

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр
(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
АВТОМАТИЗОВАНОЇ СИСТЕМИ ФІКСУВАННЯ РЕЗУЛЬТАТІВ
КІБЕРСПОРТИВНИХ ЗМАГАНЬ
RESEARCH AND SOFTWARE IMPLEMENTATION OF AUTOMATED
SYSTEM FOR RECORDING THE RESULTS OF CYBER SPORT
COMPETITIONS

Виконав: студент(ка) 2 курсу, групи КНТ-129м
Спеціальності 121 Інженерія програмного
забезпечення
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Зубко Е. О.
(прізвище та ініціали)

Керівник Колпакова Т.О.
(прізвище та ініціали)

Рецензент Казурова А.Є.
(прізвище та ініціали)

1. Тема проекту (роботи) Дослідження та програмна реалізація автоматизованої системи фіксування результатів кіберспортивних змагань.
Research and Software Implementation of Automated System for Recording the Results of Cyber Sport Competitions.
- керівник проекту (роботи) Колпакова Тетяна Олексіївна, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
- затверджені наказом закладу вищої освіти від “03” листопада 2020 року № 301
2. Строк подання студентом проекту (роботи) 10 грудня 2020 року
3. Вихідні дані до проекту (роботи) рекомендована література, технічне завдання.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Дослідження предметної області та постановка задачі дослідження.
2. Аналіз технологій та бібліотек. 3. Основні рішення з реалізації системи. 4.
Керівництво програміста. 5. Керівництво користувача. 6. Організаційно-
економічна частина. 7. Охорона праці та безпека у надзвичайних ситуаціях.
8. Додатки.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Колпакова Т. О., доцент		
Організаційно-економічна частина	Пожуєва Т.О., професор		
Охорона праці та цивільна безпека	Коробко О.В., ст. викладач		
Нормоконтролер	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “4” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання
2	Аналіз предметної області.	2-3 тижні	Розділ 1
3	Розробка та удосконалення методів й алгоритмів вирішення задачі.	4-5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7-8 тижні	Розділ 4
6	Тестування програми користувачами.	9 тижні	Розділ 5
7	Розробка організаційно-економічної частини	10 тиждень	Розділ 6
8	Розробка заходів з охорони праці та цивільної безпеки.	11 тиждень	Розділ 7
9	Оформлення пояснювальної записки та документів до неї.	12-13 тиждень	ПЗ, Додатки
10	Нормоконтроль та рецензування.	14-15 тижні	
11	Захист роботи.	16 тиждень	

Студент(ка) _____ Зубко Е. О.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи) _____ Колпакова Т. О.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до атестаційної роботи магістра: 109 с., 11 табл., 31 рис., 3 дод., 31 джерело.

РОЗПІЗНАВАННЯ ТЕКСТУ, PYTHON, TESSERACT, OCR, DISCORD, PIL, OPENCV.

Об'єкт дослідження – процес розпізнавання тексту з зображення для фіксування результатів кіберспортивних змагань.

Предмет дослідження – модель для автоматизованого розпізнавання текстових результатів змагання з зображення.

Мета роботи – дослідити та створити модель для розпізнавання тексту з зображення для використання їх в автоматизованій системі фіксування результатів кіберспортивних змагань.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, використання бібліотеки для роботи по розпізнаванню тексту з зображення Tesseract, OpenCV, графічна бібліотека Pillow, бібліотека discord.py, мова програмування Python 3.8, середовище розробки програм PyCharm, персональний комп'ютер з процесором Intel Core i9, оперативною пам'яттю 16 Гб, 32" монітором, клавіатурою та маніпулятором типу «миша», що функціонує під управлінням операційної системи Windows 10.

Результат. Створено систему, яка дозволяє спостерігати за онлайн змаганнями та отримувати результати за допомогою автоматизованого розпізнавання тексту з зображення.

Висновки. Розроблена система для проведення онлайн змагань, яка дозволяє покращити їх зручність та швидкість у отриманні результатів учасників.

Галузь використання – будь-які кіберспортивні організації чи організатори кіберспортивних змагань.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 109 pages, 11 tables, 31 figures, 3 appendixes, 31 sources.

TEXT RECOGNITION, PYTHON, TESSERACT, OCR, DISCORD, PIL, OPENCV.

The object of research is the process of recognizing text from an image to record the results of e-sports competitions.

The subject of research is a model for automated recognition of text results of the image competition.

The purpose of the work is to research and create a model for text recognition from images for use in an automated system for recording the results of e-sports competitions.

Materials, methods and technical means: structural and object-oriented programming, use of the library for work on text recognition from the image Tesseract, OpenCV, graphic library Pillow, library for work with the messenger Discord discord.py, Python 3.8 programming language, development environment PyCharm software, a personal computer with an Intel Core i9 processor, 16 GB of RAM, 32" monitor, keyboard and mouse-type manipulator running Windows 10.

Result. A system has been created that allows you to watch online competitions and get results using automated text recognition from images.

Conclusions. A system for conducting online competitions has been developed, which allows to improve their convenience and speed in obtaining the results of participants.

Field of use – any e-sports organizations or organizers of e-sports competitions.

ЗМІСТ

	С.
Перелік умовних позначень і скорочень	9
Вступ.....	10
1 Дослідження предметної області та постановка задачі дослідження.....	11
1.1 Характеристика предметної області	11
1.2 Огляд існуючих методів вирішення задачі	12
1.2.1 Toornament.com	12
1.2.2 Weplay.tv	13
1.2.3 Game.tv	14
1.2.4 Challonge.com	16
1.2.5 Cybernetica.agency	17
1.3 Висновки за розділом 1	18
2 АНАЛІЗ ТЕХНОЛОГІЙ ТА бібліотек	19
2.1 Аналіз бібліотек розпізнавання	19
2.1.1 Tesseract	19
2.1.2 OpenCV	21
2.1.3 Anyline.....	23
2.1.4 AForge.Net	24
2.1.5 VXL.....	25
2.1.6 LTI.....	26
2.1.7 Google Text Recognition API	27
2.2 Технології розробки.....	27
2.3 Висновки за розділом 2	30
3 ОСНОВНІ РІШЕННЯ З РЕАЛІЗАЦІЇ СИСТЕМИ	31

3.1 Особливості системи	31
3.1.1 Логічна структура системи	31
3.2 Особливості мови програмування Python	33
3.3 Особливості використання бібліотеки ІО та функції BytesІО	34
3.4 Особливості використання бібліотеки OpenCV та дослідження її ефективності.....	35
3.5 Особливості використання бібліотеки Tesseract.....	40
3.6 Особливості використання бібліотеки PyParsing	41
3.7 Схема бази даних	42
3.8 Frontend	43
3.9 Висновки за розділом 3	44
4 Керівництво програміста	45
4.1 Призначення системи	45
4.2 Умови експлуатації програми.....	45
4.3 Характеристика програми	45
4.4 Звернення до системи	46
4.5 Вхідні і вихідні дані	46
4.6 Повідомлення	46
5 Керівництво користувача	47
5.1 Робота з системою.....	47
5.2 Повідомлення користувачу	52
6 Економіко-організаційна частина	53
6.1 Планування розробки програмного виробу	53
6.2 Визначення витрат на розробку програми	56
6.2.1 Розрахунок основної заробітної плати	56

6.2.2	Розрахунок додаткової заробітної плати.....	57
6.2.3	Відрахування на єдиний соціальний внесок	57
6.2.4	Визначення витрат на матеріали	58
6.2.5	Витрати на спеціальне устаткування.....	59
6.2.6	Накладні витрати	62
6.3	Оцінка економічної ефективності	63
7	Охорона праці та безпека в надзвичайних ситуаціях.....	64
7.1	Аналіз потенціальних небезпек	64
7.2	Заходи по забезпеченню безпеки	66
7.3	Заходи з забезпечення виробничої санітарії та гігієни праці	72
7.4	Заходи безпеки у надзвичайних ситуаціях	73
7.4.1	Заходи з пожежної безпеки.....	73
7.4.2	Заходи з цивільної оборони	77
	ВИСНОВКИ.....	81
	Перелік джерел посилання	82
	Додаток А Технічне завдання	85
	Додаток Б Текст програми	90
	Додаток В Слайди презентації.....	101

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

API	– Application Programming Interface, прикладний програмний інтерфейс;
CNN	– Convolutional Neural Network, рекурентна нейронна мережа;
CRM	– Customer Relationship Management, управління відносинами з клієнтами;
CV	– Computer Vision, комп'ютерний зір;
GDPR	– General Data Protection Regulation, загальний регламент про захист даних;
LSTM	– Long Short-Term Memory, довга короткострокова пам'ять;
OCR	– Optical Character Recognition, оптичне розпізнавання символів;
PIL	– Python Imaging Library, бібліотека зображень Python;
RNN	– Recurrent Neural Network, періодична нейронна мережа;
ГІК	– графічний інтерфейс користувача.

ВСТУП

На сьогоднішній день створення власної системи фіксування результатів кіберспортивних змагань є актуальним питанням для будь-якої гри чи кіберспортивної організації, особливо якщо вона матиме автоматизовану систему отримання та зберігання результатів. У світі на даний час існує понад десять тисяч кіберспортивних команд, близько тисячі кіберспортивних дисциплін та велика кількість проектів, які тільки розвиваються та не мають власної вбудованої системи для проведення турнірів. Тому організаціям, які бажають проводити турніри в таких проектах, вкрай необхідно мати власну систему.

Така система дозволить не тільки проводити турніри, а ще автоматично фіксувати результати з зображень учасників турніру та зберігати їх до бази даних. Також система може мати додаткові можливості, наприклад, відображення статистики на веб-сайті з автоматичним оновленням актуальної інформації. У світі існує велика кількість автоматичних систем для проведення кіберспортивних змагань, та їх кількість постійно зростає, бо з'являється все більше нових ігор, які позиціонують себе як киберспортівні дисципліни.

Таким чином питання розробки своєї автоматизованої системи з використанням сучасних технологій фіксації результатів є актуальною проблемою. В рамках дипломної роботи було поставлено завдання розробки такої автоматизованої системи.

Розробку подібних систем можна виконувати як за допомогою вже готових доступних рішень, з відкритим вихідним кодом або платних, також існують вже готові веб-сервіси для фіксування, але вони часто мають обмежений функціонал та платні. В роботі було проаналізовано обидва підходи для виявлення їх переваг та недоліків.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Характеристика предметної області

Автоматизована система фіксування результатів – це система, яка дозволяє фіксувати результати отримані певним чином для їх збереження та обробки. Вона може виглядати як веб-сайт, веб-сервіс чи будь-який скрипт, який має мінімальний інтерфейс. В нашій системі буде використовуватися автоматизована система отримання результатів з зображення, які будуть відправляти учасники через месенджер Discord. Основними відмінностями автоматизованої системи від звичайної є швидкість в отриманні результатів, учасником не потрібно вручну заносити дані, а можна швидко зробити скріншот та відправити його до системи для фіксації результатів.

Основними функціями системи є:

- надання повної інформації результатів турніру;
- прийом та обробка зображення з результатами;
- персоналізація результатів;
- проведення турнірів;
- збір та аналіз інформації.

Система повинна дозволяти в будь-який час учаснику турніру отримати актуальну інформацію о його результатах на турнірі, а також турнірну таблицю по всім учасникам турніру.

Система також включає всі функції для вирішення повного комплексу задач, які необхідні для проведення турніру, а саме:

- реєстрація учасників на турнір;
- створення захищених лоббі для учасників;
- видача ролей учасників турніру;
- оповіщення про дату та час початка турніру.

1.2 Огляд існуючих методів вирішення задачі

В наш час є велика кількість готових рішень для проведення онлайн турнірів та фіксації їх результатів, але не завжди їх використання є вдалим вибором, тому що готові системи не завжди мають потрібний функціонал або в них немає підтримки певних дисциплін.

1.2.1 Toornament.com

Toornament.com – англomовний умовно-безкоштовний веб-сервіс для проведення кіберспортивних змагань [1].

Основні можливості:

- створювати турніру за певною кіберспортивною дисципліною;
- повноцінна інформаційна панель, яка дає повний контроль над робочим процесом турніру;
- інтеграція сервісів сайту завдяки REST API;
- сервіс пропонує сучасні формати турнірів;
- підтримка Battle Royale та Sim-Racing з кількістю учасників до 100 гравців;
- сервіс відповідає GDPR;
- легкий експорт даних гравців та результатів турніру;
- гарна сторінка турніру, яка має адаптивний дизайн для роботи з мобільних пристроїв.

На рисунку 1.1 відображено головну сторінку сервісу Toornament.com.

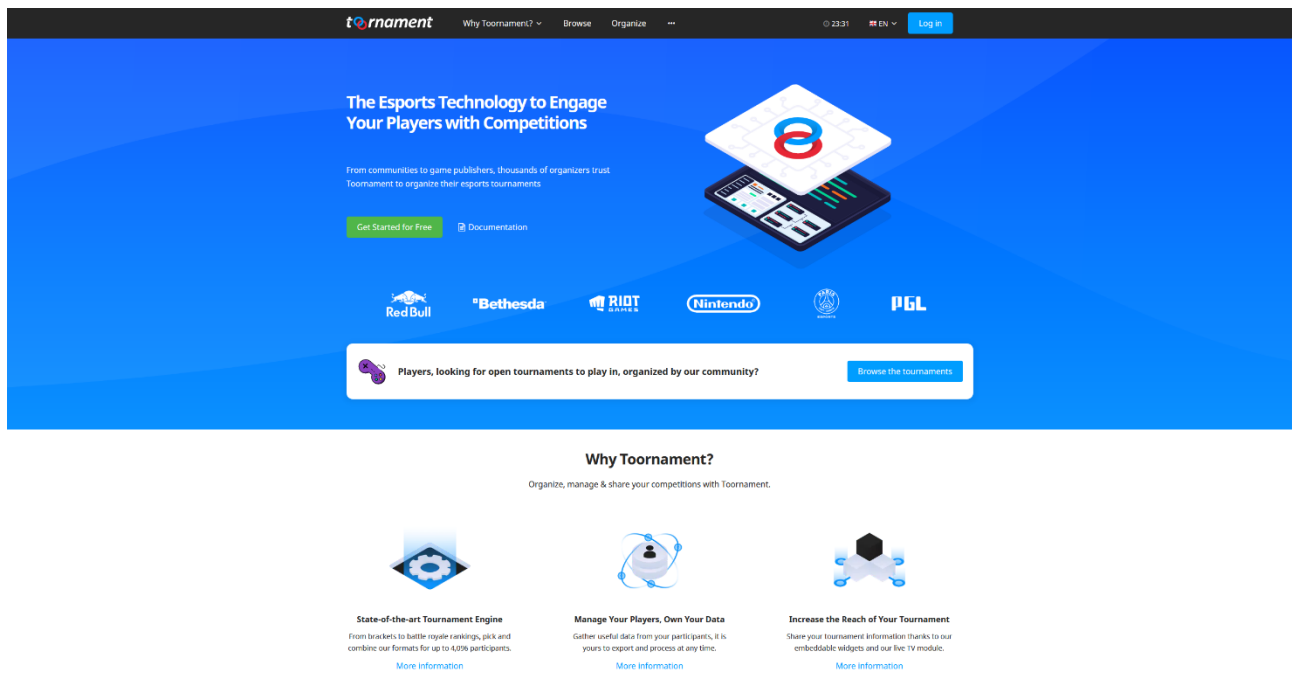


Рисунок 1.1 – Вигляд головної сторінки сервісу Toornament.com

Недоліки сервісу Toornament.com:

- немає автоматичної фіксації результатів;
- не підтримує необхідну ігрову дисципліну;
- неможлива робота сервісу через месенджер Discord;
- необхідні грошові вкладення за роботу сервісу;
- неможливо змінити функціонал;
- немає української/російської мови інтерфейсу.

1.2.2 Weplay.tv

Weplay.tv – веб-сервіс, турнірна платформа для проведення турнірів та пошуку кіберспортивних команд [2].

Основні можливості:

- швидкий пошук актуальних турнірів;
- повна інтеграція сервісу з месенджером Discord;
- пошук команди суперників або суперника для solo-турнірів;

– можливість створювати свій турнір з трьох кіберспортивних дисциплін;

– чітка організація структури турніру та його позиціонування.

На рисунку 1.2 відображено головну сторінку веб-сервісу weplay.tv.

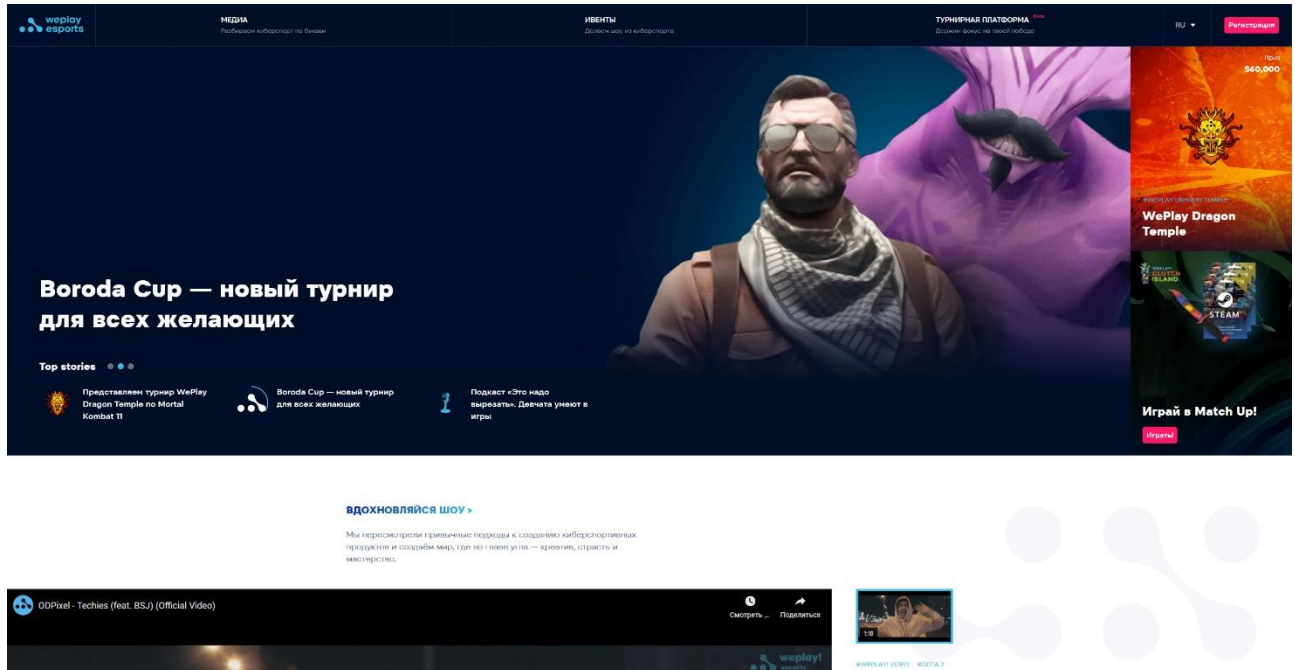


Рисунок 1.2 – Вигляд головної сторінки веб-сервісу weplay.tv

Недоліки:

- обмежена кількість кіберспортивних дисциплін;
- немає автоматичної фіксації результатів;
- складна процедура створення власного турніру;
- обмежений функціонал;
- немає доступу до повної статистики.

1.2.3 Game.tv

Game.tv – веб-сервіс з повною інтеграцією з месенджером Discord та власним штучним інтелектом для автоматичного створення та проведення турнірів [3].

Основні можливості:

- створення турніру по усім популярним кіберспортивним дисциплінам;
- повноцінна інтеграція та управління сервісом через месенджер Discord;
- штучний інтелект, який може автоматизовано проводити турнір;
- можливість створювати турніри по мобільним іграм та управляти ходом турніру з мобільного пристрою;
- гарна статистика та інтеграція з власним додатком завдяки REST API;
- управління турніром на усіх етапах його проведення;
- імпорт і експорт результатів турніру;
- підтримка великої кількості мов для об'єднання міждержавних команд.

На рисунку 1.3 відображено головну сторінку веб-сервісу game.tv.

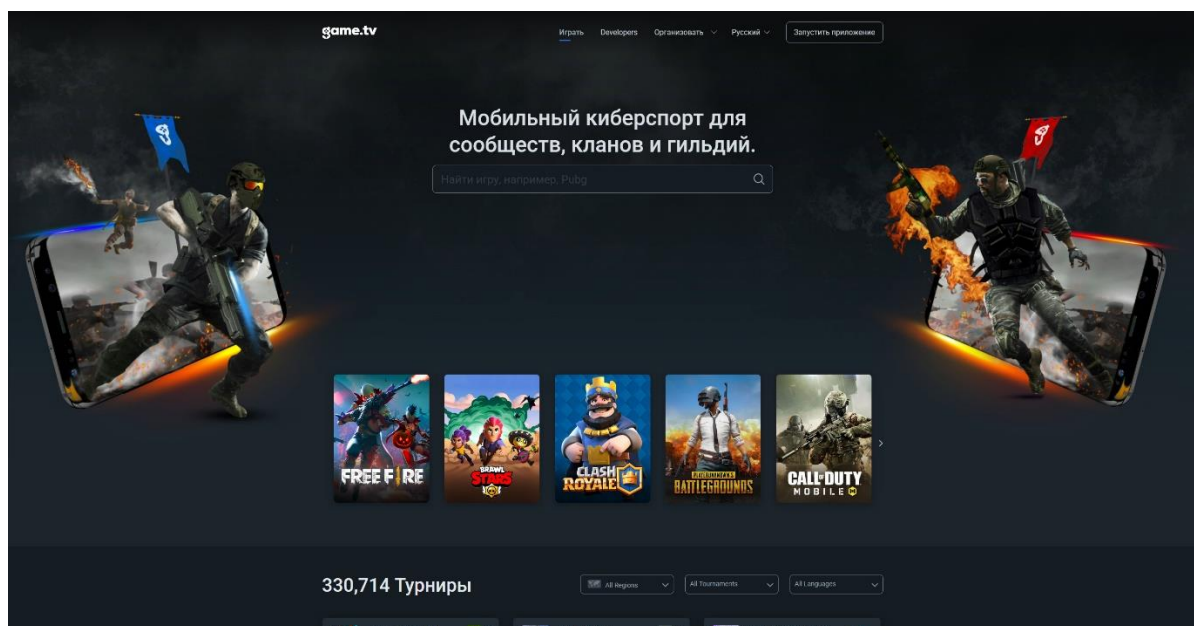


Рисунок 1.3 – Вигляд головної сторінки веб-сервісу game.tv

Недоліки game.tv наступні:

- немає автоматичної фіксації результатів;
- відсутня необхідна кіберспортивна дисципліна;
- використання API платне.

