.
Кількість руди за одну жилу	1300 одиниць
Кількість руди, потрібної для обміну на NFT предмет (невзаємозамінний токен)	110000 одиниць
Конвертація невзаємозамінних токенів у криптовалюту DRONE	1 раз на добу
Середня вартість токену криптовалюти DRONE станом на грудень 2023 р.	24 грн.

Результати експерименту

Критерій	Ручний видобуток	Автоматизований видобуток
Час роботи на добу	10 годин	24 години
Час простою	14 годин	Простою немає
Результат видобутку за добу	36400 одиниць	124800 одиниць
Конвертація невзаємозамінних токенів у криптовалюту DRONE	За добу не накопичується потрібна кількість	1 токен
Накопичена сума за одну добу, яку можна обміняти	-	24 грн.
Накопичена сума за місяць за вирахуванням витрат на електроенергію	133,44 грн.*	263,81 грн. + 4 токени в пасивному режимі накопичення

* Розрахунки проведені за умови 5 денного робочого тижня, 22 робочих дні на місяць.

10

ВИСНОВКИ

1. Проаналізовано предметну область блокчейн ігор, особливості ігрового процесу та методи і засоби підтримки ігрового процесу в такому класі ігор. Визначено види зображень для обробки, які застосовано при розробці методики підтримки ігрового процесу блокчейн ігор.
2. Розглянуто методи обробки зображень, які можуть бути використані для підтримки ігрового процесу в блокчейн іграх, та виконано їх порівняльний аналіз. Для побудови методики використано метод зіставлення шаблонів.
3. Розроблено алгоритм обробки зображень блокчейн гри, в якому можна зазначити такі особливості роботи: автоматизація процесів, циклічність роботи, оптимізація часу, саме ці процеси відображає розроблений алгоритм.
4. Для перевірки працездатності методики розроблено додаток, який реалізує методику підтримки ігрового процесу в комп'ютерній грі MIR M: Vanguard and Vagabond та надає можливість автоматизованого видобутку криптовалюти.
5. Проведено аналіз результатів впровадження методики. Використання методики підтримки ігрового процесу блокчейн ігор на основі обробки зображень дозволило збільшити ефективність видобутку криптовалюти наступним чином: автоматизовані засоби забезпечують цілодобовий видобуток криптовалюти без простоїв; у порівнянні з ручним процесом час видобутку на добу та кількість видобутих одиниць збільшились в 2,4 рази. При цьому за місяць накопичується до 4 повних токенів в пасивному режимі, які можна обміняти в майбутньому без застосування видобутку в ці дні. З урахуванням витрат на електроенергію із застосуванням розробленої методики можна очікувати зростання прибутку в грошовому еквіваленті в 1,98 рази більше, ніж при ручному видобутку при одноекранному використанні методики і в 3,96 разів більше при двоекранному режимі.

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Коновал М.М., Золотухіна О.А. "Розробка методики підтримки ігрового процесу блокчейн ігор на основі обробки зображень" // Науково-практична конференція «Проблеми комп'ютерної інженерії», 10 грудня 2023 року, Державний університет інформаційно-комунікаційних технологій, Київ, Україна, С. 137.

ДЯКУЮ ЗА УВАГУ!

