

ДОСЛІДЖЕННЯ ТА РОЗРОБКА БОТІВ ДЛЯ АВТОМАТИЗАЦІЇ ІГРОВИХ ПРОЦЕСІВ ПІД УПРАВЛІННЯМ ЛЮДИНИ ТА ШТУЧНОГО ІНТЕЛЕКТУ

СТУДЕНТ ГРУПИ ФЕІМ-22 ТОВКАЧ Б. М.
НАУКОВИЙ КЕРІВНИК ДОЦ. ДЕМКІВ. Л. С.

ПЕРЕДМОВА

Консольний проєкт (бот) для автоматизації ігрового процесу, з яким у мене є досвід, мав підстави розвиватися і надалі, адже програма без взаємодії з людиною не могла бути користувацькою, тому що не було інтерфейсної частини. **Використання центрального процесора** комп'ютера, замість відеокарти, при виконанні роботи нейронними мережами та **непродумана архітектура програми** - **створювали ряд проблем** для оптимізації бота. Проєкт, який буде представлено в роботі, це розвиток попередніх напрацювань.

ПЛАН ПРЕЗЕНТАЦІЇ

- Вступ
- Огляд технологій та інструментів
- Розробка та реалізація проєкта
- Результати тестування
- Дослідження отриманих результатів
- Висновки

ВСТУП

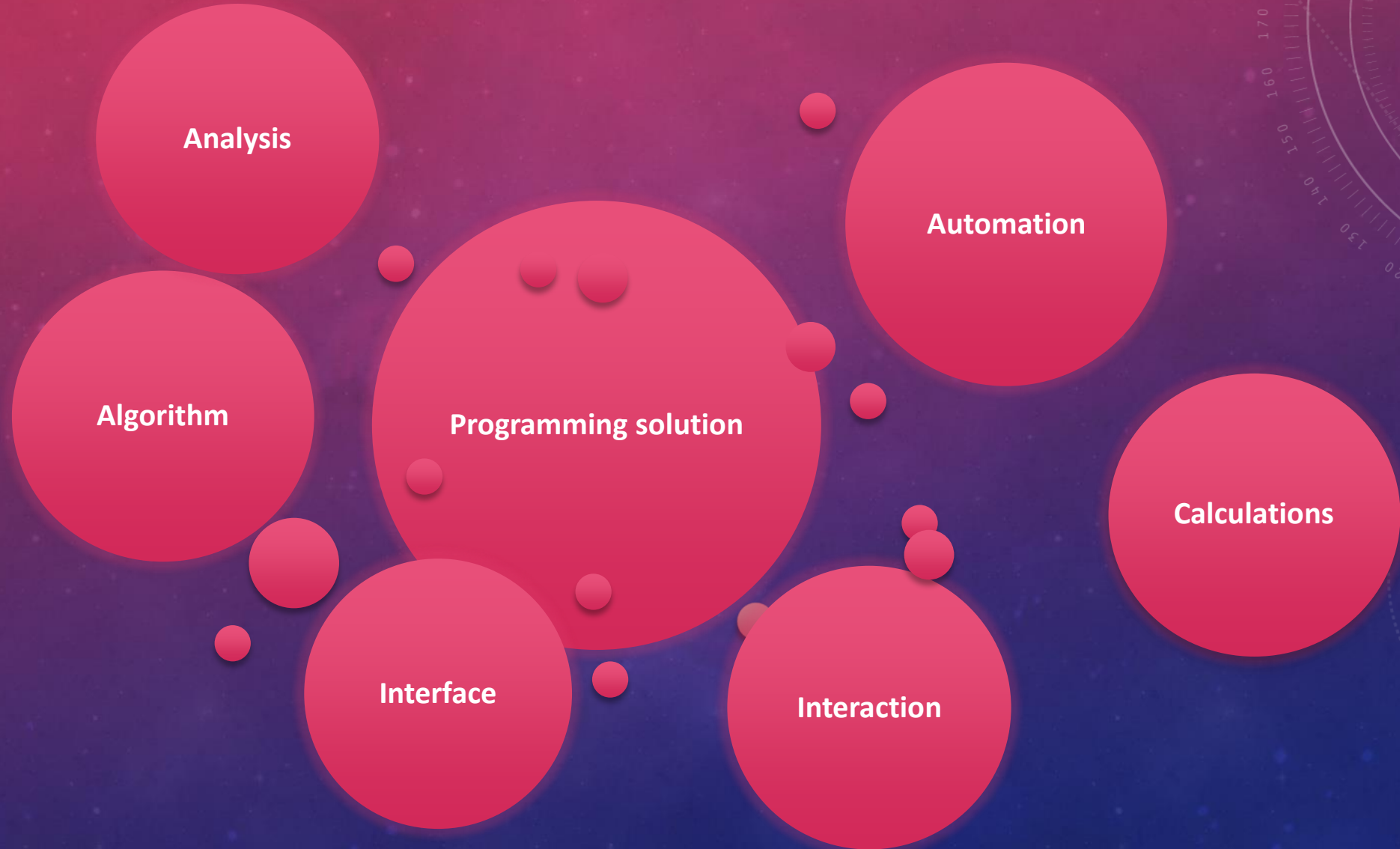
Програми, які допомагають, або надають перевагу, можуть застосовуватися в іграх для **оптимізації ігрового процесу**. Людина потребує **графічного інтерфейсу** для повноцінної взаємодії з ботом, що полегшить **аналіз та моніторинг**, як гри, так і програми. Розробка засобів моніторингу та аналізу із подальшою реалізацією у **системі управління робототехнікою** показує результат на прикладі Boston Dynamics.