1.2.4 Challonge.com

Challonge.com – веб-сервіс, створений в 2009 році для генерації турнірів з великою кількістю гравців [4].

Основні можливості:

- велика кількість типів систем турніру;
- створення турніру з заздалегідь прописаним призовим фондом;
- система продажу квитків для глядачів на трансляцію турнірів;
- повноцінна панель для відстеження усіх етапів підготовки турніру;
- API, яке дозволяє отримати будь-яку інформацію о турнірі у режимі онлайн;
- немає прив'язки до певної кіберспортивної дисципліни;
- пошук партнерів у спільноті, яка вже є на сайті;
- повідомлення покупців і співробітників відділів;
- гнучка система налаштувань та форматування результатів.

На рисунку 1.4 відображено головну сторінку веб-сервісу challonge.com.

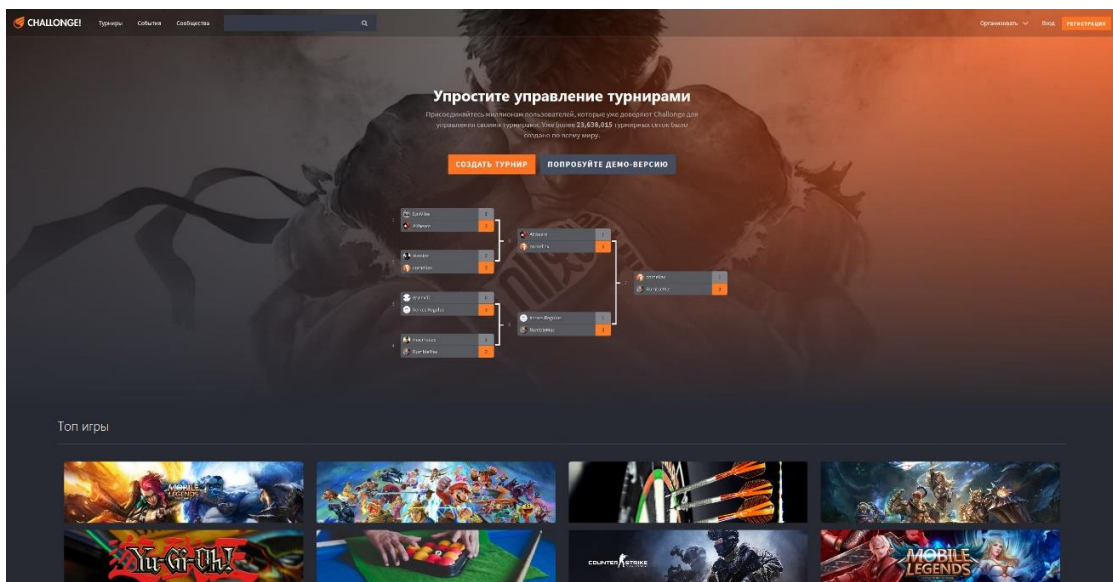


Рисунок 1.4 – Вигляд головної сторінки веб-сервісу challonge.com

Недоліки:

- неповноцінний переклад російською та повна відсутність перекладу українською мовою;
- немає автоматичної фіксації результатів;
- створений турнір важко знайти без прямого посилання;
- немає інтеграції з месенджером Discord;
- для створення більше ніж одного турніру треба платити.

1.2.5 Cybernetica.agency

Cybernetica.agency – веб-сервіс кіберспортивного маркетингового агентства, який займається створенням та проведенням великомасштабних кіберспортивних турнірів [5].

Основні можливості:

- повноцінна розробка усіх етапів турніру з нуля;
- пошук команд та місця проведення турнірів;
- створення повноцінного додатку або веб-сайту для керування турніром та перегляду результатів турніру;
- створення та розвиток медійного поширення турніру для залучення нової аудиторії;
- інтеграція з будь-якою системою управління (CRM).

На рисунку 1.5 відображено головну сторінку веб-сервісу cybernetica.agency.

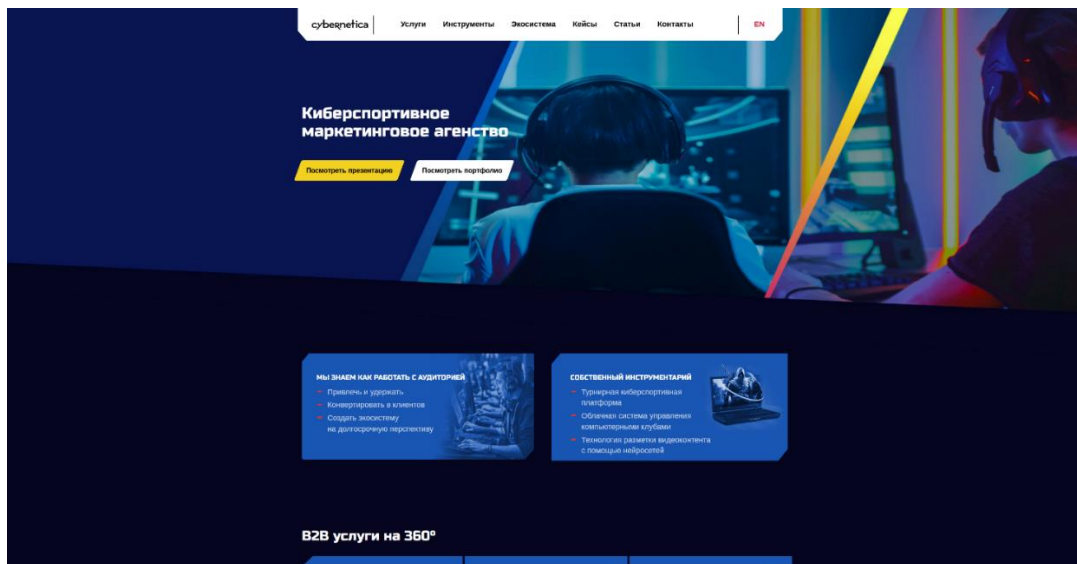


Рисунок 1.5 – Вигляд головної сторінки веб-сервісу cybernetica.agency

Недоліки:

- сервіс підходить тільки для великомасштабних турнірів;
- дуже велика вартість розробки;
- надмірний функціонал, який не завжди потрібен.

1.3 Висновки за розділом 1

На основі проведеного аналізу готових рішень веб-сервісів для проведення кіберспортивних змагань можна зробити висновок, що вони можуть використовуватись для проведення турнірів у певних кіберспортивних дисциплінах.

Головним недоліком таких сервісів є те, що вони не можуть надати функціонал з автоматизованого фіксування результатів, та неможливість додати власний функціонал. Не всі сервіси підтримують інтеграцію з месенджером Discord, який є загальновизнаним інтерфейсом для проведення турнірів з багатьох кібердисциплін. Також веб-сервіси не можна повноцінно використовувати безкоштовно.

Тому кращим рішенням є створення власної системи фіксування результатів кіберспортивних змагань з потрібним функціоналом.

2 АНАЛІЗ ТЕХНОЛОГІЙ ТА БІБЛІОТЕК

2.1 Аналіз бібліотек розпізнавання

На рисунку 2.1 наведено графік швидкодії найбільш популярних з сучасних бібліотек комп'ютерного зору. Всі вони мають різну швидкість для вирішення задачі, тому треба знайти ту бібліотеку, яка буде найкраще справлятися з розпізнаванням результатів [6].

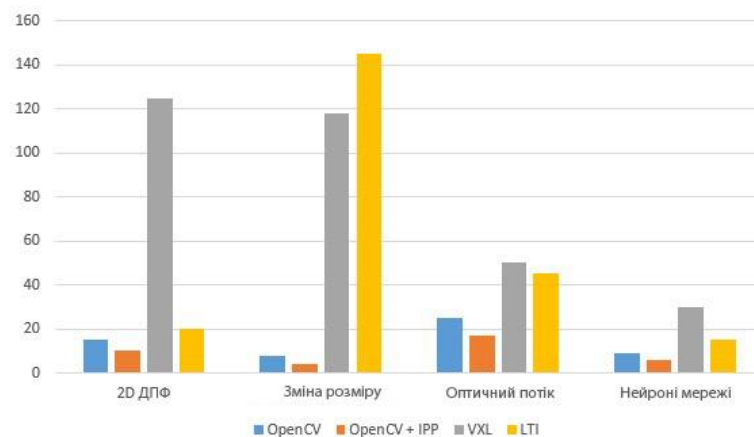


Рисунок 2.1 – Графік порівняння швидкодії бібліотек комп'ютерного зору

2.1.1 Tesseract

Tesseract – це механізм розпізнавання тексту з відкритим кодом, який доступний за ліцензією Apache 2.0, і його розвиток фінансується Google з 2006 року. У 2006 році Tesseract вважався одним з найточніших механізмів OCR з відкритим кодом. Ви можете використовувати його безпосередньо або можете використовувати API для вилучення надрукованого тексту із зображень. Найкраща частина полягає в тому, що вона підтримує широкий спектр мов. Завдяки обгорткам Tesseract можна зробити сумісним з різними мовами програмування та фреймворками. У дипломі я буду використовувати обгортку Python з назвою pytesseract. Він використовується для розпізнавання тексту з великого документа, а також може використовуватися для розпізнавання тексту

із зображення одного текстового рядка. На рисунку 2.2 наведено візуальне зображення архітектури розпізнавання Tesseract, як це представлено у роботі [7].

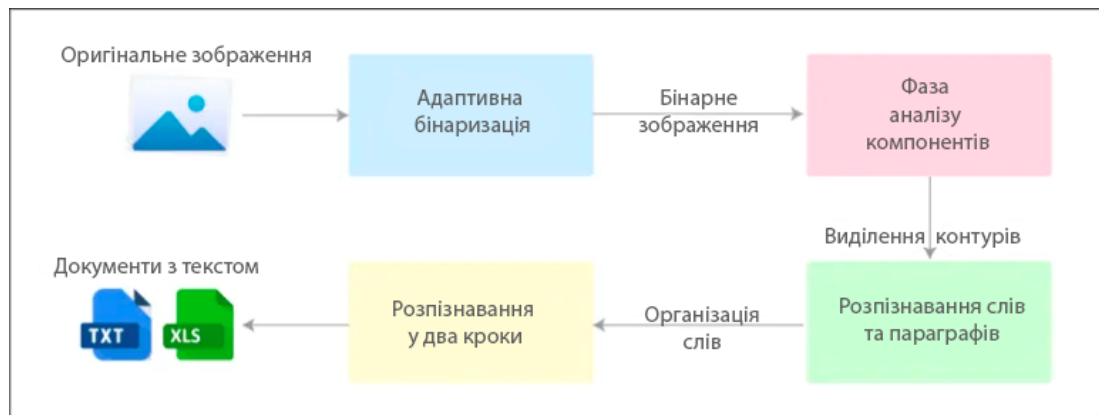


Рисунок 2.2 – Візуальне зображення архітектури розпізнавання Tesseract

Tesseract версії 4.00 має налаштований розпізнавач текстових рядків у новій неймережевій підсистемі. У наші дні люди зазвичай використовують рекурентну нейронну мережу (CNN), щоб розпізнати зображення, що містить один символ. Текст, який має довільну довжину та послідовність символів, розбирається за допомогою періодичної нейронної мережі (RNN) та довгострокової короткочасної пам'яті (LSTM), де LSTM є популярною формою RNN. Вхідне зображення Tesseract у форматі LSM обробляється у полях (прямокутник), рядок за рядком, який вставляється в модель LSTM та дає результат.

За замовчуванням Tesseract розглядає вхідне зображення як сторінку тексту в сегментах. Ви можете налаштувати різні сегментації Tesseract, якщо вам потрібно захопити невелику область тексту із зображення.

Переваги:

- має відкритий вихідний код;
- досить легко навчити OCR розпізнавати потрібні шрифти і підвищити якість розпізнається інформації. Після швидких налаштувань бібліотеки і певного навчання якість результатів розпізнавання стрімко зростає;
- є підтримка різних мов програмування.

Недоліки:

- можлива початкова недостатня точність розпізнавання, що усувається шляхом тренування і навчання алгоритму розпізнавання;
- для розпізнавання тексту в реальному часі потрібна додаткова обробка отриманого зображення;
- невелика точність розпізнавання при використанні стандартних файлів з даними про шрифтах, словах і символах.

2.1.2 OpenCV

OpenCV (Open Source Computer Vision Library) являє собою бібліотеку програмного забезпечення для комп'ютерного зору з відкритим вихідним кодом і комп'ютерного навчання. OpenCV була створена для забезпечення загальної інфраструктури додатків для комп'ютерного зору і прискорення використання сприйняття машини в комерційних продуктах. Будучи ліцензованим BSD продуктом, OpenCV розширює можливості для використання і модифікації коду в комерційних проєктах [8].

В якості мети для розробки даної бібліотеки було поставлено підвищення ефективності в додатках реального часу. Функціонал бібліотеки реалізований на мові С. OpenCV має можливість використовувати багатоядерні процесори. У разі необхідності автоматичної оптимізації на платформах Intel, можливо додаткове придбання та інтеграція з бібліотекою IPP.

До складу бібліотеки входить більше 2500 оптимізованих алгоритмів, які включають в себе повний набір класичних і сучасних алгоритмів комп'ютерного зору і машинного навчання. Дані алгоритми можуть використовуватися для виявлення і розпізнавання осіб, ідентифікації об'єктів, класифікації дій людини на відео, відстеження рухомих об'єктів і рухів самої камери і т.д. OpenCV має більше 47 тисяч користувачів спільноти, а передбачувана кількість завантажень перевищує 14 мільйонів. Бібліотека широко використовується в компаніях, дослідницьких групах і в урядових органах.

Крім таких відомих компаній як Google, Yahoo, Microsoft, Intel, IBM, є безліч різноманітних стартапів, таких як Applied Minds, VideoSurf і Zeitera, які так само використовують функціонал OpenCV.

Бібліотека має інтерфейси для C ++, Python, Java і MATLAB і підтримує Windows, Linux, Android і Mac OS. OpenCV орієнтується в основному на додатки в режимі реального часу і використовує переваги команд MMX і SSE, коли вони доступні. В даний час активно розвиваються повнофункціональні інтерфейси CUDA і OpenCL. На рисунку 2.3 представлена загальна структура проекту OpenCV.



Рисунок 2.3 – структура проекту OpenCV

Я буду використовувати цю бібліотеку для попередньої обробки для виявлення тексту з файлу зображення. Tesseract вимагає чистого зображення для виявлення тексту, саме тут OpenCV відіграє важливу роль, оскільки виконує такі дії над зображенням, як перетворення кольорового зображення у двійкове зображення, регулювання контрастності зображення, виявлення країв та багато іншого.

Переваги OpenCV:

- наявність ліцензії BSD;
- велика кількість інструмента для роботи з зображенням;
- можливість аналізу та розпізнавання на відео.

Недоліки CodeIgniter:

- важко створити ідеальну систему;
- невисока швидкість розпізнавання;
- потребує багато ресурсів на роботу;
- треба використовувати з іншими бібліотеками для підвищення результатів розпізнавання.

2.1.3 Anyline

Anyline надає багатоплатформний SDK, який дозволяє розробникам легко інтегрувати функції OCR в додатки. Дана OCR бібліотека цікавить численними можливостями налаштування параметрів розпізнавання і наданими моделями для вирішення конкретних прикладних завдань. Варто зазначити, що бібліотека платна і призначена для комерційного використання [9].

Переваги:

- досить просте налаштування розпізнавання потрібних шрифтів;
- розпізнавання тексту в реальному часі;
- легка і зручна настройка параметрів розпізнавання;
- бібліотека може розпізнавати штрих-коди і QR-коди;
- надає готові модулі для вирішення різних завдань.

Недоліки:

- невисока швидкість розпізнавання;
- для отримання задовільних результатів потрібно первісна настройка шрифтів для розпізнавання;
- бібліотека платна.

2.1.4 AForge.Net

AForge.NET є бібліотекою з відкритим вихідним кодом, створеною на мові C #, яка призначена для розробників і дослідників в області комп'ютерного зору. Крім того, в бібліотеці є функціонал для розробників в області штучного інтелекту. Спектр можливостей бібліотеки досить широкий: обробка зображень, нейронні мережі, генетичні алгоритми, нечітка логіка, машинне навчання, робототехніка і багато іншого [10].

Бібліотека включає кілька основних компонентів. AForge.Imaging – бібліотека підпрограм для обробки зображень і фільтрів. AForge.Vision – бібліотека комп'ютерного зору. AForge.Video - набір бібліотек для роботи з відео. AForge.Neuro – бібліотека для виконання різноманітних дій і операцій з нейронними мережами. AForge.Genetic – бібліотека підпрограм для використання генетичних алгоритмів для вирішення різних завдань. AForge.Fuzzy – бібліотека для роботи з нечіткою логікою. AForge.Robotics – бібліотека, яка забезпечує підтримку деяких методів, що застосовуються в сфері робототехніки. AForge.MachineLearning – бібліотека для роботи з елементами машинного навчання [11]. Існує, так само як і у бібліотеки OpenCV, активна спільнота, в якому можна зачерпнути необхідну інформацію, ставлячи питання розробникам, або поділитися власними напрацюваннями. Але, на жаль, кількість учасників цієї спільноти поступається своєю кількістю аналогічним по OpenCV. Ще одним обмеженням на шляху розробника є той факт, що вся документація по бібліотеці написана тільки на англійській мові. Зважаючи на це можливі труднощі у вивченні і освоєнні даного фреймворку. На рисунку 2.4 наведена загальна структура бібліотеки AForge.NET.

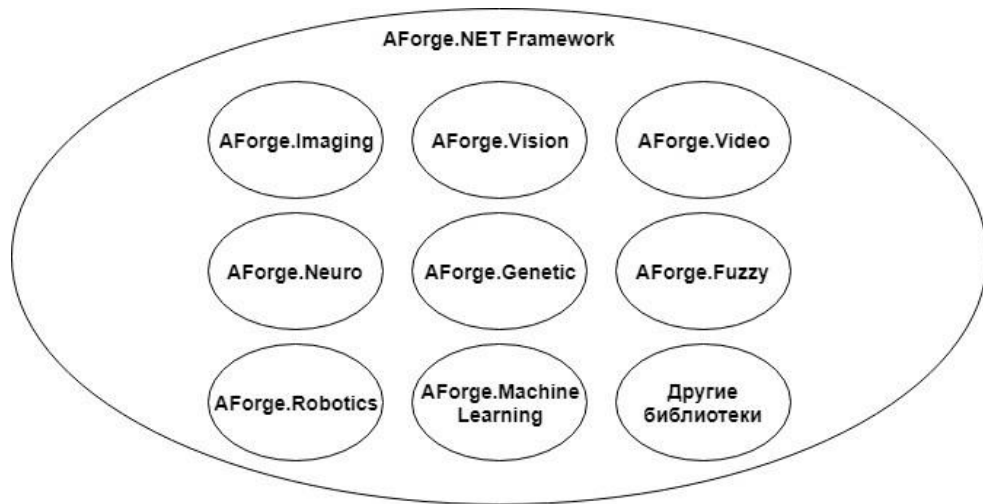


Рисунок 2.4 – Загальна структура бібліотеки AForge.NET

2.1.5 VXL

VXL – це набір бібліотек, написаних на мові C++, які призначені для наукових досліджень і реалізації технологій комп'ютерного зору [12]. VXL була написана в ANSI / ISO C ++ і призначена для портативних платформ. Бібліотека складається з декількох основних складових: VNL (числа) – чисельні алгоритми і контейнери, наприклад, матриці, вектори, оптимізатори і т.д., VIL (зображення) – завантаження, збереження і редагування зображень у багатьох найбільш поширених форматах (також існує можливість роботи з дуже великими зображеннями), VGL (геометрія) – геометрія точок, кривих та інших елементарних об'єктів в одно-, дво- і тривимірному просторах, VSL (вхідний і вихідний потоки), VBL (основні шаблони), VUL (утиліти) – різний функціонал для незалежних платформ. Особливість бібліотеки полягає в тому, що кожен її компонент може використовуватися окремо, не посилаючись на інші компоненти. Таким чином, в додатку можна використовувати тільки те, що дійсно необхідно. На рисунку 2.5 наведено ієрархічну будову ядра VXL [13].

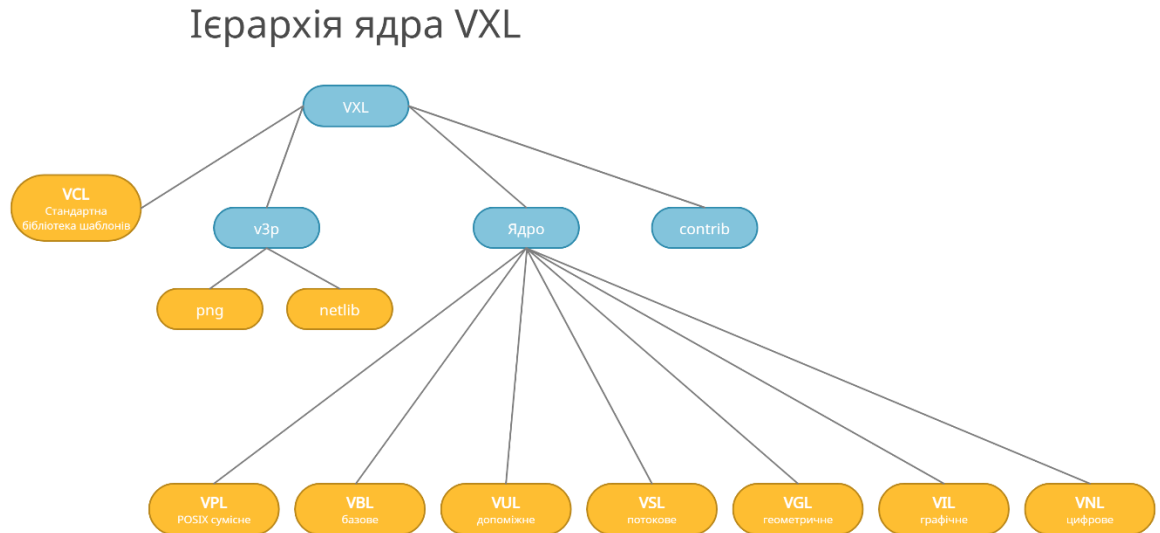


Рисунок 2.5 – Ієрархічна будову ядра VXL

2.1.6 LTI

LTI або LTI-lib – об'єктно-орієнтована бібліотека алгоритмів і структур даних. Вона часто застосовується при обробці зображень і в сфері комп'ютерного зору. LTI-lib була розроблена як частина науково-дослідних проектів в області комп'ютерного зору з технологіями робототехніки, розпізнавання об'єктів, голосу і жестів. Основною метою розробки даної бібліотеки є створення об'єктно-орієнтованої бібліотеки на мові C++, що багато в чому спрощувало б використання коду і його обслуговування, але при цьому були б забезпечені швидкі алгоритми, які можна було б використовувати в реальних додатках [14].