ДОДАТОК А
ПРОГРАМНИЙ КОД

```
import cv2
import numpy
import mss
import keyboard
import pyautogui
from os import listdir
from time import time, sleep
"""
Only gathering version 1
"""
repairIcon =  cv2.imread('imgs2wind\\repairLogo.png', cv2.IMREAD_UNCHANGED)
locationIcon =  cv2.imread('imgs2wind\\locationLogo.png', cv2.IMREAD_UNCHANGED)
miningIcon = cv2.imread('imgs2wind\\miningLogo.png', cv2.IMREAD_UNCHANGED)
waystoneIcon = cv2.imread('imgs2wind\\waystoneIcon.png', cv2.IMREAD_UNCHANGED)
snakepitArea = cv2.imread('imgs2wind\\snakePitArea1.png', cv2.IMREAD_UNCHANGED)
textIcon = cv2.imread('imgs2wind\\textLogo.png', cv2.IMREAD_UNCHANGED)
resIcon = cv2.imread('imgs2wind\\resLogo.png', cv2.IMREAD_UNCHANGED)
minedStoneIcon = cv2.imread('imgs2wind\\minedStoneLogo.png', cv2.IMREAD_UNCHANGED)
minedStoneIcon2 = cv2.imread('imgs2wind\\minedStoneLogo2.png', cv2.IMREAD_UNCHANGED)
minedStoneIcon3 = cv2.imread('imgs2wind\\minedStoneLogo3.png', cv2.IMREAD_UNCHANGED)
minedStoneIcon4 = cv2.imread('imgs2wind\\minedStoneLogo4.png', cv2.IMREAD_UNCHANGED)
minedStoneIcon5 = cv2.imread('imgs2wind\\minedStoneLogo6.png', cv2.IMREAD_UNCHANGED)

rightUpperAccount = False
leftUpperAccount = True


pyautogui.PAUSE = 0

print("Press 's' to start playing.")
keyboard.wait('s')
sct = mss.mss()

leftUpperQuarter = {
        'left': 1,
        'top': 1,
        'width': 960,
        'height': 520
    }
leftUpperQuarterMap = {
        'left': 800,
        'top': 100,
        'width': 200,
        'height': 95
    }
rightUpperQuarterMap = {
        'left': 800+960,
        'top': 100,
        'width': 200,
        'height': 95
    }
fullScreen = {
        'left': 1,
        'top': 1,
        'width': 1919,
        'height': 1079
    }
rightUpperQuarter = {
        'left': 960,
        'top': 1,
        'width': 960,
```

```
        'height': 520
    }
leftUpperQuarterFull = {
        'left': 1,
        'top': 1,
        'width': 980,
        'height': 520
    }
leftDownQuarter = {
        'left': 0,
        'top': 520,
        'width': 980,
        'height': 520
    }
rightDownQuarter = {
        'left': 960,
        'top': 520,
        'width': 980,
        'height': 520
    }



def placementFix():
    while True:
        fullScrn = numpy.array(sct.grab(fullScreen))
        minedStoneCheck = cv2.matchTemplate(fullScrn, minedStoneIcon, cv2.TM_CCOEFF_NORMED)
        _, minedmax1_val, _, minedmax1_loc = cv2.minMaxLoc(minedStoneCheck)
        minedStoneCheck2 = cv2.matchTemplate(fullScrn, minedStoneIcon2, cv2.TM_CCOEFF_NORMED)
        _, minedmax2_val, _, minedmax2_loc = cv2.minMaxLoc(minedStoneCheck2)
        minedStoneCheck3 = cv2.matchTemplate(fullScrn, minedStoneIcon3, cv2.TM_CCOEFF_NORMED)
        _, minedmax3_val, _, minedmax3_loc = cv2.minMaxLoc(minedStoneCheck3)
        minedStoneCheck4 = cv2.matchTemplate(fullScrn, minedStoneIcon4, cv2.TM_CCOEFF_NORMED)
        _, minedmax4_val, _, minedmax4_loc = cv2.minMaxLoc(minedStoneCheck4)
        minedStoneCheck5 = cv2.matchTemplate(fullScrn, minedStoneIcon5, cv2.TM_CCOEFF_NORMED)
        _, minedmax5_val, _, minedmax5_loc = cv2.minMaxLoc(minedStoneCheck5)
        minedArray = numpy.array([minedmax1_val, minedmax2_val, minedmax3_val, minedmax4_val, minedmax5_val])
            locsArray = numpy.array([minedmax1_loc, minedmax2_loc, minedmax3_loc, minedmax4_loc, minedmax5_loc])