Бібліотека була розроблена з застосуванням GCC (набір компіляторів, застосовуваний для різноманітних мов програмування) під Linux і Visual C++ під Windows NT. Багато класів інкапсулюють Windows / Linux функціональність для того, щоб спростити вирішення системних або апаратних завдань (наприклад, класи для багатопоточності і синхронізації, виміру часу і доступ до послідовного порту).

2.1.7 Google Text Recognition API

Google Text Recognition API – це процес виявлення тексту в зображеннях і відео потоках і розпізнавання міститься в ньому тексту. Після виявлення розпізнавач визначає фактичний текст в кожному блоці і розбиває його на слова і рядки. Він виявляє текст різних мов (французька, німецька, англійська і т.д.) в режимі реального часу [15].

Варто відзначити, що, в цілому, дана OCR з поставленим завданням впоралася. Ми отримали можливість розпізнавати текст як в real-time, так і з уже готових зображень текстових документів. В ході аналізу даної бібліотеки ми виявили як переваги, так і недоліки її використання.

Переваги:

- можливість розпізнавання тексту в реальному часі;
- можливість розпізнавання тексту з зображень;
- невеликий розмір бібліотеки;
- висока швидкість розпізнавання.

Недоліки:

- великий розмір файлів з навченими даними (~ 30Mb);
- потрібне постійне підключення до інтернету;
- платна для комерційного використання.

2.2 Технології розробки

В якості середовища розробки було обрано PyCharm. PyCharm – інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний зневаджувач, інструмент для запуску юніт-тестів і підтримує веб-розробку на Django. PyCharm розроблена російською компанією JetBrains на основі IntelliJ IDEA [16].

PyCharm працює під операційними системами Windows, Mac OS X і Linux.

PyCharm являє собою інтелектуальний редактор для Python, HTML і JavaScript з можливостями аналізу коду у реальному часі, запобігання помилок у початковому коді і автоматизованими засобами рефакторінга для Python та JavaScript. Автодоповнення коду в PyCharm підтримує специфікацію Python будь-якої версії, включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис коротких масивів. Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

Основні переваги:

- статичний аналіз коду, підсвічування синтаксису і помилок;
- навігація серед проектів і сирцевого коду: відображення файлової структури проекту, швидкий перехід між файлами, класами, методами і використаннями методів;
- рефакторинг: перейменування, витяг методу, введення змінної, введення константи, підняття і опускання методу тощо;
- інструменти для веб-розробки з використанням фреймворку Django;
- вбудований зневаджувач для Python;
- вбудовані інструменти для юніт-тестування;
- розробка з використанням Google App Engine;
- підтримка систем контролю версій: загальний користувацький інтерфейс для Mercurial, Git, Subversion, Perforce і CVS з підтримкою списків змін та злиття.

На рисунках 2.6 – 2.8 зображено можливості середовища PyCharm.

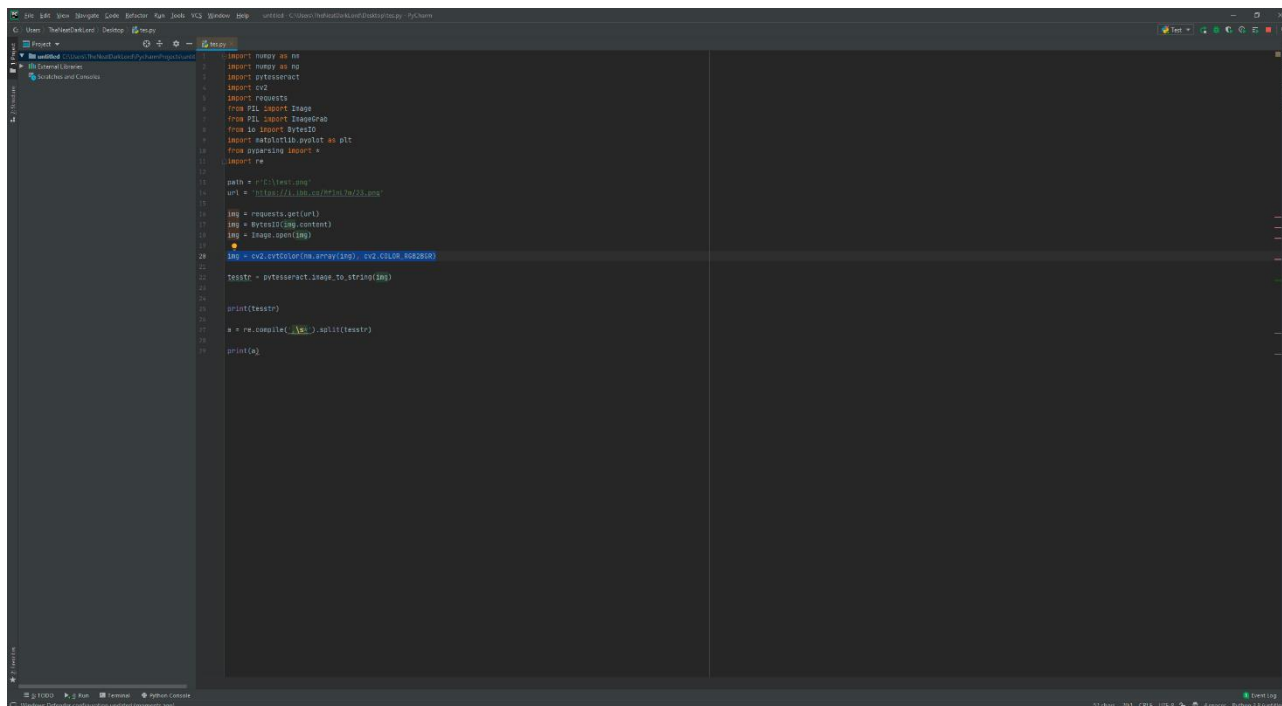


Рисунок 2.6 – Підсвічування коду у програмі PyCharm

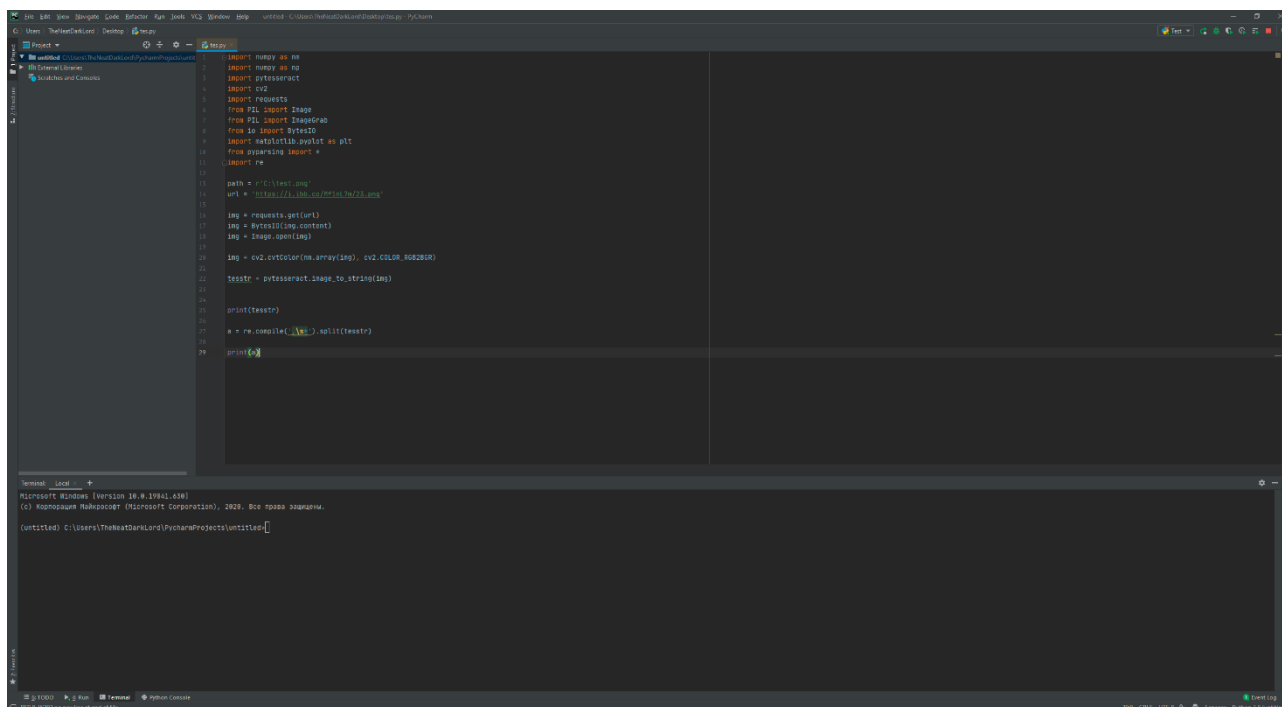


Рисунок 2.7 – Робота з консоллю у PyCharm

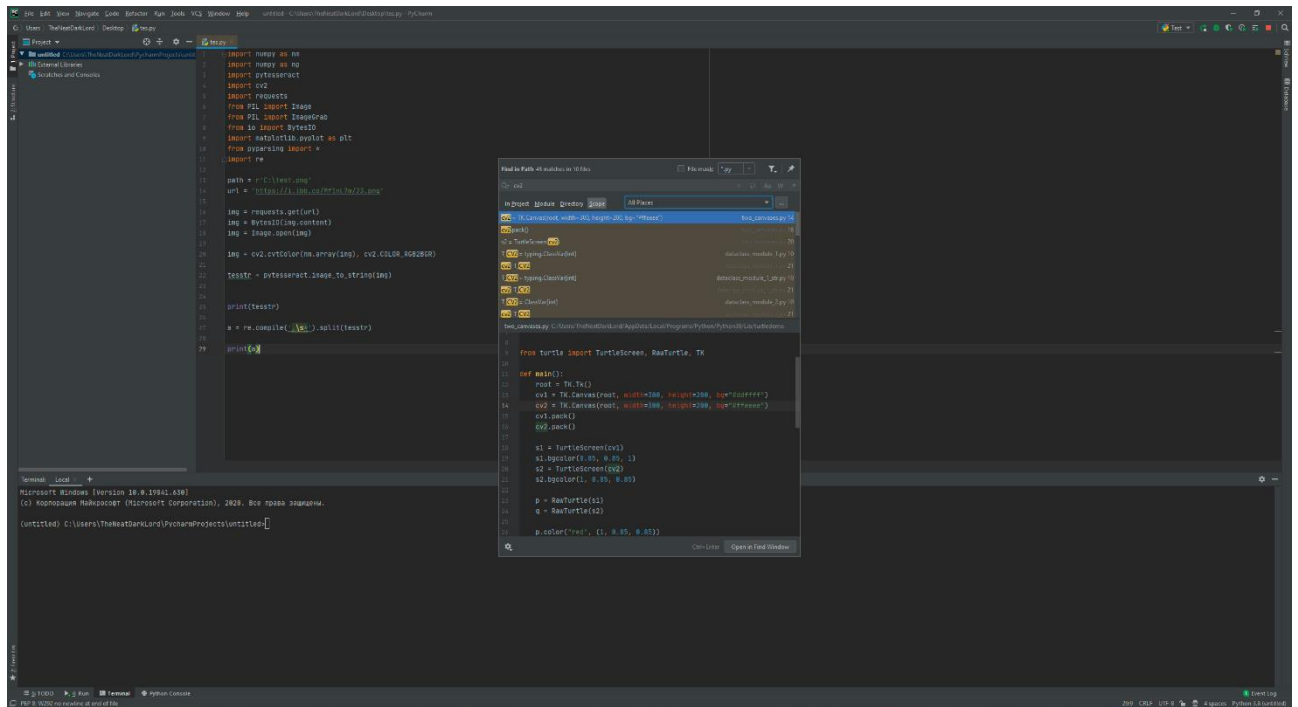


Рисунок 2.8 – Робота пошуку по всьому проекту у PyCharm

2.3 Висновки за розділом 2

З проаналізованих бібліотек найбільш відповідними до поставлених задач є бібліотека Tesseract та OpenCV, тому що вони надають достатній функціонал для реалізації сучасної системи автоматизованої фіксації результатів турніру з великою швидкістю роботи та інтуїтивно зрозумілим інтерфейсом, а також підходять для обраної мови розробки Python.

3 ОСНОВНІ РІШЕННЯ З РЕАЛІЗАЦІЇ СИСТЕМИ

3.1 Особливості системи

3.1.1 Логічна структура системи

Для вирішення поставленої задачі було запропоновано наступну логічну схему роботи системи (рис. 3.1) та схему функціонування автоматичної фіксації результатів (рис. 3.2).

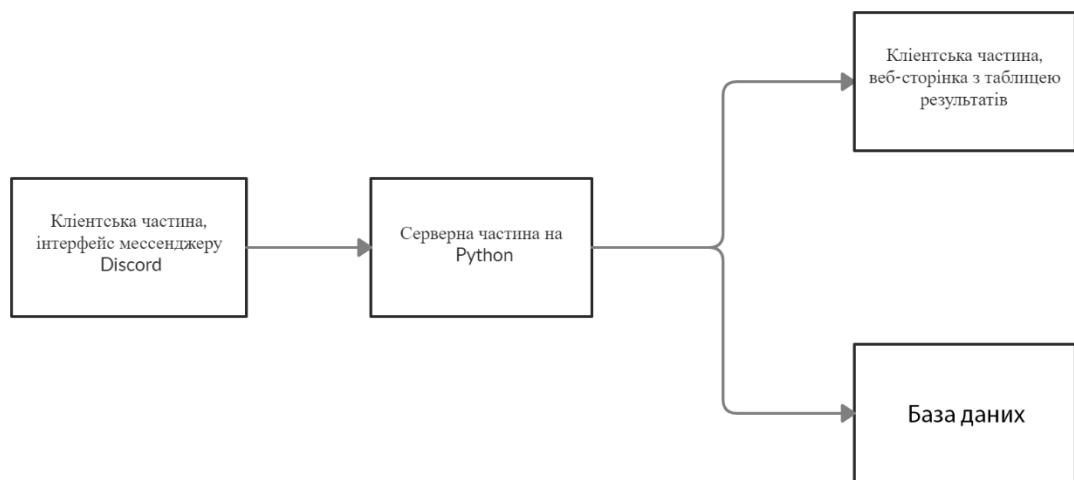


Рисунок 3.1 – Логічна схема роботи системи

Клієнтська частина відповідає за графічний інтерфейс користувача (ГІК) та дії користувача та адміністратора системи, в нашому випадку вона представлена інтерфейсом месенджеру Discord з можливістю командами відправляти запити до серверної частини та управляти процесом проведення турніру.

Серверна частина представляє собою скрипт, який написаний на Python та працює на сервері в асинхронному режимі, він отримує команди з клієнтської частини та відповідає за проведення турніру, автоматичну фіксацію результатів та збереження їх до бази даних.

Клієнтська частина, яка представлена веб-сторінкою відображає фінальну рейтингову таблицю по усім учасникам турніру.

Також система відповідає системними повідомленнями на усі команди користувача до текстового каналу у месенджер Discord.

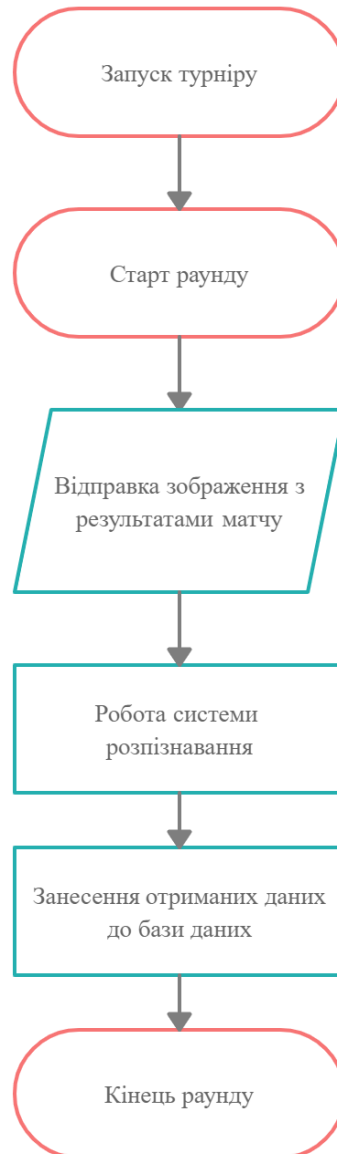


Рисунок 3.2 – Схема функціонування автоматичної фіксації результатів

Після старту раунду, коли учасник вибуває з матчу чи перемагає, зображення з результатами матчу потрапляє до текстового чату в месенджері Discord.

Розроблена модель отримує це зображення та за допомогою послідовності певних кроків виокремлює з нього результати учасника і записує їх до бази даних для подальшого виведення в рейтинговій таблиці на сайті.

Бібліотека Tesseract містить вбудовану нейромереживу модель для розпізнавання символів різними мовами, але сама вона являє собою модель типу "чорна скриня". В роботі запропоновано підвищити ефективність результатів її роботи завдяки попередній обробці вхідного файлу методами обробки зображень, а саме cvtColor, GaussianBlur, threshold. Реалізація цих методів була впроваджена засобами бібліотеки OpenCV.

Постобробка результатів на основі регулярних виразів та використанні бібліотеки PyParsing дозволяє виокремлювати шукані результати з усього масиву текстової інформації, отриманого з зображення.

На рисунку 3.3 представлена повна модель обробки та розпізнавання зображення.

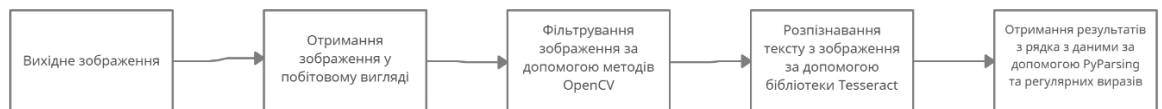


Рисунок 3.3 – Модель обробки та розпізнавання зображення

3.2 Особливості мови програмування Python

Python – це інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією [17].

Синтаксис ядра Python мінімалістичний. У той же час стандартні функції можна легко розширити за допомогою бібліотек, які включають великий обсяг корисних функцій для вирішення будь-яких задач. Python підтримує багато парадигм програмування: структурне, об'єктно-орієнтоване, функціональне, імперативне.

Основні архітектурні риси – динамічна типізація, механізм обробки виключень, автоматичне керування пам'яттю, підтримка багатопоточних обчислень і зручні високорівневі структури даних.

Код в Python організовується у функції та класи, які можуть об'єднуватися в модулі.

Python – це мова програмування, що активно розвивається, нові або оновлені версії виходять приблизно раз в півроку-рік. Python підтримує динамічну типізацію, тобто тип змінної визначається лише під час виконання і не треба задавати це вручну. В Python є багато вбудованих типів: буліан, рядок, Unicode-рядок, ціле число довільної точності, число з плаваючою крапкою, комплексне число ті інші. З колекцій в Python є: список, кортеж, словник, множина та інші. Всі значення є об'єктами, в тому числі функції, методи, модулі, класи. Додати новий тип можна лише написавши клас (class) або визначивши новий тип в модулі розширення. Система класів підтримує спадкування (одиначне і множинне) і метапрограмування. Можливо успадкування від більшості вбудованих типів і типів розширень.

Всі об'єкти діляться на посилальні і атомарні. До атомарних відносяться int, long та інші. При присвоєнні атомарних об'єктів копіюється їх реальне значення, в той час як для посилальних копіюється тільки покажчик на об'єкт, таким чином, обидві змінні після присвоювання можуть використовувати одне і те ж значення. Довідкові об'єкти бувають змінювані і незмінні. Наприклад, рядки і кортежі є незмінними, а списки, словники і інші об'єкти – змінними.

3.3 Особливості використання бібліотеки ІО та функції BytesIO

Зображення з результатами розроблена модель отримує за допомогою Discord API і потім переводимо у побітовий вигляд для полегшення роботи з зображенням в майбутньому.

Об'єкт BytesIO не пов'язаний з будь-яким реальним файлом на диску. Це всього лише фрагмент пам'яті, який веде себе як файл. Він має той самий API,

що і файл-об'єкт, в якому функція `open` з режимом `r + b` дозволяє читати і записувати двійкові дані.

`BytesIO` (це близький родич `StringIO`, який завжди знаходиться в текстовому режимі) може бути корисний, коли вам потрібно передавати дані в API або з нього, які очікують надання файлового об'єкта, але там, де ви вважаєте за краще передавати дані безпосередньо до програми. Ви можете завантажити свої вхідні дані у `BytesIO` перш ніж передавати їх в бібліотеку. Після його повернення ви можете отримати будь-які дані, які бібліотека записала в файл з `BytesIO` використовуючи метод `getvalue()`.

Використання функції `open` дозволяє відкривати файли, які збережені до оперативної пам'яті за допомогою функцій `Bytes IO` [18].

3.4 Особливості використання бібліотеки OpenCV та дослідження її ефективності

Після отримання зображення та переведення його у побітовий вигляд потрібно провести фільтрацію зображення за допомогою вбудованих функцій бібліотеки для покращення точності подальшого розпізнавання результатів.

OpenCV – це бібліотека комп'ютерного зору і машинного навчання з відкритим вихідним кодом. У неї входять понад 2500 алгоритмів, в яких є як класичні, так і сучасні алгоритми для комп'ютерного зору і машинного навчання. Ця бібліотека має інтерфейси на різних мовах, серед яких є Python, Java, C++ і Matlab.

Перш за все було використано функцію `cvtColor` для зміни колірного простору з BGR, у якому ми отримуємо це зображення, до RGB, а також знебарвлення зображення до чорно-білого варіанту.

```
img = cv2.cvtColor(nm.array(img), cv2.COLOR_RGB2BGR)
```

Проведемо аналіз та передамо до Tesseract зображення після обробки першою функцією. На виході отримуємо знебарвлене зображення (рис. 3.4), а також результуючий рядок з розпізнаним текстом (рис. 3.5).



Рисунок 3.4 – Вихідне зображення після функції cvtColor

```
R 7
| le hd
> ' @
é Y
_ <- R
A ,
> - }
: <= = > I< i
Assists;0 8 \* ens " = H
Time Survived: 811 = \
seconds , s |
- ON
Accolades XP: 199 . ee. ee = ee
ee i ie Peta ae
ed » (ie SG a FON Me 4 A Aihataler \
0
0 he Sa a
a Serre ee hea ASSISTS: 0
= exit we Sy
any <e
INS a) -
io* Final Se
Q
```

Рисунок 3.5 – Результуючий рядок з розпізнаним текстом

Як видно з рисунка 3.5, після такої обробки зображення Tesseract не зміг коректно розпізнати необхідний текст з результатами матчами.

Далі використовуємо функцію для видалення шуму з зображення GaussianBlur У цьому підході замість коробчатого фільтра, що складається з рівних коефіцієнтів фільтра, використовується ядро Гауса. OpenCV пропонує в основному чотири типи методів розмиття, в нашому випадку найефективнішим є саме Гаусове розмиття. Це робиться за допомогою функції cv2.GaussianBlur(). Ми повинні вказати ширину та висоту ядра, які повинні бути позитивними та непарними. Ми також повинні вказати стандартне відхилення у напрямках X та Y, sigmaX та sigmaY відповідно. Якщо вказано лише sigmaX, sigmaY приймається рівним sigmaX. Якщо обидва дані вказані як нулі, вони обчислюються на основі розміру ядра. Гаусова фільтрація є дуже ефективною для усунення гауссових шумів із зображення. Це фактично видаляє високочастотний вміст (наприклад: шум, краї) із зображення, в результаті чого краї розмиваються, коли застосовується цей фільтр.

```
blur = cv2.GaussianBlur(img, (5,5), 0)
```

Проведемо аналіз точності розпізнавання з використанням двох функцій. Результуюче зображення після обробки (рис. 3.6) та рядок з розпізнаним текстом (рис. 3.7).



Рисунок 3.6 – Вихідне зображення після обробки двома функціями

```
= . bd
» r
es (3 .
f
ee ee ee J
___ Assists: 0 =e,
SO aie eo s '
Prato) To \ "a
aan Nl
[Rees et ee Ce . ~ ~ _-
— ~ ee aly
LLGBILALED a
x) & EXILES: 0
Pa ee Cn)
= Exit re) rhe )
rods =
Se) =
i. * Final 7
Q
```

Рисунок 3.7 – Результуючий рядок з розпізнаним текстом

Як видно з рисунка 3.7 після обробки зображення двома функціями після розпізнавання Tesseract зміг видобути більше корисної інформації, але точність розпізнавання все ще є досить малою і не надає можливості виокремити конкретний результат.

Наступним кроком приберемо усю зайву інформацію за допомогою ступеню яскравості замінив її на білий колір, а корисну інформацію навпаки виділимо чорним. Для цього використовується функція `threshold`, де `cv2.THRESH_BINARY_INV` передається як додатковий прапор. Порогове значення ми вибираємо довільно та намагаємося знайти оптимальне значення до поки не залишаться лише корисні частини зображення. Також завдяки використанню прапору `cv.THRESH_OTSU` алгоритм буде знаходити оптимальне порогове значення, яке повертатиметься як перший вихід [19].

```
thresh = cv2.threshold(blur, 0, 255,
cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU) [1]
```

Проведемо аналіз точності розпізнавання з використанням трьох функцій. Результуюче зображення після обробки (рис. 3.8) та рядок з розпізнаним текстом (рис. 3.9).

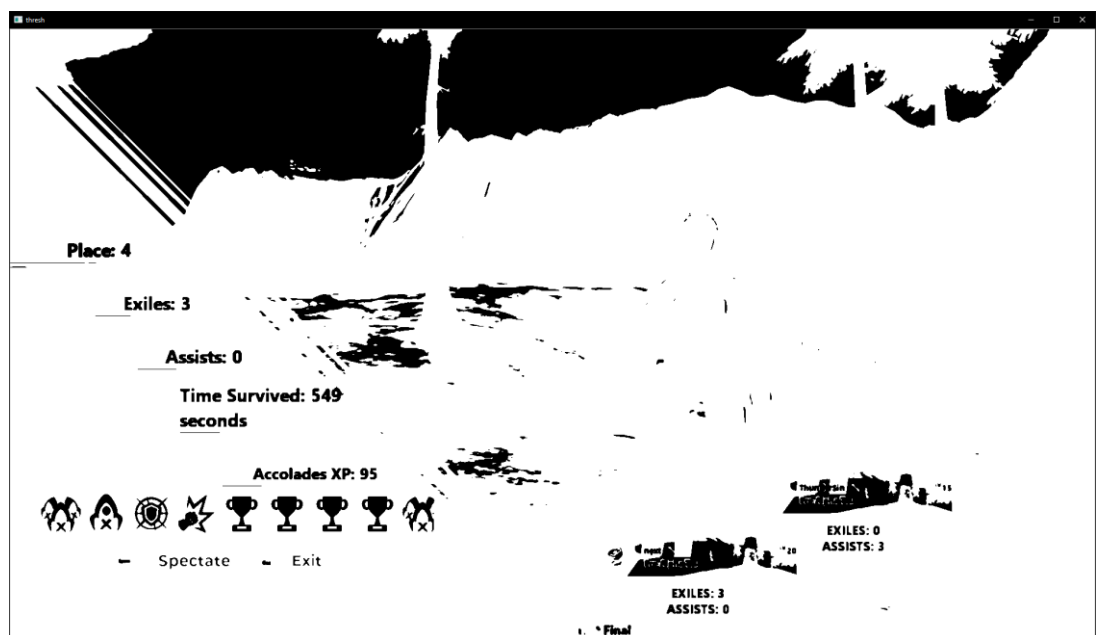


Рисунок 3.8 – Вихідне зображення після обробки трьома функціями

```

Place: 4
Exiles: 3

Assists: 0

Time Survived: 549
seconds

Accolades XP: 95

OG LLL LR

Thundersin =f : '
Pre-Alpha Tester

EXILES: 0
os 4 ~ Tyg ASSISTS: 3
"Pre-Alpha Tester

EXILES: 3
ASSISTS: 0

* Final
Q

```

Рисунок 3.9 – Результуючий рядок з розпізнаним текстом

Як видно з рисунка 3.9 після використання функції з видалення шумів у купі з іншими функціями ми отримуємо майже ідеальний рядок з розпізнаними результатами.

3.5 Особливості використання бібліотеки Tesseract

Python-tesseract – це обгортка для Google Tesseract-OCR Engine. Він також корисний як самостійний сценарій виклику для tesseract, оскільки він може читати всі типи зображень, що підтримуються бібліотеками зображень Pillow і Leptonica, включаючи jpeg, png, gif, bmp, tiff та інші. Крім того, якщо Python-tesseract буду використовувати його як скрипт, він зможе надрукувати розпізнаний текст замість того, щоб записувати його у файл.

Особливостями роботи бібліотеки є те що, якщо зображення попередньо не обробляти, точність зчитування є дуже низькою та в виводі є багато зайвої інформації. Для розроблюваної моделі з метою покращення результатів

розпізнавання слід додати розширення для підключення додаткових баз-словників. Підключимо словник з англійськими словами, бо у змаганні, що використовується для тестування системи, інтерфейс та результати виводяться саме цією мовою. Python-tesseract має додаткові словники для української та багатьох інших мов, тож модель можна адаптувати під потреби проєкту, на якому вона буде використовуватись.

```
txt = pytesseract.image_to_string(img, lang='eng')
```

3.6 Особливості використання бібліотеки PyParsing

Модель розпізнавання Tesseract у комбінації з методами попередньої обробки зображень виокремлює всі символи, що присутні на слайді, тож є потреба видобути саме результати матчу, відділивши їх, наприклад, від імені гравця або назви команди. Зробити це можна за допомогою PyParsing та регулярних виразів, орієнтованих на пошук певних слів.

PyParsing – модуль синтаксичного аналізу для мови Python. В нашому випадку виконується пошук по кінцівкам слів Assist (допомоги) та Exiles (вбивства) для отримання цілочисленного результату цих параметрів гравця за матч [20].

```
parse_module.parseString(r'ists:\s?\d{1,2}',txt).split(':')[1].strip()
parse_module.parseString(r'iiles:\s?\d{1,2}',txt).split(':')[1].strip()
```

3.7 Схеми бази даних

За базу даних була обрана реалізація SQL. За останній час стали набирати популярності нереляційні бази даних, такі як MongoDB і NoSQL. Слід зазначити, що нереляційні бази даних дуже добре застосовні до вузького спектру завдань, але значно поступаються реляційним, якщо мова заходить про «класичних» базах даних. Така база розробляється без знань про всі можливі застосування в системі. Правила нормалізації якраз покликані підготувати базу даних до будь-яких запитів від користувача з найменшими витратами за часом і пам'яті [21].

Варто відзначити вбудований в Python інструментарій для роботи з базами даних за допомогою MySQL. Наша система проводить запис одразу до двох баз даних, до власної бази даних (рис. 3.10), а також до бази даних, яка використовується для веб-сторінки з рейтинговою таблицею (рис. 3.11).

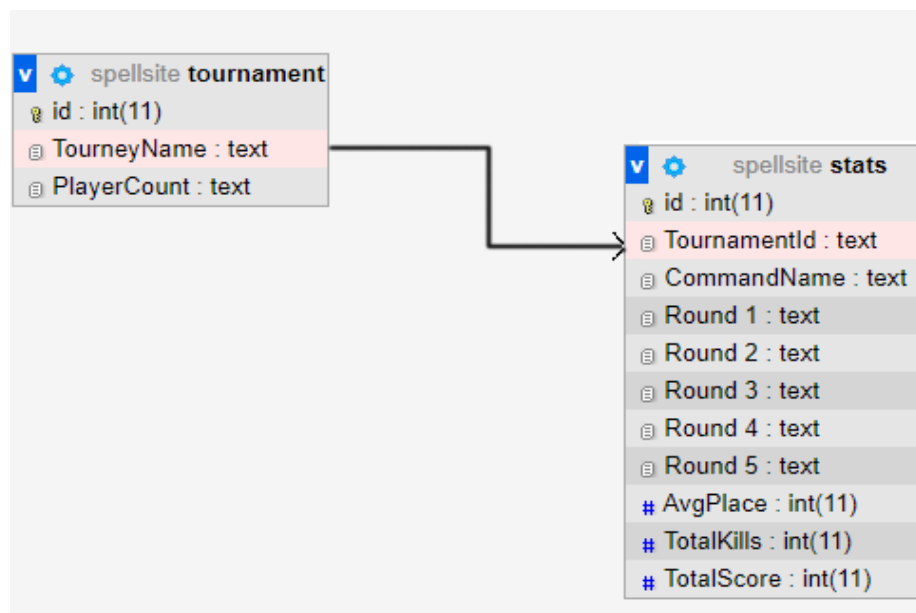


Рисунок 3.10 – База даних для веб-сторінки з рейтинговою таблицею

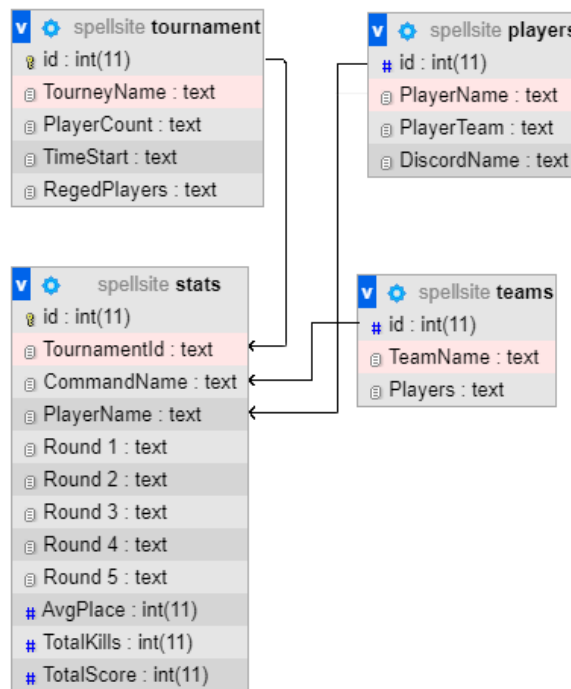


Рисунок 3.11 – База даних системи

3.8 Frontend

Вище була розглянута серверна складова проекту та частина програми, яка оброблює запити від клієнтської частини, вона ще називається frontend і повертає відповіді до неї. Frontend – це інтерфейс взаємодії між користувачем і основний програмно-апаратною частиною.

До фронтенду входять всі модулі, які так чи інакше безпосередньо генерують і впливають на роботу системи. В нашому випадку основний інтерфейс представлений можливостями текстового каналу месенджеру Discord, який завдяки широкій інтеграції з системою завдяки API дозволяє керувати системою на усіх етапах проведення турніру, а також відправки зображення для автоматичної фіксації результатів змагання з них.

Іншим інтерфейсом є сторінка веб-сайту з фінальною рейтинговою таблицею, яка дозволяє переглядати результати під час турніру, а також остаточні результати після його завершення.

3.9 Висновки за розділом 3

Як видно з дослідження, використання попередньої обробки зображення за допомогою вбудованих функцій OpenCV значно підвищує точність розпізнавання тексту з зображення, що робить таку модель розпізнавання доволі ефективною для розв’язання поставленої задачі і доцільною у використанні в автоматизованій системі фіксування результатів кіберспортивних змагань.

Також було спроектовано базу даних, у якій зберігаються уся інформація про гравців, команди та результати усіх проведених змагань.

Використання месенджера Discord у якості інтерфейсу для управління системою дало змогу зробити роботу з системою зручною та нескладною для користувачів.

4 КЕРІВНИЦТВО ПРОГРАММІСТА

4.1 Призначення системи

Система призначена для відстеження проведення кіберспортивних змагань онлайн, а також автоматичного фіксування результатів та їх збереження для подальшого використання.

4.2 Умови експлуатації програми

Для успішної роботи з системою достатньо мінімальних системних ресурсів, а саме:

- операційна система: Windows XP, Vista, 7, 8, 10, Mac OS 10.6 або більш пізня версія, Ubuntu 10.04 або більш пізня версія;
- процесор: Intel Pentium 4 / Athlon 64 або пізнішої версії з підтримкою SSE2;
- ОЗУ: 2 Гб;
- месенджер Discord;
- язык програмування Python, середовище розробки Python та додаткові бібліотеки.

4.3 Характеристика програми

Сервіс розроблений за допомогою сучасної мови програмування Python та додаткових прикладних та має стандартну структуру додатків на Python, що дозволяє додати будь-який новий функціонал до системи.

Поточний проект реалізує наступні функціональні вимоги:

- створення та проведення змагання через месенджер Discord;
- реєстрація учасників та персоналізація їх даних;
- онлайн проведення та автоматичне фіксування результатів змагання;

- виведення результатів турніру на веб-сторінку у режимі онлайн;
- автоматичне переміщення учасників турніру;
- повне управління ходом турніру через спеціальні команди адміністратора.

4.4 Звернення до системи

Для запуску системи потрібно запустити Python скрипт, який запускає систему та користуватися її функціоналом за допомогою інтерфейсу месенджера Discord.

4.5 Вхідні і вихідні дані

Вхідними даними до програми є інформація про учасників, а також зображення з ігрової сесії на яких містяться результати матчу.

Вихідними даними є база даних з результатами турніру, а також веб-сторінка, де ці результати можна побачити.

4.6 Повідомлення

При виникненні виняткових ситуацій чи помилки, буде виведене текстове повідомлення, що сповіщає про те, яка помилка трапилась, також інформація про виникаючі помилки логується до файлу `error.log`, який знаходиться в кореневій папці проекту.

5 КЕРІВНИЦТВО КОРИСТУВАЧА

5.1 Робота з системою

Для того щоб запустити систему та відкрити реєстрацію на турнір треба у відповідному чаті вписати команду `!openregistration` з параметрами:

- solo для турніру, де команда складається з одного учасника;
- duo для турніру, де команда складається з двох учасників;
- trio для турніру, де команда складається з трьох учасників;
- squad для турніру, де команда складається з чотирьох учасників. (рис

5.1)

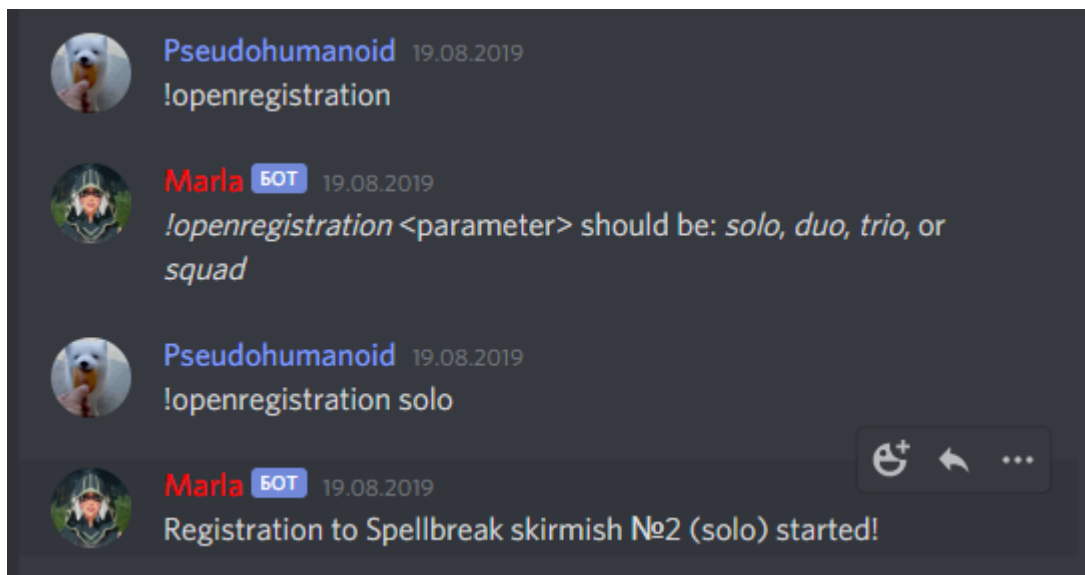


Рисунок 5.1 – Команда відкриття реєстрації на турнір

Для того, щоб почати користуватися системою, а саме зареєструватися у турнірі, треба у відповідному чаті вписати команду `!reg` з параметром, який містить назву команди чи нікнейм гравця, або без параметру для автоматичного отримання нікнейму з месенджера Discord (рис. 5.2).

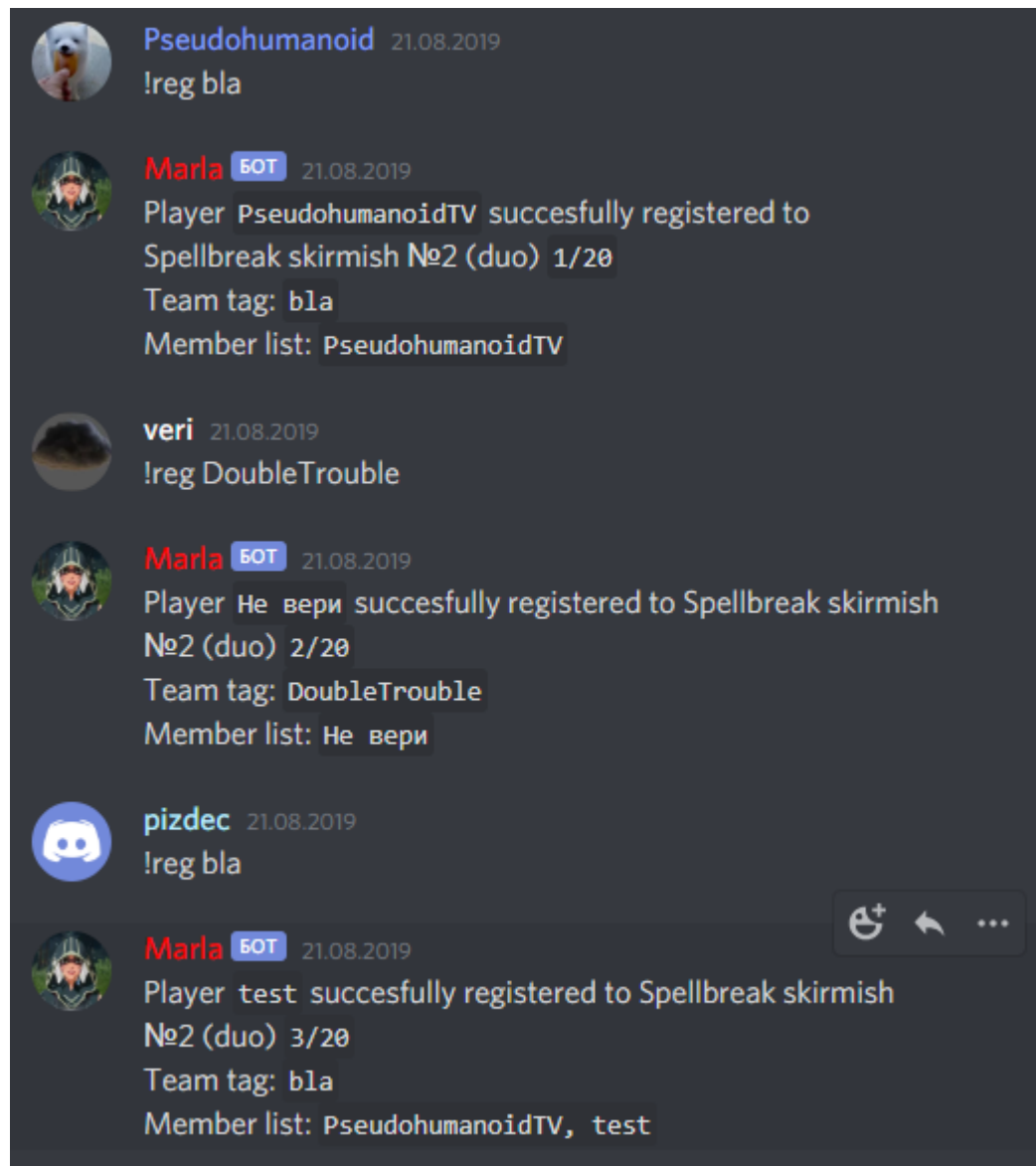


Рисунок 5.2 – Команда реєстрації на турнір

Для того щоб відмінити реєстрацію та відмовитися від участі у змаганні треба ввести !unreg (рис. 5.3).