            indexMax = numpy.where(minedArray == numpy.amax(minedArray))
            if numpy.amax(minedArray) > .75:
                _x = locsArray[indexMax[0][0]][0]
                _y = locsArray[indexMax[0][0]][1]
                pyautogui.click(x=_x, y=_y)
                sleep(1)
                break


    def locationFix(offsetX = 0, offsetY = 0, account = 0):
        tries = 0
        pyautogui.click(x=875 + offsetX, y=150 + offsetY)
        sleep(3)
        pyautogui.click(x=50 + offsetX, y=80 + offsetY)
        sleep(3)
        pyautogui.click(x=43 + offsetX, y=350 + offsetY)
        sleep(3)
        while True:
            tries += 1
            if account == 0:
```

```
121.                 fullScr = numpy.array(sct.grab(leftUpperQuarter))
122.             elif account == 1:
123.                 fullScr = numpy.array(sct.grab(rightUpperQuarter))
124.             spCheck = cv2.matchTemplate(fullScr, snakepitArea, cv2.TM_CCOEFF_NORMED)
125.             _, spmax_val, _, spmax_loc = cv2.minMaxLoc(spCheck)
126.             print(f'spmax {spmax_val}')
127.             if spmax_val >= .8:
128.                 pyautogui.click(x=spmax_loc[0]+65, y=spmax_loc[1]+10)
129.                 sleep(3)
130.                 break
131.             if tries >= 500:
132.                 break
133.         pyautogui.click(x=550 + offsetX, y=270 + offsetY)
134.         sleep(8)
135.         pyautogui.click(x=850 + offsetX, y=80 + offsetY)
136.         sleep(2)
137.         pyautogui.click(x=830 + offsetX, y=120 + offsetY) #tp target
138.         sleep(2)
139.         pyautogui.click(x=850 + offsetX, y=320 + offsetY)
140.         sleep(2)
141.         pyautogui.click(x=520 + offsetX, y=350 + offsetY)
142.         sleep(25)
143.
144.     def repair(offsetX = 0, offsetY = 0):
145.         pyautogui.click(x=870 + offsetX, y=45 + offsetY)
146.         sleep(3)
147.         pyautogui.click(x=810 + offsetX, y=430 + offsetY)
148.         sleep(3)
149.         pyautogui.click(x=520 + offsetX, y=320 + offsetY)
150.         sleep(3)
151.         pyautogui.click(x=930 + offsetX, y=85 + offsetY)
152.         sleep(170)
153.         pyautogui.click(x=810 + offsetX, y=330 + offsetY)#repairGear
154.         sleep(3)
155.         pyautogui.click(x=120 + offsetX, y=460 + offsetY)
156.         sleep(3)
157.         pyautogui.click(x=530 + offsetX, y=430 + offsetY)
158.         sleep(3)
159.         pyautogui.click(x=935 + offsetX, y=45 + offsetY)
160.         sleep(3)
161.         pyautogui.click(x=890 + offsetX, y=490 + offsetY) #close
162.         sleep(3)
163.
164.     def mine(offsetX = 0, offsetY = 0, state = 0):
165.         pyautogui.click(x=920 + offsetX, y=310 + offsetY)
166.         sleep(1)
167.         if state == 1:
168.             pyautogui.click(x=815 + offsetX, y=305 + offsetY)
169.         else:
170.             pyautogui.click(x=845 + offsetX, y=305 + offsetY)
171.         sleep(1)
172.
173.
174.
175.
176.
177.     while True:
178.
179.         charCam = numpy.array(sct.grab(leftUpperQuarter))
180.         mapCam = numpy.array(sct.grab(leftUpperQuarterMap))
181.         fullScr = numpy.array(sct.grab(fullScreen))
182.
183.         charCam2 = numpy.array(sct.grab(rightUpperQuarter))
184.         mapCam2 = numpy.array(sct.grab(rightUpperQuarterMap))
185.
186.         deathCheck = cv2.matchTemplate(fullScr, resIcon, cv2.TM_CCOEFF_NORMED)