Рисунок 5.3 – Команда відмови від реєстрації на турнір

Після реєстрації усіх учасників для запуску турніра треба ввести команду !startskirmish, а для початку раунду !startround (рис. 5.4).

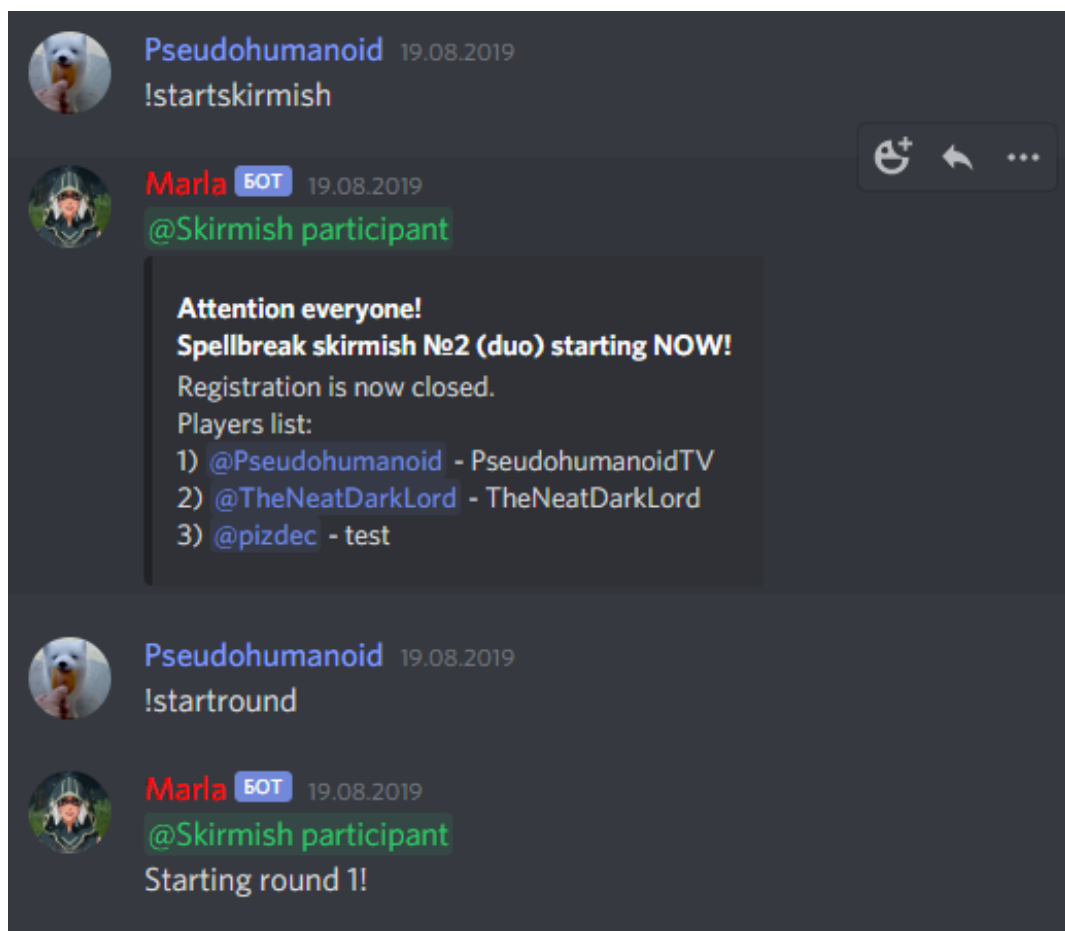


Рисунок 5.4 – Команди старту турніру та раунду

Після введення команди !startround потрібно до спеціально відведеного чату завантажувати зображення з результатами ігрової сесії для занесення їх до системи (рис. 5.5).

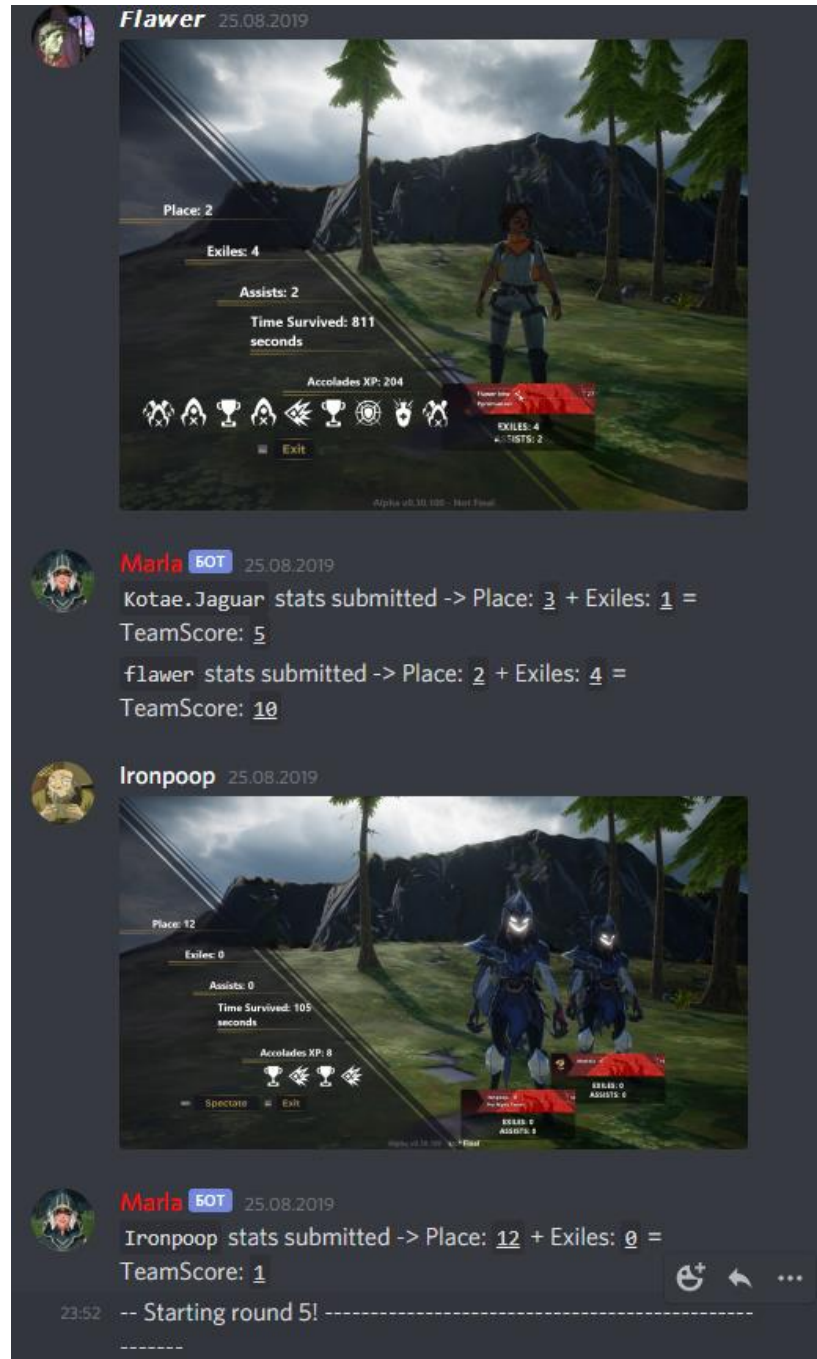


Рисунок 5.5 – Відправка зображень до системи

Для того щоб переглянути турнірну таблицю на даний час необхідно перейти на веб-сторінку, яка створюється для кожного турніру окремо (рис. 5.6).

ГЛАВНАЯ НОВОСТИ ТУРНИРЫ МЕДИА **SPELLBREAK** ГАЙДЫ КЛАССЫ ПРЕДМЕТЫ КОНТАКТЫ

SPELLBREAK SQUAD TOURNAMENT №2

Чтобы принять участие в турнире по Spellbreak заходите в дискорд и регистрируетесь в канале #skirmish-chat. Призовой фонд турнира 20\$ (на команду).
1 kill = 1 pt

GROUP PLACEMENT REWARDS:
1st place = 8pts
2nd place = 7pts
3rd place = 6pts
4th place = 5pts
5th place = 4pts
6-9th place = 3pts
10th+ place = 1pt

#	Name	Team name	Round 1	Round 2	Round 3	Round 4	Round 5	Average Place	Total Kills	Team Total Score
1		Hoodnibbas						3	23	50
	Dwerkie		1/5/15	1/4/16	2/2/10	8/0/1	3/3/8		14	
	Syncade v2		1/2/15	1/4/16	2/2/10	8/0/1	3/1/8		9	
2		ggflawer						4	14	36
	flawer		9/1/3	2/2/8	5/0/2	2/4/10	1/2/13		9	
	g00d_g00d		9/1/3	2/0/8	5/1/2	0/0/10	1/3/13		5	
3		englishspeakersmen						4	9	31
	TheNeatDarkLord		5/2/5	3/0/4	1/2/10	1/2/10	11/1/2		7	
	Akakievich		5/2/5	3/0/4	0/0/10	0/0/10	0/0/2		2	
4		nextSin						4	15	20
	next		4/0/1	6/3/4	4/3/6	4/2/5	4/3/4		11	

Рисунок 5.6 – Веб-сторінка з результатами турніру

Фінальна турнірна таблиця буде виведена також на веб-сторінці (рис. 5.7).

#	Name	Team name	Round 1	Round 2	Round 3	Round 4	Round 5	Average Place	Total Kills	Team Total Score
1		Hoodnibbas						3	23	50
	Dwerkie		1/5/15	1/4/16	2/2/10	8/0/1	3/3/8		14	
	Syncade v2		1/2/15	1/4/16	2/2/10	8/0/1	3/1/8		9	
2		ggflawer						4	14	36
	flawer		9/1/3	2/2/8	5/0/2	2/4/10	1/2/13		9	
	g00d_g00d		9/1/3	2/0/8	5/1/2	0/0/10	1/3/13		5	
3		englishspeakersmen						4	9	31
	TheNeatDarkLord		5/2/5	3/0/4	1/2/10	1/2/10	11/1/2		7	
	Akakievich		5/2/5	3/0/4	0/0/10	0/0/10	0/0/2		2	
4		nextSin						4	15	20
	next		4/0/1	6/3/4	4/3/6	4/2/5	4/3/4		11	
	ThunderSin		4/0/1	6/0/4	4/2/6	4/2/5	4/0/4		4	
5		Moctola						7	10	18
	Ironpoop		3/3/9	5/0/1	7/0/2	12/0/1	6/3/5		6	
	Moctola		3/2/9	5/0/1	7/1/2	0/0/1	6/1/5		4	
6		baba						5	1	17
	Kotae.Jaguar		12/0/1	4/0/1	3/0/4	3/1/5	2/0/6		1	
	GIADe non		12/0/1	4/0/1	3/0/4	0/0/5	0/0/6		0	
7		Banshee						0	0	0
	HoboBagginsTM		0/0/0	0/0/0	0/0/0	0/0/0	0/0/0		0	
8		Bodies						0	0	0
	Amadey11		0/0/0	0/0/0	0/0/0	0/0/0	0/0/0		0	
9		chat						0	0	0
	He aspx		0/0/0	0/0/0	0/0/0	0/0/0	0/0/0		0	
10		forZe.BubiSTAR						0	0	0
	forZe.BubiSTAR		0/0/0	0/0/0	0/0/0	0/0/0	0/0/0		0	
	forZe.RockSTAR		0/0/0	0/0/0	0/0/0	0/0/0	0/0/0		0	
11		Jaguar						0	0	0
	Jaguar.Deblerin		0/0/0	0/0/0	0/0/0	0/0/0	0/0/0		0	
12		Trash						0	0	0
	Borsalino		0/0/0	0/0/0	0/0/0	0/0/0	0/0/0		0	

Рисунок 5.7 – Веб-сторінка з фінальними результатами турніру

У каналі “правила та інформація” можна знайти усю необхідну інформацію про турнір, який планується. При натисканні на відповідний голосовий канал користувач зможе зайти до нього та почати спілкуватися.

5.2 Повідомлення користувачу

Як тільки турнір почнеться, усі учасники отримають відповідне повідомлення до текстового каналу та будуть переміщені у відповідні голосові чати своїх команд.

Якщо при реєстрації учаснику виводиться повідомлення, що такий учасник вже зареєстрований, то потрібно спочатку відмінити реєстрацію відповідною командою, а потім спробувати знов.

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

У атестаційній роботі магістра розроблена та реалізована автоматизована система фіксування результатів кіберспортивних змагань, що дозволяє фіксувати в режимі реального часу під час турніру результати гравців та заносити їх до таблиці, а також у кінці відображати рейтингову таблицю.

Існуючі аналоги таких систем для проведення турнірів та фіксування результатів (Toornament.com, Weplay.tv, Game.tv, Challonge.com) мають такі недоліки: неможливість автоматичної фіксації результатів, немає підтримки роботи з месенджером Discord, деякі з них потребують плати для використання усіх функцій. Всі ці недоліки накладають ряд обмежень на можливості використання даних систем.

Розроблена технологія дозволить повноцінно проводити турніри з можливістю швидко фіксувати результативність команд у кіберспортивних дисциплінах, де немає такої можливості у самій грі.

Передбачається, що система буде розповсюджуватися, як готовий продукт розрахований для одного користувача (адміністратора турніру) з вартістю 10000 грн. за копію, що значно дешевше за інші аналоги.

6.1 Планування розробки програмного виробу

Весь комплекс розробки програмного забезпечення можна поділити на етапи. Для кожного етапу вказується трудомісткість, кількість виконавців та тривалість робіт. У розробці приймають участь програміст впродовж 1 місяця і консультант впродовж 0,18 місяця. Розробка програми починається 1 листопада 2020 року та має бути завершена до 1 грудня 2020 року. Тривалість робіт розраховується за формулою 6.1:

$$T_{ц} = \frac{Q}{R}, \quad (6.1)$$

де $T_{\text{ц}}$ – довжина циклу, днів;

Q – трудомісткість, людино-днів;

R – кількість виконавців, люд.

Отримана інформація зведена в табл. 6.1., за даними якої складається зведений стрічковий графік планування розробки програмного продукту, який представляє собою таблицю, в першому стовпці якої розміщені в порядку збільшення термінів початку виконання всі види роботи, а навпроти – календарний період їх виконання. Даний графік наведено в табл. 6.2.

Таблиця 6.1 – Характеристика робіт по розробці програми

Найменування роботи	Трудомісткість		Виконавці		Трива- лість
	чол.- дн.	% до підсумку	спеціаль- ність	кіл. чол.	днів
1	2	3	4	5	6
1. Дослідження вимог до проектованого продукту	2	7,7	програміст	2	1
			консультант		
2. Розробка технічного завдання	2	7,7	програміст	2	1
			консультант		
3. Проектування загальної структури програми	4	15,38	програміст	2	2
			консультант		
4. Розробка схеми функціонування програми	2	7,7	програміст	1	2

Продовження табл. 6.1

1	2	3	4	5	6
5. Створення програмного коду	10	38,46	програміст	1	10
6. Тестування та налагодження програмного продукту	4	15,38	програміст	1	4
7. Складання програмної документації	2	7,7	програміст	1	2
Всього	26	100,0			22

Таблиця 6.2 – Лінійний графік Ганта розробки програмного продукту



Позначення: – програміст; – консультант.

Необхідні початкові дані:

- час на розробку – 1 міс.;
- консультант працює – до 0,18 міс. (4 дн.) із заробітною платою 4000 грн./міс;
- програміст працює – 1 міс. (22 дн.) із заробітною платою 6000 грн./міс.

Останнім етапом планування розробки програмного продукту є звідний стрічковий графік розробки програмного продукту є лінійний графік Ганта (табл. 6.2), побудований за даними таблиці 6.1.

6.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складається калькуляція кошторисної вартості робіт яка включає наступні статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок;
- витрати на спеціальне обладнання;
- матеріали і комплектуючі вироби;
- накладні витрати;
- податки.

6.2.1 Розрахунок основної заробітної плати

Витрати по цій статті складаються з планового фонду заробітної плати усіх категорій працівників, зайнятих в розробці програмного продукту. Розрахунок заробітної плати ведеться на основі відомостей про трудомісткість (табл. 6.1). Результати розрахунків зведені в табл. 6.3.

Таблиця 6.3 – Розрахунок основної заробітної плати

Посада виконавця	Кількість, осіб	Місячний оклад	Кількість, міс. роботи	Премії та доплати, грн.	Середня місячна зарплата, грн.	Сумарна зарплата, грн.
Програміст	1	6000	1	—	6000	6000
Консультант	1	4000	0,18	—	4000	720
Всього	2					6720

6.2.2 Розрахунок додаткової заробітної плати

Додаткову заробітну плату приймають рівною 10% від основної заробітної плати. Розрахунок основної та додаткової заробітної плати наведений в табл. 6.4.

Таблиця 6.4 – Розрахунок додаткової заробітної плати

Посада виконавця	Додаткова заробітна плата, грн.	Сума основної та додаткової заробітної плати, грн.
Програміст	600	6600
Консультант	72	792
Всього	672	7392

6.2.3 Відрахування на єдиний соціальний внесок

Відрахування на єдиний соціальний внесок визначають у відсотковому відношенні від суми основної і додаткової зарплат, з урахуванням премій і доплат. Премії і доплати в нашому випадку дорівнюють 0. Відрахування на соціальне страхування складають 22%.

$$ВІД = (ЗП_{осн} + ЗП_{дод}) \cdot 0,22, \quad (6.2)$$

де ВІД – сума відрахувань;

$ЗП_{осн}$ – основна заробітна плата, грн;

$ЗП_{дод}$ – додаткова заробітна плата, грн.

$$ВІД = (6600 + 792) \cdot 0,22 = 1626,24 \quad (6.3)$$

6.2.4 Визначення витрат на матеріали

Матеріали, які знадобилися для розробки даної програми представлені в табл. 6.5.

Таблиця 6.5 – Витратні матеріали

Матеріал	Кількість	Вартість, грн.
Папір формату А4	500 листів	40
DVD-RW диск	1 шт.	15
Тонер для принтера	1 шт.	120

Витрати на матеріали розраховуються за формулою 6.4:

$$M = \sum_{i=1}^n (Ц_i \cdot N_i \cdot (1 + K_{Т.З.}) - Ц_{i0} \cdot N_{i0}), \quad (6.4)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектувальні вироби, грн.;

$K_{Т.З.}$ – коефіцієнт, що враховує транспортно-заготівельні витрати (0,03 - 0,1);

Ц_i – ціна i -го найменування матеріалу, напівфабрикату, комплектувального, грн.;

N_i – потреба в i -му матеріалі, напівфабрикаті, комплектувальному;

Ц_{io} – ціна поворотних відходів i -го найменування матеріалу, грн.;

N_{io} – кількість зворотних відходів i -го найменування;

n – кількість найменувань матеріалів, напівфабрикатів, комплектувальних.

Розрахуємо витрати на матеріали:

$$M = (1 + 0,05) \cdot (40 + 15 + 120 + 50) = 236,25 \quad (6.5)$$

Разом витрати на матеріали склали 236,25 грн.

6.2.5 Витрати на спеціальне устаткування

У цю статтю входять витрати на оплату машинного часу ЕОМ. Витрати на оплату машинного часу ЕОМ для написання і відладки цієї програми визначають по формулі:

$$C_{mo} = P_{екс} \cdot t_{mo}, \quad (6.6)$$

де C_{mo} – витрати на оплату машинного часу, грн.;

$P_{екс}$ – експлуатаційні витрати на одну годину машинного часу, грн.;

t – машинний час ЕОМ для написання і відладки цього програмного продукту, ч.

Для розрахунку експлуатаційних витрат на одну годину машинного часу складемо кошторис витрат на зміст і експлуатацію цього устаткування.

Амортизаційні відрахування визначають по формулі:

$$A = \Phi_6 \cdot \frac{H_a}{100}, \quad (6.7)$$

де Φ_6 – балансова вартість обчислювальної техніки, грн.;

N_a – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки.

Балансова вартість обчислювальної техніки становить 5000 грн., а норма амортизаційних відрахувань – 25%. Таким чином амортизаційні відрахування становлять 1250 грн.

Витрати на електроенергію:

$$C_e = N_n \cdot \Phi_{ef} \cdot K_{зч} \cdot K_{зп} \cdot C_e, \quad (6.8)$$

де N_n – номінальна потужність ЕОМ, кВт;

Φ_{ef} – річний ефективний фонд часу роботи ЕОМ, ч.;

$K_{зч}$ – середній коефіцієнт завантаження за часом;

$K_{зп}$ – коефіцієнт завантаження по потужності;

C_e – ціна одного кВт·год електроенергії, грн.

$$C_e = 0,2 \cdot 1800 \cdot 0,9 \cdot 0,6 \cdot 1,6 = 311,04 \text{ грн} \quad (6.9)$$

Зарплата обслуговуючого персоналу розраховується за формулою 6.10:

$$ЗП_{обсл} = ФЗП_p \cdot (1 + K_{від}) \cdot \frac{t_{обсл}}{\Phi_{ef.обсл}}, \quad (6.10)$$

де $ФЗП_p$ – річний фонд зарплати обслуговуючих працівників, грн.;

$K_{від}$ – коефіцієнт відрахувань на соціальне страхування і до інших фондів, %;

$t_{обсл}$ – час впродовж року, необхідний на техобслуговування ЕОМ, год/рік.

$\Phi_{ef.обсл}$ – річний ефективний фонд часу обслуговуючого персоналу, год/рік.

Машину обслуговує один електронщик з основною заробітною платою 4000 грн./міс.

$$\Phi ЗП_2 = 4000 \cdot 12 = 48000 \text{ грн.}; \quad (6.11)$$

$$K_{\text{від}} = 0,385; \Phi_{\text{еф.обсл}} = 12 \text{ год/рік}; t_{\text{обсл}} = 1750 \text{ год}, \quad (6.12)$$

$$ЗП_{\text{обсл}} = 48000 \cdot (1 + 0,385) \cdot 12 / 1750 = 427,37 \text{ грн.} \quad (6.13)$$

Сума витрат по статті "Поточний ремонт устаткування" дорівнює 3% від балансової вартості устаткування, що дорівнює 150 грн.

Інші витрати складають 5% від суми попередніх:

$$(1250 + 427,37 + 150 + 236,25) \cdot 0,05 = 2063,62 \cdot 0,05 = 103,18 \text{ (грн)} \quad (6.14)$$

Зведемо отримані дані в табл. 6.6.

Таблиця 6.6 – Кошторис витрат на зміст і експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація устаткування	1250
Експлуатація устаткування	311,04
Зарплата обслуговуючого персоналу з відрахуваннями	427,37
Поточний ремонт устаткування	150
Інші витрати	103,18
Всього	2241,59

Експлуатаційні витрати на одну годину машинного періоду складають:

$$P_{\text{екс}} = 2241,59/1800 = 1,25 \text{ грн./ч.} \quad (6.15)$$

Розробка і відладка програмного продукту займає 176 годин машинного часу (30 днів по 8 робочих годин), тобто $t_{\text{мо}} = 176 \text{ г.}$

Витрати на оплату машинного часу склали:

$$C_{\text{мо}} = 1,25 \cdot 176 = 220 \text{ грн.} \quad (6.16)$$

6.2.6 Накладні витрати

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і так далі), зміст і експлуатацію будинків. Накладні витрати включаються у вартість розробки системи непрямым шляхом - у відсотках до основної заробітної плати розробників. В даному випадку накладні витрати складають 40% до основної заробітної плати розробників, що складає 2688 грн. Результати калькуляції кошторисної вартості робіт по розробці програми і ціна програмного продукту наведені у таблиці 6.7.

Таблиця 6.7 – Калькуляція кошторисної вартості робіт по розробці програми і ціна програмного продукту

№	Найменування статей витрат	Сума, грн.	Питома вага до підсумку, %
1	Основна заробітна плата	6720	55,3
2	Додаткова заробітна плата	672	5,5
3	Соціальні відрахування	1626,24	13,4
4	Витрати на матеріали	236,25	1,9
5	Витрати на оплату машинного часу	220	1,8
6	Накладні витрати	2688	22,1
7	Разом витрати	12162,49	100,00
8	Прибуток (30%)	3648,74	–
9	Всього	15811,23	–
10	НДС (20%)	3162,24	–
11	Всього	12648,99	–

6.3 Оцінка економічної ефективності

Планується продати не менше, ніж 10 систем, кожна з яких матиме вартість 10000 грн., тобто у сумі прибуток становитиме 100 000 грн. З врахуванням ризику у 50% це становитиме 50000 грн.

Оскільки надходження від продажів у 50000 грн. істотно вище, ніж вартість розробки, яка становить 12648,99 грн., то проект є економічно ефективним.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенціальних небезпек

Сукупність чинників виробничого середовища, що впливає на здоров'я і працездатність людини в процесі роботи, називається умовами праці. Організація і поліпшення умов праці на робочому місці є одним з важливих резервів продуктивності і ефективності праці.

При роботі користувача, що працює з ПЕОМ постійно або періодично діють наступні небезпечні і шкідливі чинники:

- можливість ураження електрострумом внаслідок небезпечного рівня напруги в електричному ланцюзі, тому що ПЕОМ є електроустановкою, тому у приміщенні можуть виникнути аварійні ситуації: замикання, загоряння проводки, поразка операторів електрострумом;

- виникнення на екрані монітора статистичних зарядів, в наслідок роботи ПЕОМ;

- підвищений рівень електромагнітних випромінювань внаслідок широкого спектру випромінювання від дисплея, який включає рентгенівську, ультрафіолетову і інфрачервону області, а так само широкий діапазон електромагнітних випромінювань інших частот;

- невідповідність ергономічних характеристик устаткування нормованим величинам внаслідок нераціональної організації робочого місця;

- нервово-психічне навантаження; робота оператора пов'язана зі сприйняттям зображення на екрані, необхідністю постійного спостереження за динамікою зображення машинописних, графічних робіт і інших операцій;

- негативний вплив випромінювань на органи зору, що пов'язані з перенапругою м'язового апарату ока внаслідок недостатньої або підвищеної контрастності зображення монітору, мерехтінням зображення, відблисками на моніторах, що може призвести до астенопії;

- незадовільні умови освітлення робочого місця, в наслідок не рівномірного розподілу освітлювальних приладів;

- невідповідність нормам параметрів мікроклімату. ПЕОМ є джерелами тепловиділень, що є причиною підвищення температури і зниження вологості повітря на робочому місці;

- небезпеки, пов'язані з підвищеним рівнем шуму, який створюється устаткуванням та людьми, що знаходяться у приміщенні, і які призводить до погіршення слуху;

- можливість загорянь внаслідок порушень правил пожежної безпеки, що може призвести до пожеж;

- неправильні дії персоналу під час НС внаслідок відсутності управління та паніки, що може призвести до травм та летальних наслідків.

Для створення сприятливих умов роботи з комп'ютером необхідна відповідність нормативно-правовим актам по охороні праці, які регламентують показники стану навколишнього середовища:

- достатню площу й об'єм приміщення, де встановлений комп'ютер;
- показники мікроклімату робочого приміщення (температура повітря, відносна вологість, рухливість повітряного середовища);

- параметри світлового середовища робочого місця й робочої поверхні (наявність природного й штучного освітлення, освітленість екрана монітора, освітленість горизонтальної поверхні стола, відсутність прямого й відбитого сліпучого відблиску джерел світла, відсутність пульсації освітленості);

- рівень шуму й вібрації робочих місць у межах норм;

- відсутність у повітрі робочої зони шкідливих речовин.

Так як тема атестаційної роботи магістра «Визначення та відстеження облич за допомогою ансамблів випадкових вирішуючих дерев», тому нижче розглядаються заходи щодо охорони праці, виробничої санітарії, гігієни праці й пожежної безпеки при експлуатації ЕОМ з візуальними дисплейними терміналами (ВДТ) у офісному приміщенні.

7.2 Заходи по забезпеченню безпеки

Забезпечення безпеки праці, організації оптимальних умов праці, а також захисту навколишнього середовища в умовах роботи з ПК досягається шляхом дотримання норм і проведення заходів, передбачених законодавчими актами України.

ПК підключаються в мережу з напругою 220 В и частотою 50 Гц. Згідно ПУЕ приміщення відноситься до категорії «Приміщення без підвищеної небезпеки». У приміщенні повинні бути відсутні відкриті струмоведучі частини з напругою більше 12В (послідовний інтерфейс RS-232C). При виконанні правил ПУЕ враження електричним струмом можливо тільки у випадку несправності апаратури й живильних кабелів [22].

Апаратура, що перебуває під небезпечною для життя напругою, припускає забезпечення електробезпечності. Всі металеві частини обчислювальної техніки повинні бути заземлені на третій провід спеціальних триконтактних розеток, призначених для включення ПК. Струмопровідні кабелі повинні бути розташовані під дерев'яною підлогою, зверху покритою лінолеумом, або у вигляді схованої проводки в стінах, таким чином, контакт із кабелем виключається. Над кожною розеткою повинен бути попереджувачий напис «220 В». Індикаторні лампи в блоках ЕОМ сигналізують про включення блоку в мережу.

Електрична ізоляція забезпечується конструкціями пристроїв і способами їхнього підключення. У процесі експлуатації необхідно періодично робити перевірку якості ізоляції, шляхом виміру її опору. Відповідно до ПУЕ в електроустановках до 1000 В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проведенням і землею становить не менш 0,5 МОм [22].

Відповідно до ПУЕ захисне заземлення застосовується у всіх електроустановках змінного струму під напругою 380 В і вище й постійного

струму 440 В і вище. У приміщеннях з підвищеною небезпекою заземлення використовується в електроустановках змінного струму напругою 42 В та постійного струму 110 В. Захисному заземленню підлягають корпуси трансформаторів, оболонки кабелів, металеві огороження електроустановок відповідно до ГОСТ 12.1.030-81 «Электробезопасность. Защитное заземление, зануление» [23].

Вирівнювання потенціалів досягаються шляхом устрою контурних заземлювачів. У офісних приміщеннях вирівнювання потенціалів здійснюється з'єднанням металевих стоек ЕОМ, корпусів апаратів світильників і іншого устаткування з усіма доступними для дотику металевими конструкціями будинку, спорудження, а також з контуром-шиною захисного заземлення.

Електричний поділ мережі припускає поділ мережі на окремі електрично не пов'язані між собою ділянки за допомогою розділових трансформаторів. Підвищення електробезпечності відбувається за рахунок зменшення ємності мережі й підвищення опору ізоляції фаз мережі щодо землі.

Подвійна ізоляція забезпечує застосування крім основної, робочої ізоляції струмоведучих частин ще одного шару додаткової ізоляції, що ізолює людину від металевих неструмоведучих частин, які можуть випадково виявитися під напругою. При застосуванні електроустаткування з подвійною ізоляцією не потрібно ні заземлення, ні занулення їхніх корпусів.

В офісних приміщеннях розрядні струми статичної електрики найчастіше виникають при дотику до кожного з елементів ЕОМ. Такі розряди небезпеки для людини не представляють, але крім неприємних відчуттів вони можуть привести до виходу з ладу ЕОМ. Для зниження величини виникаючих зарядів статичної електрики в офісному приміщенні покриття технологічних підлог варто виконувати з одношарового полівінілхлоридного антистатичного лінолеуму. Іншим методом захисту є нейтралізація заряду статичної електрики іонізованим газом. До загальних мір захисту від статичної електрики в офісі можна віднести загальне й місцеве зволоження повітря.

Внаслідок роботи за персональним комп'ютером на фізіологію людини негативно впливають електромагнітні випромінювання. Щоб зменшити наслідки впливу на людину та знизити негативні показники у робочій зоні до допустимих значень, згідно з ГОСТ 12.2.007.0–75 «Изделия электротехнические. Общие требования безопасности», вироби, які створюють електромагнітні поля, повинні мати захисні елементи. Вимоги до захисних елементів повинні бути вказані в стандартах та технічних умовах на конкретні види виробів. Згідно з НПАОП 0.00–1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин» [24] та ДСанПіН 3.3.2.007–98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ЕОМ» [25] рекомендовано використовувати монітор типу Samsung SyncMaster 959 NF який не потребує додаткових захисних елементів при умові, що монітор працює в режимі 120 Гц.

У розділі 4 ДСанПіН 3.3.2.007-98 передбачені гігієнічні вимоги до організації устаткування робочих місць ВДТ ЕОМ і ПЭВМ.

Конструкція робочого стола повинна відповідати сучасним вимогам ергономіки й забезпечувати оптимальне розміщення на робочій поверхні використовуваного устаткування (дисплея, клавіатури, принтера) і документів. Висота робочої поверхні робочого стола із ВДТ повинні регулюватися в межах 680...800 мм, а ширина й глибина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600...1400 мм, глибина - 800..1000 мм). Приймаємо робочу поверхню з розмірами 1400x800. Щоб уникнути відблисків на екрані вікон доцільно розмістити робоче місце торцем до вікна не ближче чим на 1 м. Площа дозволяє нам розмістити додаткові меблі для зберігання документів, магнітних дисків, полками, стелажми, тумбами й т.п.

Велике значення надається ергономічним та естетичним вимогам. Цього домоглися за допомогою колірної рішення приміщення й устаткування з урахуванням вимог ГОСТ 17.2.032-78 «Рабочее место при выполнении работ сидя» – у приміщенні стіни пофарбовані фарбами пастельних тонів.

Відповідно до вимог розділу «Мінімальні вимоги з охорони праці», директиви ЄС 90/270/ЄЕС та НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» виконано такі п'ять основних вимог до моніторів, які жорстко регламентують безпечні умови роботи і захист здоров'я осіб, що працюють з комп'ютерами:

- символи на екрані чіткі і добре розрізняються;
- зображення позбавлене блимання;
- яскравість та / або контрастність легко регулюються;
- екрани вільні від відблисків і відбиття;
- випромінювання знижені до надзвичайно малих рівнів.

Під час організації робочого простору необхідно враховувати індивідуальні антропометричні параметри користувача з відповідними допусками на можливі зміни робочих поз та потребу у переміщеннях.

Площа на один працівника регламентується СНиП 2.09.04-87 «Административные и бытовые здания» і становить не менше 6 м². Значить у приміщенні довжиною $a = 5$ м, шириною $b = 4$ м, висотою $h = 3,3$ м загальною площею $S = 20$ м² можна розмістити не більше 3-х робочих місць. Крім площі приміщення береться в увагу й обсяг приміщення, що становить не менше 20м³ на одне робоче місце, що передбачено ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ЕОМ». Наше приміщення має об'єм в 66 м³, а це означає, що в нашому приміщенні можна розмістити три робочих місця із ВДТ ЕОМ.

Приміщення розташоване на першому поверсі виробничого корпусу, що припустимо й відповідає перерахованим вище правилам і нормам.

Серед користувачів ПК виявлено новий тип захворювання - синдром комп'ютерного стресу, який проявляється головним болем, запаленням слизової оболонки очей, підвищеною дратівливістю, млявістю і депресією.

Згідно з НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин» необхідно комплексно вирішувати проблеми ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку.

По-перше, для збереження здоров'я працівників, необхідно дотримуватися всіх вимог до приміщень, де відбувається робота з комп'ютерами.

По-друге, необхідно подбати про правильне розміщення робочих місць як щодо безпосередньо кімнати, так і щодо всієї будівлі підприємства.

По-третє, слід забезпечити робітників сучасною комп'ютерною меблюю, яка була б зручна у використанні і не доставляла почуття дискомфорту, була пристосована до особливостей людського організму.

Необхідно прагнути забезпечення всіх працівників сучасною комп'ютерною технікою (рідкокристалічні дисплеї, які можна назвати майже «зеленими» пристроями, безпечними для здоров'я людей; використання різноманітних сучасних ергономічних клавіатур). В іншому випадку необхідно забезпечити техніку спеціальними захисними пристроями (наприклад, захисний фільтр).

Крім того, повинен бути встановлений правильний режим праці та відпочинку

Час перерв протягом робочого дня для 8-годинної зміни розподіляється наступним чином:

– для I категорії (до 20 000 тис. друкованих знаків) – 2 перерви по 15 хв. Через 2 години після початку зміни і після обідньої перерви;

– для II категорії (до 40 000 тис. друкованих знаків) – через 2 години після початку зміни і через 1,5-2 години після обідньої перерви по 15 хв. або 10 хв., через кожен годину роботи;

– для III категорії (до 60 000 тис. друкованих знаків) – через 1,5-2 години після початку зміни і через 1,5-2 години після обідньої перерви по 20 хв., або по 15 хв. через кожен годину.

При 12-годинній зміні перерви в перші 8 годин такі ж, як і при 8-годинній зміні, протягом останніх 4 годин, незалежно від категорії і виду робіт, – кожен годину по 15 хв.

Не рекомендується працювати за ПК більше 2 годин підряд без перерви, У процесі роботи по можливості, щоб зменшити негативний вплив монотонності, слід змінювати тип і зміст діяльності.

Дотримання цих заходів призводить до того, що підвищується продуктивність праці, скорочується кількість професійних захворювань, крім це працівники знаходяться завжди в гарному настрої і не скаржаться на погане самопочуття.

Заходи мінімізації негативного впливу випромінювань ВДТ на органи зору. Відеотермінальний пристрій повинен відповідати наступним технічним вимогам:

- яскравість свічення екрана повинна бути не менше $100 \text{ кд} / \text{м}^2$;
- мінімальний розмір світної точки – не більше 0,4 мм для монохромного дисплея і не більше 0,6 мм – для кольорового дисплея;
- контрастність зображення знаку – не менше 0,8;
- частота регенерації зображення при роботі з позитивним контрастом у режимі обробки тексту – не менше 72 Гц;
- кількість точок на рядку – не менше 640;
- низькочастотне тремтіння зображення в діапазоні $0,05 \cdot 10 \text{ Гц}$ повинно знаходитися в межах 0,1 мм;
- екран повинен мати покриття антивідблиску;
- розмір екрану повинен бути не менше 31 см по діагоналі, а висота символів на екрані – не менше 3,8 мм, при цьому відстань від очей оператора до екрану повинна бути в межах 40*80 см.

7.3 Заходи з забезпечення виробничої санітарії та гігієни праці

Для зниження шкідливого впливу комп'ютера на людину повинні дотримуватися певні вимоги до умов роботи, до робочого місця й крім того, дотримуватися режим роботи на комп'ютері.

Об'ємно-планувальні рішення будинків і приміщень для роботи з візуальними дисплейними терміналами (ВДТ) ЕОМ повинні відповідати вимогам ДСанПіН 3.3.2.007-98 «Державним санітарним правилам і нормам роботи з візуальними дисплейними терміналами електронно-обчислювальних машин». Розміщення робочих місць із ВДТ ЕОМ у підвальних приміщеннях, на цокольних поверхах заборонено [24].

В офісному приміщенні дотримуються норми ДБН В.2.5-28: 2018 «Природне і штучне освітлення» відносно штучного освітлення: освітленість робочих місць у горизонтальній площині на рівні 0,8 м від підлоги не менш 400 лк, вертикальна освітленість у площині екранів дисплея – не менш 300 лк, коефіцієнт пульсації освітленості – не більше 10%. Забезпечуються таке освітлення розташованими на стелі світильниками ЛСП з люмінесцентними лампами типу ЛБ-60 [25].

Віконні прорізи приміщень для роботи із ВДТ обладнані регульованими пристроями.

Оптимальні норми температури повітря в робочій зоні виробничих приміщень у холодний і перехідний періоди року при II категорії робіт згідно ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень – 22...24оС, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек. У теплий період року температура – 23...25оС, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек [26].

Для забезпечення вищевказаних параметрів передбачене застосування вентиляції – природної (аерації й провітрювання), механічної (загальобмінної, місцевої приточної і витяжної), кондиціювання повітря.

У приміщенні забезпечений приплив свіжого повітря, кількість якого визначається техніко-економічним розрахунком і вибором схеми системи вентиляції. Мінімальна витрата повітря визначається з розрахунку 50...60 м³/год на одну людину, але не менш дворазового повітрообміну в годину.

Згідно з ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» зниження шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму проникаючого ззовні, є дуже важливим завданням. Зниження шуму в джерелі виникнення забезпечується застосуванням пружних прокладок між основою машини, приладу й опорною поверхнею. Як прокладки використовуються гума, повсть, пробка, різної конструкції амортизатори. Під настільні апарати можна підкладати м'які килимки із синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, - прокладки з м'якої гуми, повсті, товщиною 6 - 8 мм.

Звукоізоляція огорожувальних конструкцій приміщення із ВДТ забезпечує параметри шуму, які відповідають вимогам ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» [27].

7.4 Заходи безпеки у надзвичайних ситуаціях

7.4.1 Заходи з пожежної безпеки

Заходи щодо протипожежного захисту розділяються на організаційні, експлуатаційні, технічні й режимні (спеціальні).

Організаційні заходи: навчання робітників та службовців правилам пожежної безпеки, організація пожежної охорони, проведення бесід, лекцій, видання необхідних інструкцій, плакатів і т.п.

Технічні заходи передбачають дотримання протипожежних правил і норм при устрої систем опалення, вентиляції, кондиціонуванні повітря, блисковкозахисті при спорудженні будинків, установці технологічного устаткування й ін.

Експлуатаційні заходи передбачають правильну експлуатацію систем опалення, вентиляції й кондиціювання повітря, блисковкозахисту технологічних машин і устаткування, правильне утримання будинків і територій і т.п.

Режимні заходи передбачають заборону або обмеження застосування відкритого вогню в вогнебезпечних місцях, паління в невстановлених місцях, обов'язкове дотримання норм і правил при роботі з вогнебезпечними й вибухонебезпечними речовинами.

Категорія приміщення за вибухопожежобезпекою відноситься до категорії Д.

Одним з найбільш важливих завдань пожежної профілактики є захист будівельних конструкцій від руйнування й забезпечення їхньої достатньої міцності в умовах впливу високих температур при пожежі. Залежно від меж вогнестійкості будівельних конструкцій ДБН В.2.1.1-2002 «Протипожежні норми» установлені вісім ступенів вогнестійкості будинків: I, II, III, IIIa, IIIб, IV, IVa, V [28].

Враховуючи високу вартість електронного обладнання, а також категорію їх пожежної безпеки, будівлі повинні мати ступінь вогнестійкості не нижче III. Тому для виготовлення будівельних конструкцій використовують, як правило, цегла, залізобетон, скло, метал та інші негорючі матеріали. Застосування дерева - обмежена, а в разі його використання необхідно просочувати вогнезахисним составом - антипіреном або покривати вапняно-цементною, азбестоцементною, гіпсовою штукатуркою. При виготовленні будівельних конструкцій слід застосовувати спеціальні фарби, які в умовах звичайної експлуатації захищають метал від корозії, а при пожежі спучуються, що призводить до збільшення їх термічного опору і до збільшення межі вогнестійкості забарвлених конструкцій. Для запобігання розповсюдження вогню під час пожежі з однієї частини будівлі в іншу влаштовують протипожежні перешкоди у вигляді протипожежних стін, перегородок, перекриття, тощо.

Пожежна профілактика – це комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, запобігання пожежі, обмеження його розповсюдження, а також на створення умов для успішного гасіння пожежі.

Протидимний захист будівель, спрямований на незадимлюваність евакуаційних шляхів, окремих приміщень і видалення продуктів горіння в певному напрямку, є першорядним завданням пожежної профілактики.

Для виявлення початкової стадії загоряння, швидкого й точного оповіщення служби пожежної охорони про час і місце виникнення пожежі встановлені системи автоматичної пожежної сигналізації (АПС). Вони можуть самостійно пускати в хід установки пожежогасіння, коли пожежа не досягла великих розмірів. Системи АПС складаються з пожежних сповіщувачів, ліній зв'язку й прийомних пультів (станцій).

Для подачі сигналу про пожежу встановлені пожежні сповіщувачі. Залежно від контролюваного фактору, вони діляться на теплові, димові, світлові, комбіновані.

До первинних засобів гасіння пожеж, призначених для локалізації невеликих загорянь, належать пожежні цівки, внутрішні пожежні водопроводи, вогнегасники, сухий пісок, азбестові ковдри й т.п.