```

```
187.          _, deathmax_val, _, deathmax_loc = cv2.minMaxLoc(deathCheck)
188.
189.          if leftUpperAccount:
190.              repairCheck = cv2.matchTemplate(charCam, repairIcon,
    cv2.TM_CCOEFF_NORMED)
191.              _, rmax_val, _, rmax_loc = cv2.minMaxLoc(repairCheck)
192.              locCheck = cv2.matchTemplate(mapCam, locationIcon, cv2.TM_CCOEFF_NORMED)
193.              _, locmax_val, _, locmax_loc = cv2.minMaxLoc(locCheck)
194.              mining = cv2.matchTemplate(charCam, miningIcon, cv2.TM_CCOEFF_NORMED)
195.              _, minemax_val, _, minemax_loc = cv2.minMaxLoc(mining)
196.
197.          if rightUpperAccount:
198.              repairCheck2 = cv2.matchTemplate(charCam2, repairIcon,
    cv2.TM_CCOEFF_NORMED)
199.              _, rmax_val2, _, rmax_loc2 = cv2.minMaxLoc(repairCheck2)
200.              locCheck2 = cv2.matchTemplate(mapCam2, locationIcon,
    cv2.TM_CCOEFF_NORMED)
201.              _, locmax_val2, _, locmax_loc2 = cv2.minMaxLoc(locCheck2)
202.              mining2 = cv2.matchTemplate(charCam2, miningIcon, cv2.TM_CCOEFF_NORMED)
203.              _, minemax_val2, _, minemax_loc2 = cv2.minMaxLoc(mining2)
204.
205.          if keyboard.is_pressed('F5'):
206.              locationFix()
207.          elif keyboard.is_pressed('F6'):
208.              placementFix()
209.          elif keyboard.is_pressed('F7'):
210.              repair()
211.          #cv2.rectangle(charCam, minemax_loc, (minemax_loc[0] + miningIcon.shape[1],
    minemax_loc[1] + miningIcon.shape[0]),(1,255,255))
212.
213.          if deathmax_val> .85 or keyboard.is_pressed('F8'):
214.              pyautogui.screenshot(f'deaths\\{len(listdir("deaths"))}.png')
215.              print('making scrs')
216.              sleep(4)
217.              pyautogui.click(x=deathmax_loc[0]+10, y=deathmax_loc[1]+10)
218.              sleep(10)
219.              continue
220.
221.          # if leftUpperAccount:
222.          #     if locmax_val > .75:
223.          #         onLocation = True
224.          #     else:
225.          #         onLocation = False
226.          #     if onLocation and rmax_val < .8:
227.          #         print('Idle leftUpper')
228.          #         sleep(.1)
229.          #     elif rmax_val > .8:
230.          #         print('Repair leftUpper')
231.          #         repair()
232.          #     elif not onLocation:
233.          #         locationFix()
234.          #         mine()
235.
236.          if rightUpperAccount:
237.              if locmax_val2 > .75:
238.                  onLocation2 = True
239.              else:
240.                  onLocation2 = False
241.              if onLocation2 and rmax_val2 < .8:
242.                  print('Idle rightUpper')
243.                  sleep(.1)
244.              elif rmax_val2 > .8:
245.                  print('Repair rightUpper')
246.                  repair(960,0)
247.              elif not onLocation2:
248.                  locationFix(960,0,1)
```

```
                mine(960,0)


        cv2.imshow('Char', charCam)
        cv2.imshow('Map ', mapCam)
        if rightUpperAccount:
            cv2.imshow('Char2', charCam2)
            cv2.imshow('Map2 ', mapCam2)

        cv2.waitKey(1)
        sleep(.10)
        if keyboard.is_pressed('F4'):
            break
```