Для гасіння вогню у випадку пожежі в приміщенні площею 20 м² повинно бути 2 вуглекислотних вогнегасники ВВ-2 (ВВ-5 – 1 штука), можливо також використання порошкових вогнегасників ВП-3.

На випадок пожежі передбачені шляхи евакуації працюючих: проходи, проїзди, евакуаційні виходи.

Евакуаційні виходи розташовуються розосереджено, а їх кількість - не менше двох із будівлі. При кількості працюючих у зміні до 50 чоловік передбачений другий (запасний вихід) на зовнішні сходи.

Коридори й проходи, призначені для евакуації, мають найменшу довжину й мінімальну кількість поворотів. На всьому протязі проходу немає порогів або проміжних щаблів.

Важливу роль у забезпеченні безпечного виходу людей грає протидимовий захист евакуаційних ходів. У будинках висотою до 9 поверхів незадимлюваність сходових кліток на час евакуації досягається їхньою ізоляцією від підвалів, горищ і поверхів. Для цього влаштовані відособлені входи в підвали, вхід на сходову клітку з поверхів здійснюють через тамбур – шлюз із підпором повітря, відокремлюють горища від сходових кліток перекриттями з негорючих матеріалів.

У таблиці 7.1 наведені відстані від найбільш віддаленого робочого місця до найближчого евакуаційного виходу.

Таблиця 7.1 – Відстані від найбільш віддаленого робочого місця до найближчого евакуаційного виходу

Об'єм приміщення, тис. м ³	Ступінь вогнестійкості будівлі	Відстань, м, при щільності людського потоку в загальному проході, люд/м ²		
		до 1	від 1 до 3	від 3 до 5
Незалежно від об'єму	I, II, III	Не обмежується		
	IV	160	95	65
	V	120	70	50

Для відводу надлишкової теплоти служать системи вентиляції й кондиціювання повітря. Однак вони становлять додаткову пожежну небезпеку, тому що, з одного боку, вони забезпечують подачу кисню - окислювача в усі приміщення, а з іншого боку - при виникненні пожежі поширюють вогонь і продукти горіння по всіх приміщеннях і установках, з якими зв'язані повітроводи.

Для тепло- і звукобірної ізоляції систем вентиляції застосовують такі негорючі матеріали, як мати з мінеральної вати, скловолокна, стиліту.

Напруга до електроустановок подається по кабельних лініях, які становлять особливу пожежну небезпеку.

Одним з найбільш важливих завдань пожежної профілактики є захист будівельних конструкцій від руйнування й забезпечення їхньої достатньої міцності в умовах впливу високих температур при пожежі.

Підпільні простори під знімними підлогами розділяють неспаленими перегородками з межею вогнестійкості не менш 0.75 г на відсіки площею не більше 250 м², комунікації через які прокладають у спеціальних обоймах із застосуванням негорючих ущільнюючих матеріалів [29].

7.4.2 Заходи з цивільної оборони

Порядок проведення рятувальних робіт та інших невідкладних робіт при аваріях на АЕС.

Сутність рятувальних та інших невідкладних робіт - це усунення безпосередньої загрози життю та здоров'ю людей, відновлення життєзабезпечення населення, запобігання або значне зменшення матеріальних збитків. Рятувальні та інші невідкладні роботи включають також усунення пошкоджень, які заважають проведенню рятувальних робіт, створення умов для наступного проведення відновлювальних робіт. РІНР поділяють на рятувальні роботи і невідкладні роботи.

До рятувальних робіт відносяться:

- розвідка маршруту руху сил, визначення обсягу та ступеня руйнувань, розмірів зон зараження, швидкості і напрямку розповсюдження зараженої хмари чи пожежі;
- локалізація та гасіння пожеж на маршруті руху сил та ділянках робіт;
- визначення об'єктів і населених пунктів, яким безпосередньо загрожує небезпека;
- визначення потрібного угруповання сил і засобів запобігання і локалізації небезпеки;
- пошук уражених та звільнення їх з-під завалів, пошкоджених та палаючих будинків, із загазованих та задимлених приміщень;

- розкриття завалених захисних споруд та рятування з них людей;
- надання потерпілим першої допомоги та евакуація їх (при необхідності) у лікувальні заклади;
- вивіз або вивід населення із небезпечних місць у безпечні райони;
- організація комендантської служби, охорона матеріальних цінностей і громадського порядку;
- відновлення життєздатності населених пунктів і об'єктів;
- пошук, розпізнавання і поховання загиблих;
- санітарна обробка уражених;
- знезараження одягу, взуття, засобів індивідуального захисту, територій, споруд, а також техніки;
- соціально-психологічна реабілітація населення.

До невідкладних робіт відносяться:

- прокладання колонних шляхів та улаштування проїздів (проходів) у завалах та на зараженій території;
- локалізація аварій на водопровідних, енергетичних, газових і технологічних мережах;
- ремонт та тимчасове відновлення роботи комунально-енергетичних систем і мереж зв'язку для забезпечення рятувальних робіт;
- зміцнення або руйнування конструкцій, які загрожують обвалом і безпечному веденню робіт.

Рятувальні та інші невідкладні роботи здійснюються у три етапи:

На першому етапі вирішуються завдання:

- щодо екстреного захисту населення;
- з запобігання-розвитку чи зменшення впливу наслідків;
- з підготовки до виконання РІНР.

Основними заходами щодо екстреного захисту населення є:

- оповіщення про небезпеку;
- використання засобів захисту;
- додержання режимів поведінки;

- евакуація з небезпечних у безпечні райони;
- здійснення санітарно-гігієнічної, протиепідемічної профілактики і надання медичної допомоги;

- локалізація аварій;
- зупинка чи зміна технологічного процесу виробництва;
- попередження (запобігання) і гасіння пожеж.

На другому етапі проводяться:

- пошук потерпілих;
- витягання потерпілих з-під завалів, з палаючих будинків, пошкоджених транспортних засобів;

- евакуація людей із-зони лиха, аварії, осередку ураження;
- надання медичної допомоги;
- санітарна обробка людей;
- знезараження одягу, майна, техніки, території;
- проведення інших невідкладних робіт, ЩО сприяють і забезпечують здійснення рятувальних робіт.

На третьому етапі вирішуються завдання щодо забезпечення життєдіяльності населення у районах, які потерпіли від наслідків НС:

- відновлення чи будівництво житла;
- відновлення енерго-, тепло-, водо-, газопостачання, ліній зв'язку;
- організація медичного обслуговування;
- забезпечення продовольством і предметами першої необхідності;
- знезараження харчів, води, фуражу, техніки, майна, території;
- соціально-психологічна реабілітація;
- відшкодування збитків;
- знезараження майна, території, техніки.

Відновлювальні роботи здійснюють спеціально створені підрозділи (бригади). Залежно від рівня надзвичайної ситуації (загальнодержавного, регіонального, місцевого чи об'єктового) для проведення РІНР залучаються сили і засоби ЦО центрального, регіонального або об'єктового підпорядкування.

При радіаційних аваріях на АЕС викинуті із реактора радіонукліди піднімаються в атмосферу і переносяться у вигляді аерозолей на значну відстань. Потім вони випадають разом з пилом і дощем на місцевість, утворюючи обширні зони радіоактивного забруднення, які є небезпечними для людей і навколишнього середовища. Ступінь радіаційної небезпеки для населення визначається кількістю і складом радіонуклідів, викинутих у зовнішнє середовище, відстанню від місця аварії до населеного пункту, метеоумов і пори року в час аварії.

Організація і проведення РІНР при аварії на АЕС полягає у виконанні заходів, до яких відносяться:

- оповіщення населення про, аварію і постійне його інформування про наявну обстановку та порядок дій в даних умовах;
- використання засобів колективного і індивідуального захисту;
- організація дозиметричного контролю
- проведення йодної профілактики населення, що опинилося в зоні радіоактивного зараження;
- введення обмеженого перебування населення на відкритій місцевості (режими радіаційного захисту);
- здійснення евакуації населення (за розпорядженням Уряду) та інші заходи.

Після евакуації населення приступають до дезактивації території і техніки [30]-[31].

ВИСНОВКИ

В ході виконання дипломної роботи було досліджено та розроблено модель розпізнавання тексту з зображення для автоматизованої системи фіксування результатів кіберспортивних з використанням сучасних технологій, таких як мова програмування Python, та додаткових бібліотек – Tesseract, OpenCV. Вебінтерфейс з рейтинговою таблицею був розроблений з використанням фреймворку Laravel, на фреймворку Bootstrap.

Розроблена система дозволяє проводити кіберспортивні змагання онлайн, а також автоматично фіксує результати.

В роботі були реалізовані такі функціональні вимоги:

- інтерфейс проведення турніру через месенджер Discord;
- автоматичне зчитування результатів с зображень учасників;
- повноцінна турнірна таблиця на сайті, яка оновлюється онлайн;
- повний контроль за ходом турніру через адміністративні команди;
- можливість дискваліфікації нечесних гравців;
- зручний перегляд статистики та формування турнірної таблиці;
- можливість розширити функціонал розроблюваної системи;
- максимальна швидкість та простота у роботі з системою;
- зберігання та обробка даних учасників у захищеному вигляді.

За результатами роботи можна зробити висновок, що розроблена модель розпізнавання тексту з зображення в автоматизованої системи фіксування результатів кіберспортивних змагань є ефективним рішенням, тому що вона має високу швидкість та точність розпізнавання тексту з зображення, також система може легко масштабуватися та доповнюватися новим функціоналом у майбутньому.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Information about the possibilities of the service toornament.com [Electronic resource]. – Access mode: https://www.toornament.com/en_GB/why-toornament/tournament-engine.
2. Інформація о возможностях сервиса weplay.tv [Электронный ресурс]. – Режим доступа: <https://weplay.tv/ru/tournaments>.
3. Інформація о возможностях сервиса game.tv [Электронный ресурс]. – Режим доступа: <https://www.game.tv/ru/how-to-use-tourney>.
4. Інформація о возможностях сервиса challonge.com [Электронный ресурс]. – Режим доступа: <https://challonge.com/ru>.
5. Інформація о возможностях сервиса cybernetica.agency [Электронный ресурс]. – Режим доступа: <https://cybernetica.agency/ru/>.
6. OpenCV vs VXL vs LTI: Performance Test - AI Shack [Electronic resource]. – Access mode: <http://www.aishack.in/tutorials/opencv-vs-vxl-vs-lti-performance-test/>.
7. Information about Tesseract [Electronic resource]. – Access mode: <https://tesseract-ocr.github.io/tessdoc/FAQ.html>.
8. OpenCV Library [Electronic resource]. – Access mode: <https://opencv.org/>.
9. Anyline Library [Electronic resource]. – Access mode: <https://anyline.com/>.
10. Framework AForge.NET [Electronic resource]. – Access mode: <http://www.aforgenet.com/framework/>.
11. Применение библиотек AForge.NET и ее расширение Accord.NET Framework при распознавании лиц в реальном времени [Электронный ресурс]. – Режим доступа: <https://moluch.ru/archive/154/43602/>.
12. VXL – C++ Library for computer vision [Electronic resource]. – Access mode: <http://vxl.sourceforge.net/>.

13. What is VXL ? Vision Something Libraries A collection of Computer Vision libraries Open Source, grass roots effort, 53 developers [Electronic resource]. – Access mode: <https://slideplayer.com/slide/7433547>.
14. LTI-Lib [Electronic resource]. – Access mode: <http://ltilib.sourceforge.net/doc/homepage/index.shtml>.
15. Text Recognition API [Electronic resource]. – Access mode: <https://developers.google.com/vision/android/text-overview>.
16. Описание PyCharm [Електронний ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/PyCharm>.
17. Опис мови програмування Python [Електронний ресурс]. – Режим доступа: <https://uk.wikipedia.org/wiki/Python>.
18. BytesIO [Electronic resource]. – Access mode: <https://docs.python.org/3/library/io.html>.
19. Шпаргалка по OpenCV — Python [Електронний ресурс]. – Режим доступа: <https://tproger.ru/translations/opencv-python-guide/>.
20. PyParsing [Електронний ресурс]. – Режим доступа: <http://xgu.ru/wiki/pyparsing>.
21. Кузнецов, М. Самоучитель MySQL 5 [Текст] / М. В. Кузнецов, И. В. Симдянов – СПб. : БХВ-Петербург, 2006. – 560 с.
22. Правила улаштування електроустановок [Текст] : Нормативне виробничо-практичне видання : вид. 5-те, перероблене й доповнене : затв. М-вом енергетики та вугільної промисловості України 20.06.14 : введення в дію з 20.11.14. – Х. : Міненерговугілля України, 2014. – 793 с. ; 24 см. – 5000 прим.
23. Електробезпека. Захисне заземлення, занулення: ГОСТ 12.1.030-81. – [введ. 1982-07-01]. – М.: Изд-во стандартов, 1982. – 10 с.
24. Правила охорони праці під час експлуатації електронно-обчислювальних машин: НПАОП 0.00-1.28-10. Наказ Держгірпромнагляду від 26.03.2010 р. №65. – К.: Держстандарт України, 2008. – 22 с.

25. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПіН 3.3.2.007-98. – К.: Держстандарт України, 1998. – 28 с.
26. Природне и штучне освітлення: ДБН В.2.5-28-2006. – К.: Держстандарт України, 2006. – 96 с.
27. Москальова В.М. Основи охорони праці [Текст] / В.М. Москальова. – Рівне: НУВГП, 2006. – 666 с.
28. Протипожежні норми: ДБН В.2.1.1-2002. – К.: Держстандарт України, 2002. – 42 с.
29. Бедрій Я.І. Охорона праці та пожежна безпека: навчальний посібник для студентів вищих навчальних закладів та інженерів-практиків [Текст] / Я.І. Бедрій. – Тернопіль: Навчальна книга – Богдан, 2013. – 184 с.
30. Михайлюк В.О. Цивільна безпека: навч. посібник [Текст] / В.О. Михайлюк, Б.Д. Халмурадов. – К: Центр учбової літератури, 2008. – 158 с.
31. Бедрій Я.І. Цивільний захист України: навч. посібник для студентів вищих навчальних закладів [Текст] / Я. Бедрій, В. Малов. – Тернопіль: Навчальна книга - Богдан, 2014. – 392 с.

ДОДАТОК А
Технічне завдання

A.1 Вступ

На даний час існує проблема з невеликою кількістю готових та універсальних систем для проведення кіберспортивних змагань та можливістю автоматично фіксувати результати, тому для вирішення цього питання було прийнято рішення розробити свою систему. Система надає можливість проводити турніри та автоматично фіксувати результати з зображень, які надсилають учасники турніру

A.2 Підстава для розроблення

Підставою для розробки [7] є завдання на дипломний проект на тему «Дослідження та програмна реалізація автоматизованої системи фіксування результатів кіберспортивних змагань», затверджений наказом №301 від 03 листопада 2020 року.

A.3 Призначення розробки

Сервіс призначений для проведення онлайн турнірів з автоматичним фіксуванням результатів. Експлуатаційне призначення системи це використання її людьми, які є учасниками кіберспортивних змагань.

A.4 Основні вимоги до програмного забезпечення

A.4.1 Вимоги до функціональних характеристик

Функціональні вимоги користувача:

- відправка зображень для фіксування результатів;
- перегляд онлайн статистики турніру;
- онлайн реєстрація на турнір;

- отримання рейтингової таблиці турніру;
- перегляд інших учасників турніру.

Функціональні вимоги адміністратора:

- створення турніру;
- відкриття реєстрації на турнір;
- ручне редагування результатів;
- повний контроль над роботою турніру.

A.4.2 Вимоги до надійності

Сервіс повинен забезпечувати обробку збійних ситуацій.

A.4.3 Умови експлуатації

Експлуатаційними умовами є:

- Інтернет;
- сервер (який характеристики);
- ПК;
- месенджер Discord.

A.4.4 Вимоги до складу та параметрів технічних засобів

Програмний продукт є кросплатформним, тож може використовуватись як на стаціонарних комп'ютерах та ноутбуках, так і на портативних пристроях (планшетах, смартфонах, тощо.) Пристрій повинен мати встановлену операційну систему: Windows 7 та новіше, Mac OS X 12 та новіше, Android 4.4+, iOS 11 та новіше. На операційній системі має бути встановлений месенджер Discord актуальної версії.

A.4.5 Вимоги до маркування та упакування

Система може бути записана на USB-флеш накопичувач, на якому будуть знаходитись всі необхідні файли з кодом та документацією.

A.4.6 Вимоги до транспортування та збереження

Вимоги до транспортування й зберігання аналогічні вимогам, пропонованим до носія, на якому записана програма.

A.4.7 Вимоги до програмної документації

Сервіс повинен супроводжуватись інструкцією для користувача, специфікацією, керівництвом програміста.

A.5 Стадії та етапи розробки

A.5.1 Етап розробки Аналіз вимог

Це етап на якому збирається вся необхідна інформація для розробки сервісу.

A.5.2 Етап розробки Проектування

На цьому етапі визначається інтерфейс сервісу та проектується БД.

A.5.3 Етап розробки Програмування і верстка

Це основний етап розробки на якому відбувається верстка сторінок системи, проходять перші тести на реальних даних, код може дороблятися, фіксуються помилки розроблюваної системи.

A.5.4 Етап розробки Запуск і супровід

Це завершаючий етап розробки на якому формується на відправлення замовникові вся документація, на сервіс заливається необхідний контент, вибирається певна платформа, та весь проект відсилається замовникові для тестування.

A.5.5 Етап розробки Закриття проекту

На цьому етапі замовник повідомляє після своїх тестів на своєму обладнанні, що все працює та він згоден підписати акт передачі проекту та отримує від замовника ПЗ та документацію.

A.6 Порядок контролю та приймання системи

Замовник повинен провести тест на власному технічному забезпеченні. У разі знайдення помилок чи не відповідності ТЗ виконавець зобов'язується виправити всі помилки у певний термін. Якщо тестування було успішним замовник підписує акт передачі проекту та отримує від замовника ПЗ та документацію.

ДОДАТОК Б
Текст програми

Б.1 Файл Main.py

```

import discord
import asyncio
import requests
import urllib.request
import os
import datetime
import pymysql

DISCORD_BOT_TOKEN =
'Nzc4NjY3NjIzMTg0NTk3MDEy.HaWGIVOrRBNl3qui_HGcgJAZEnc'

client = discord.Client()

@client.event
async def on_ready():
    print('Logged in as')
    print(client.user.name)
    print(client.user.id)
    print('-----')
    connection = pymysql.connect(host='192.168.5.134',
                                user='root',
                                password='1234',
                                db='simplehr',
                                charset='utf8mb4',

cursorclass=pymysql.cursors.DictCursor)
    print('Connected to db')
    print('-----')

@client.event
async def on_message(message):
    if(message.attachments == "!startskirmish")
    countround = 0

```

```

print('The tournament has started')
print('Type !reg to register in tournament')

if(message.attachments == "!reg")
    playername = message.senderName
    sql1 = "Insert into players (PlayerName) values
(playername)"
    connection.execute(sql1)

if(message.attachments == "!unreg")
    playername = message.senderName
    sql1 = "Delete FROM players WHERE `PlayerName` ==
playername"
    connection.execute(sql1)

if(message.attachments == "!startround")
    countround++
    print("Attention everyone! Skirmish starting NOW!")

    if(message.attachments == img)
        plexs = imgToRes(img.url)
        sql1 = "Insert into stats (PlayerName,
`countround`, TotalKills, TotalScore) values (playername, plexs,
plexs[exiles], plexs[exiles]+plexs[place])"

        if(message.attachments == "!endskirmish")
            print("Skirmish has been ended! Thank you for
participation! Check stats at: %sitelink%")

def imgToRes(url):
    try:
        img = requests.get(url)
    except:
        return 'Unrecognized','Unrecognized'
    img = BytesIO(img.content)

```

```

img = Image.open(img)
img = cv2.cvtColor(np.array(img), cv2.COLOR_RGB2BGR)
blur = cv2.GaussianBlur(gray, (5,5), 0)
thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY_INV
+ cv2.THRESH_OTSU) [1]

text = pytesseract.image_to_string(img)
print(text)
try:
    place
re.search(r'ace:\s?\d{1,2}',text) [0].split(':')[1].strip()
except:
    place = 'Unrecognized'
try:
    exiles
re.search(r'les:\s?\d{1,2}',text) [0].split(':')[1].strip()
except:
    exiles = 'Unrecognized'
return place,exiles

client.run(DISCORD_BOT_TOKEN)

```

Б.2 Файл tournament.phtml

```

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1">

```

```

<title>Турниры Spellbreak на русском | Русскоязычное
сообщество по игре Спеллбрейк проводит турниры с призовым
фондом.</title>

<meta name="description" content="Проведение турниров по
игре Spellbreak, Tournaments Spellbreak game."/>

<meta name="keywords" content="турниры spellbreak,
проведение турниров, крутые турниры."/>

<link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css
/materialize.min.css">
<link rel="stylesheet"
href="<?=Theme::getURL() ?>/css/style.css">

</head>
<body>
<div id="main-block" class="classes">
<div id="header">
<div class="container">
<nav>
<div class="nav-wrapper">
<a href="#" data-target="mobile-
demo" class="sidenav-trigger"><i class="material-
icons">menu</i></a>
<a href="/" class="brand-logo
center">

</a>
<ul id="nav-mobile" class="left
hide-on-med-and-down">

```

```

<li><div class="buttons-
coll">
<button class="custom-
btn btn-3"><span><a href="/">Главная</a></span></button>
</div></li>
<li><div class="buttons-
coll">
<button class="custom-
btn btn-3"><span><a href="/news">Новости</a></span></button>
</div></li>
<li><div class="buttons-
coll">
<button class="custom-
btn btn-3"><span><a
href="/tournaments">Турниры</a></span></button>
</div></li>
<li><div class="buttons-
coll">
<button class="custom-
btn btn-3"><span><a href="/media">Медиа</a></span></button>
</div></li>
</ul>
<ul id="nav-mobile" class="right
hide-on-med-and-down">
<li><div class="buttons-
coll">
<button class="custom-
btn btn-3"><span><a href="/guides">Гайды</a></span></button>
</div></li>
<li><div class="buttons-
coll">
<button class="custom-
btn btn-3"><span><a href="/classes">Классы</a></span></button>
</div></li>

```

```

<li><div class="buttons-
coll">
<button class="custom-
btn btn-3"><span><a href="/items">Предметы</a></span></button>
</div></li>
<li><div class="buttons-
coll">
<button class="custom-
btn btn-3"><span><a href="/contacts">Контакты</a></span></button>
</div></li>
</ul>
</div>
</nav>
<ul class="sidenav" id="mobile-demo">
<li><a href="/">Главная</a></li>
<li><a href="/news">Новости</a></li>
<li><a
href="/tournaments">Турниры</a></li>
<li><a href="/media">Медиа</a></li>
<li><a href="/guides">Гайды</a></li>
<li><a
href="/classes">Классы</a></li>
<li><a
href="/items">Предметы</a></li>
<li><a
href="/contacts">Контакты</a></li>
</ul>
</div>
</div>
<div id="class-block">
<div class="container">
<div class="row">
<div class="col s12 topinfo">
<h1>Проведение турниров по
Spellbreak</h1>

```

```

        </div>
    </div>
    <div class="row">
        <table class="highlight centered
responsive-table" style="background-color: rgba(255,255,255,0.5)">
            <thead>
                <tr>
                    <th></th>
                    <th>Название турнира</th>
                    <th>Дата проведения</th>
                    <th>Количество игроков</th>
                    <th>Статус регистрации</th>
                </tr>
            </thead>

            <tbody>
                <?php foreach ($items as $item) {
                    echo '<tr data-
href=tournaments/' . $item[1] . '>';

                    if($item[4] == 'open'){
                        echo "<td class='cv1'>
                        <div id='outer-circle'>
                            <div id='inner-
circle'>

                                </div>
                            </div>
                        </td>";
                    };

                    if($item[4] == 'full'){
                        echo "<td class='cv2'><div
id='outer-circle'>

                            <div id='inner-
circle'>

                                </div>
                            </div>
                    }
                }
            }
        }
    }

```

```

        </div></td>";
    };
    if($item[4] == 'online'){
        echo "<td class='cv3'><div
id='outer-circle'>
        <div id='inner-
circle'>
        </div>
        </div></td>";
    };
    if($item[4] == 'ended'){
        echo "<td class='cv4'><div
id='outer-circle'>
        <div id='inner-
circle'>
        </div>
        </div></td>";
    };

    echo'
<td>Spellbreak tournament
№'.$item[1].'('.$item[3].')</td>
<td>'.$item[5].'</td>
<td>'.$item[6].'</td>';

    if($item[4] == 'open'){
        echo "<td>Регистрация
открыта</td>";
    };
    if($item[4] == 'full'){
        echo "<td>Мест нет</td>";
    };
    if($item[4] == 'online'){
        echo "<td>Турнир идет
сейчас</td>";
    };

```

```

        };
        if($item[4] == 'ended'){
            echo "<td>Турнир
завершен</td>";

        };

        echo "</tr>";
    } ?>
</tbody>
</table>
</div>
</div>
</div>

<?php
$this->theme->footer();
?>
<script type="text/javascript">
    $(function(){
        $('*[data-href]').click(function(){
            window.location = $(this).data('href');
            return false;
        });
    });
</script>
</body>
</html>

```

Б.3 Файл style.css

```

.tournaments{
    background: url("../img/fon2.png") no-repeat !important;
    height: auto !important;
    padding-bottom: 0px !important;
}

```

```
        -webkit-background-size: cover !important;
        background-size: cover !important;
    }

.tournaments::after {
    content: "";
    display: none !important;
    position: fixed;
    padding: 100px 0;
    width: 100%;
    height: 100vh;
    top: 0;
    left: 0;
}
```

ДОДАТОК В
Слайди презентації

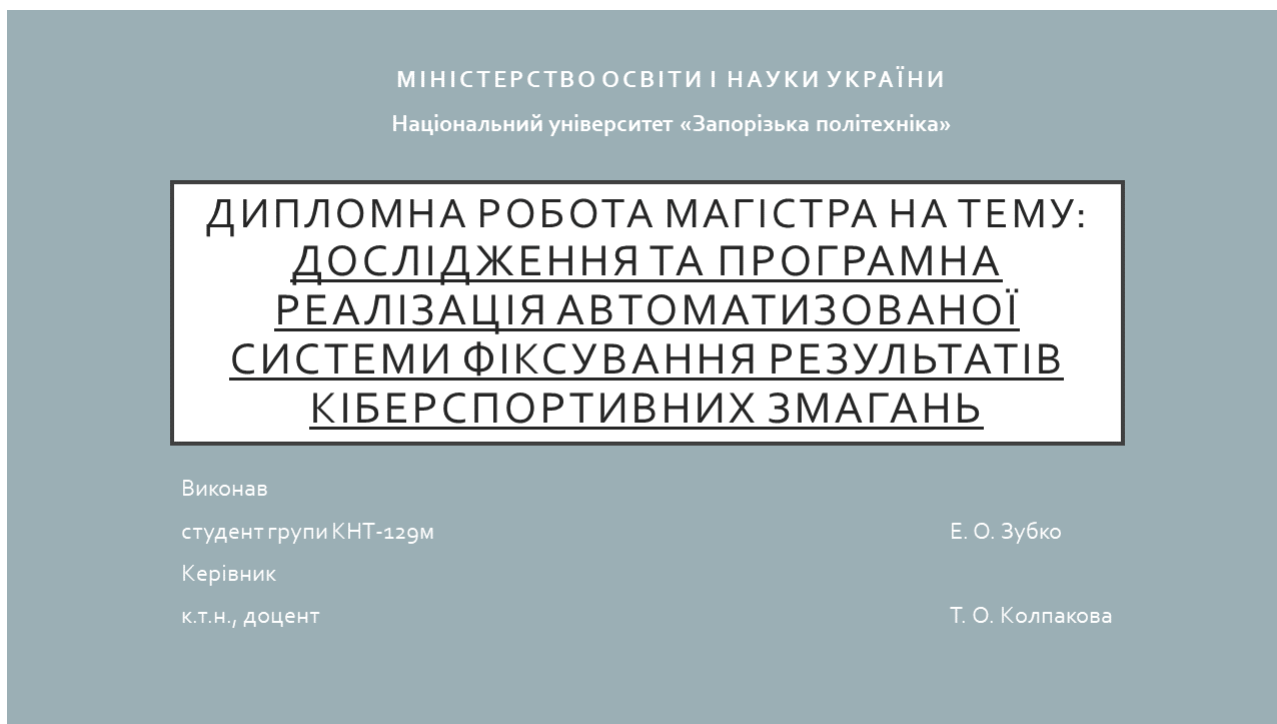


Рисунок В.1 – Титульний слайд

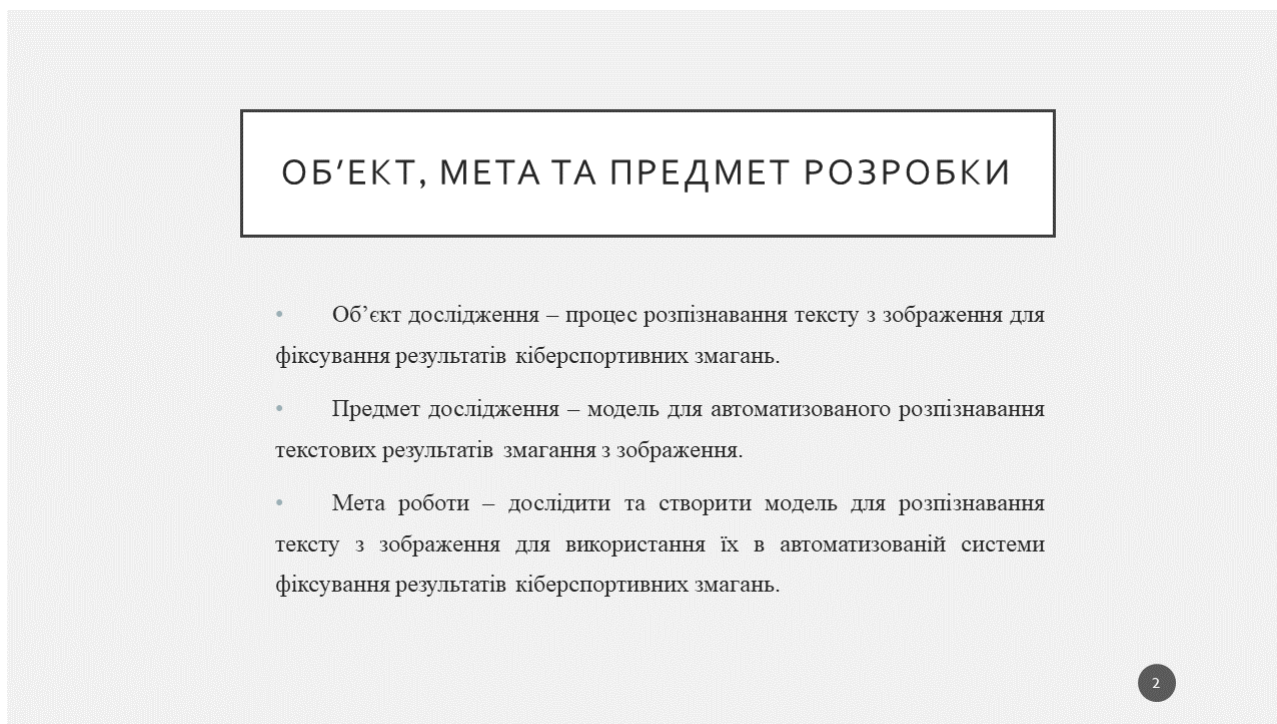


Рисунок В.2 – Об'єкт, мета та предмет розробки

ПОСТАНОВКА ЗАДАЧІ

- Виконати огляд аналогів, що реалізують поставлені завдання.
- Виконати аналіз технічних засобів, що можуть бути використані для розробки.
- Дослідити та створити модель розпізнавання тексту з зображення.
- Виконати проектування архітектури системи.
- Розробити схеми баз даних.
- Виконати розробку системи.
- Провести тестування розробленого програмного продукту.

3

Рисунок В.3 – Постановка задачі

ФУНКЦІОНАЛЬНІ ВИМОГИ

Система автоматичного фіксування результатів має мати такі функції:

- створення та керування турніром на всіх його етапах;
- Інтерфейс роботи з системою має бути месенджер Discord;
- автоматичне зчитування результатів змагань з зображень учасників;
- повноцінна турнірна таблиця на сайті, яка оновлюється онлайн;
- максимальна швидкість та простота у роботі з системою.

4

Рисунок В.4 – Функціональні вимоги

ОГЛЯД ГОТОВИХ СИСТЕМ ДЛЯ ПРОВЕДЕННЯ ТУРНІРІВ

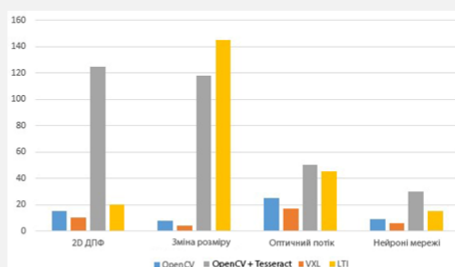
	Toornament.com	Weplay.tv	Game.tv	Challonge.com	Cybernetica.agency
Безкоштовний базовий функціонал	Є	Є	Є	Є	Немає
Підтримка месенджеру Discord	Немає	Є	Є	Немає	Є
Українська мова інтерфейсу	Немає	Є	Немає	Немає	Є
API сервісу	Є	Немає	Є	Є	Є
Автоматичне зчитування результатів	Немає	Немає	Немає	Немає	Є
Наявність онлайн сторінки з результатами	Є	Немає	Є	Є	Є
Коштовність системи з усім функціоналом	50 \$	150 \$	250 \$	400 \$	Більш ніж 10000 \$

5

Рисунок В.5 – Огляд готових систем для проведення турнірів

АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ

- OpenCV
- OpenCV + Tesseract
- VXL
- LTI



6

Рисунок В.6 – Аналіз технологій розробки



Рисунок В.7 – Логічна схема роботи системи

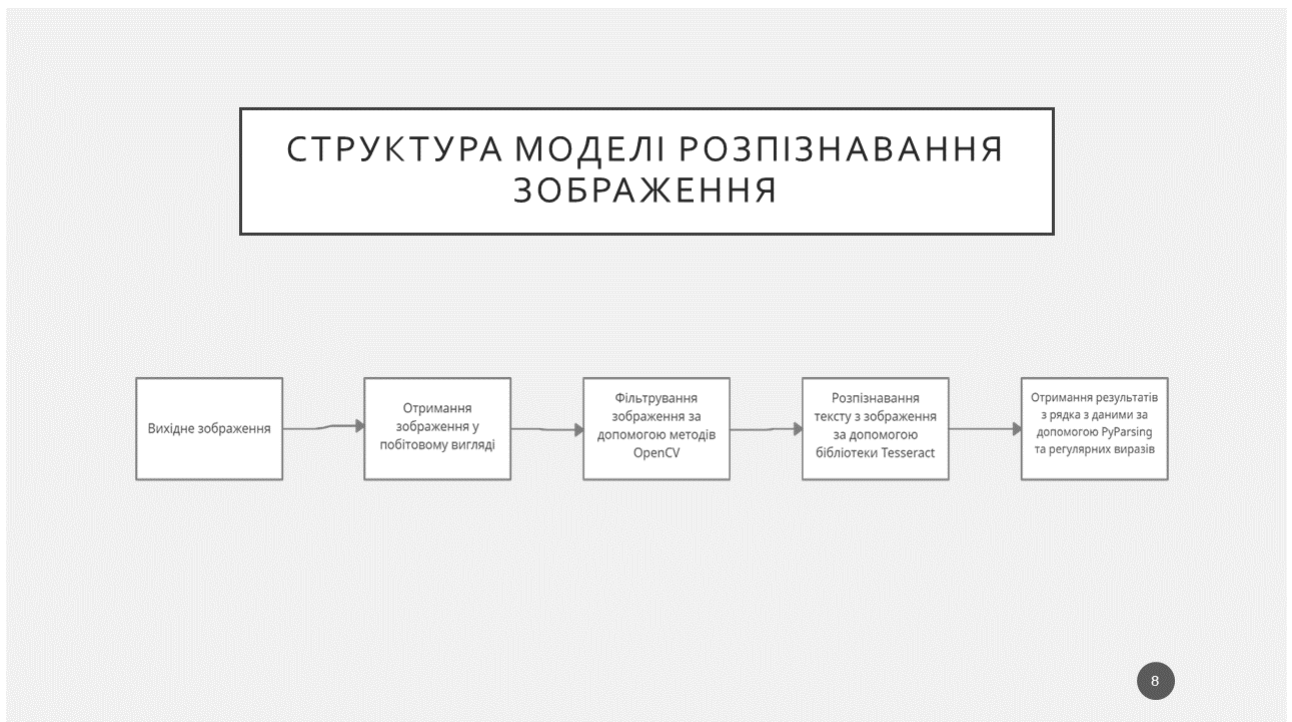


Рисунок В.8 – Структура моделі розпізнавання зображення

ДОСЛІДЖЕННЯ ЕФФЕКТИВНОСТІ МОДЕЛІ



Зображення після обробки функцією cvtColor

```
R 7
| le hd
> ' @
é Y
- <- R
A ,
> - }
: <= > I< i
Assists;0 8 \* ens " = H
Time Survived: 811 = \
seconds , s |
- ON
Accolades XP: 199 . ee. ee = ee
ee 1 le Peta ae
ed = (le SG a FON Me 4 A Aihataler \
0
0 he Sa a
a Serre ee hea ASSISTS: 0
= exit we Sy
any ce
INS a) -
lo* Final Se
9
```

Результат розпізнавання

9

Рисунок В.9 – Дослідження ефективності моделі

ДОСЛІДЖЕННЯ ЕФФЕКТИВНОСТІ МОДЕЛІ



Зображення після обробки двома функціями
cvtColor та gaussianBlur

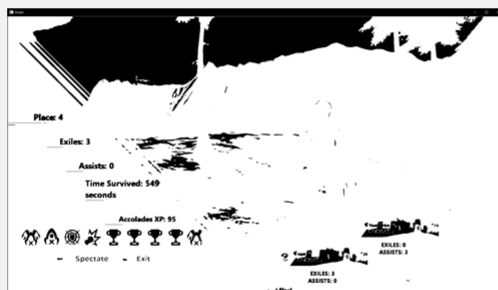
```
= . bd
» r
es (3 .
f
ee ee ee J
_ Assists: 0 =e,
SO aie eo s '
Prato) To \ "a
aan N1
[Rees et ee Ce . ~ _
~ ee aly
LLGBILALED a
x) & EXILES: 0
Pa ee Cn)
= Exit re) rhe )
rods =
Se) =
i. * Final 7
9
```

Результат розпізнавання

10

Рисунок В.10 – Дослідження ефективності моделі

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МОДЕЛІ



Зображення після обробки трьома функціями
cvtColor, gaussianBlur та threshold

```
Place: 4
Exiles: 3
Assists: 0
Time Survived: 549
seconds
Accolades XP: 95
OG LLL LR

Thundersin =f : '
Pre-Alpha Tester
EXILES: 0
os 4 ~ Tyg ASSISTS: 3
**Pre-Alpha Tester
EXILES: 3
ASSISTS: 0
* Final
0
```

Результат розпізнавання

11

Рисунок В.11 – Дослідження ефективності моделі

ІНСТРУМЕНТИ РЕАЛІЗАЦІЇ

- Для розробки системи була обрана мова програмування Python з використанням додаткових бібліотек: Tesseract, OpenCV, BytesIO, PIL, Discord API.
- В якості середовища розробки було обрано JetBrains PyCharm.
- Для створення веб-сторінки з рейтинговою таблицею був використаний backend-фреймворк Laravel та frontend-фреймворк Bootstrap.

12

Рисунок В.12 – Інструменти реалізації

ІНТЕРФЕЙС ВЗАЄМОДІЇ З СИСТЕМОЮ

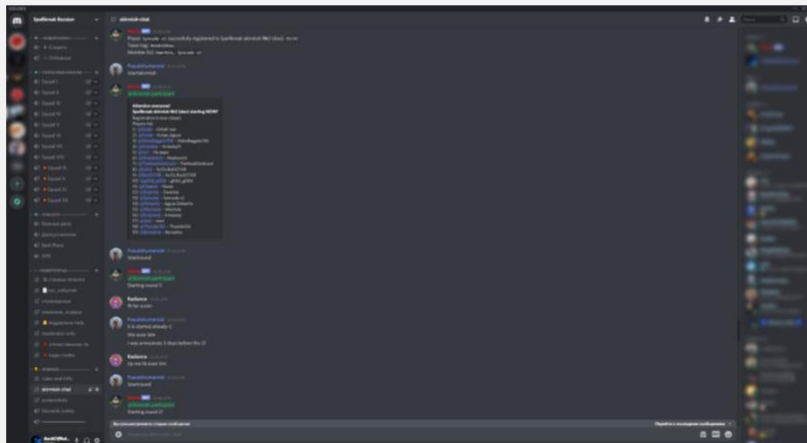


Рисунок В.13 – Інтерфейс взаємодії з системою

ВИСНОВКИ

- Досліджено та створено модель розпізнавання тексту з зображення для реалізації її у автоматизованій системі фіксування результатів кіберспортивних змагань.
- Було розроблено автоматизовану систему фіксування результатів кіберспортивних змагань.
- Практична цінність розробленої системи полягає в тому, що тепер можна легше та ефективніше проводити онлайн турніри.
- Наукова новизна роботи полягає в тому, що була запропонована удосконалена модель розпізнавання текстових даних. Модель включає нейромережеву модель Tesseract та етапи підготовки зображення та постобробки результатів.

Рисунок В.14 – Висновки

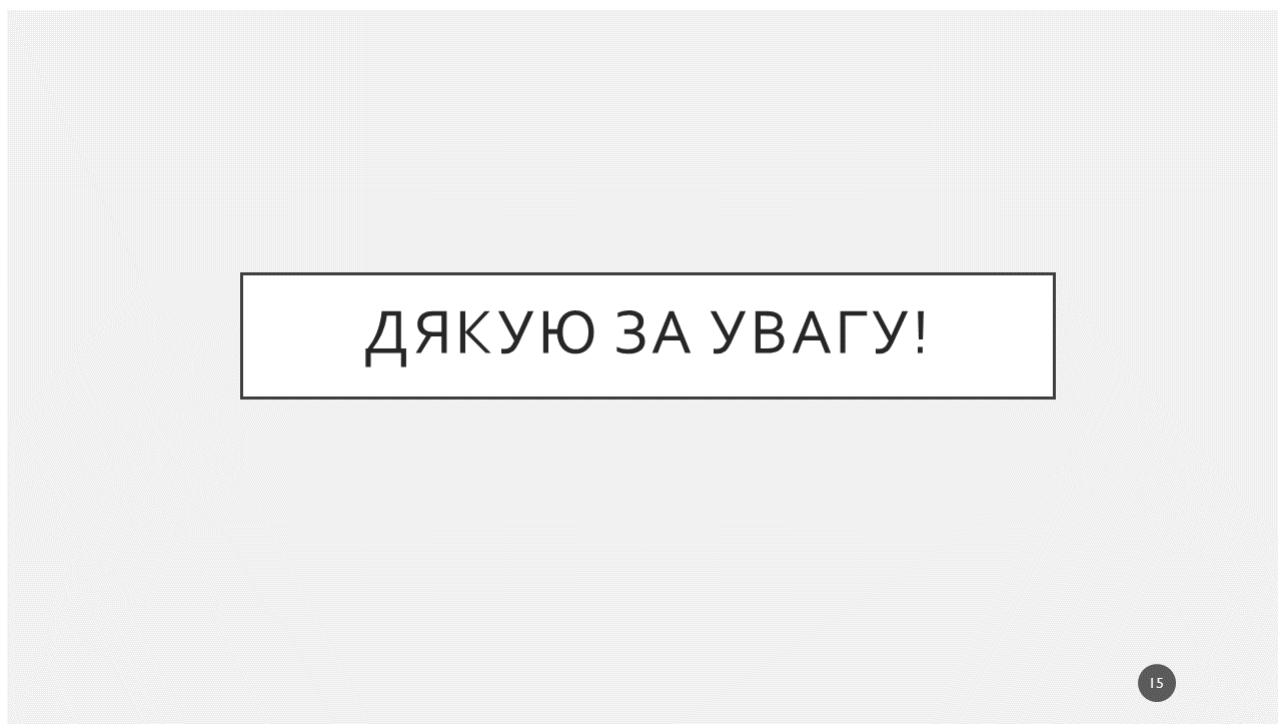


Рисунок В.15 – Дякую за увагу