Ідея роботи: розробка і дослідження оптимізованого, ефективного бота для спрощення взаємодії людини та програми з перспективами його модернізації.

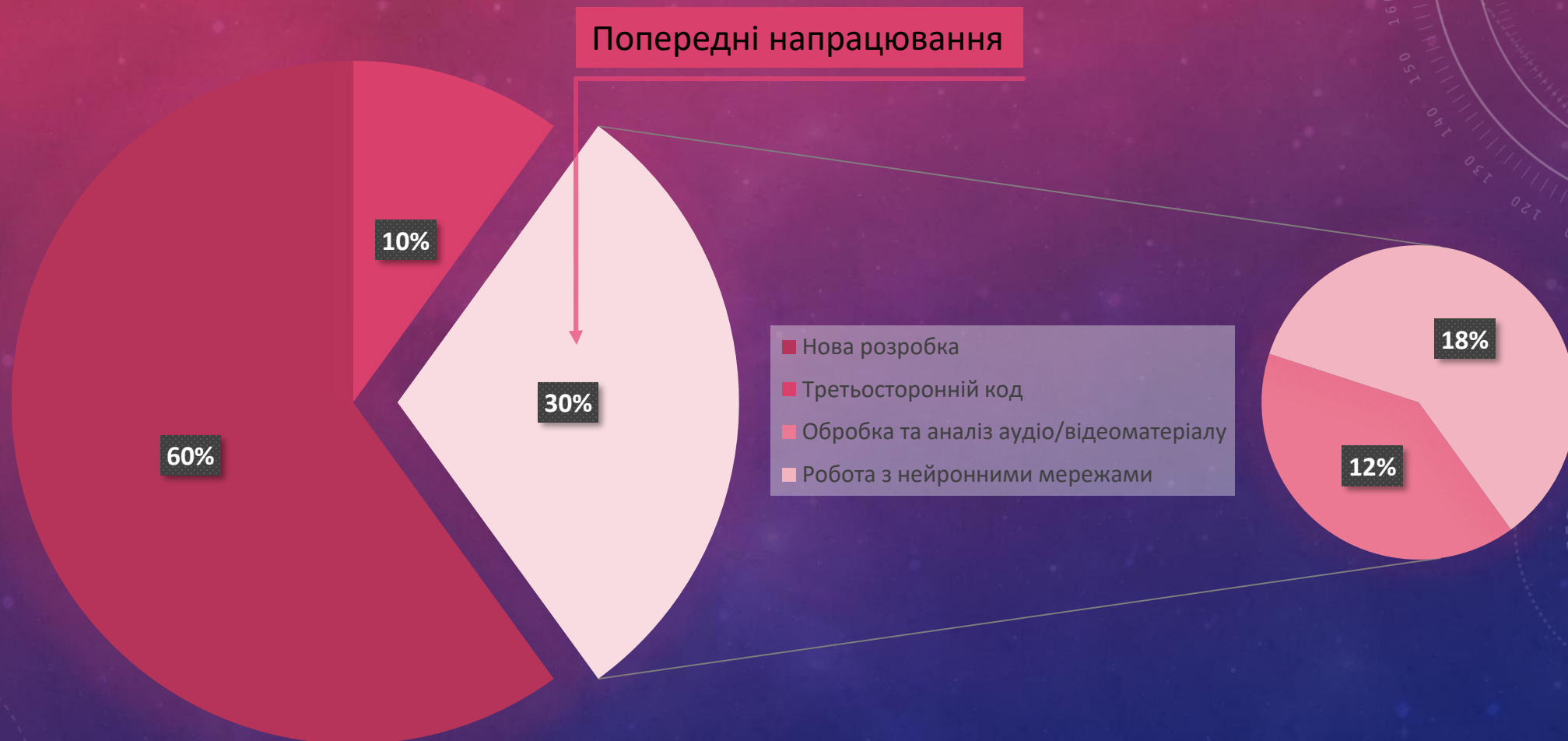
Мета роботи: реалізація віконного додатка з інтеграцією функцій розпізнавання, взаємодії з ОС та системами моніторингу. Оптимізація ключових моментів ігрового процесу та програми для зменшення навантаження на систему ПК.



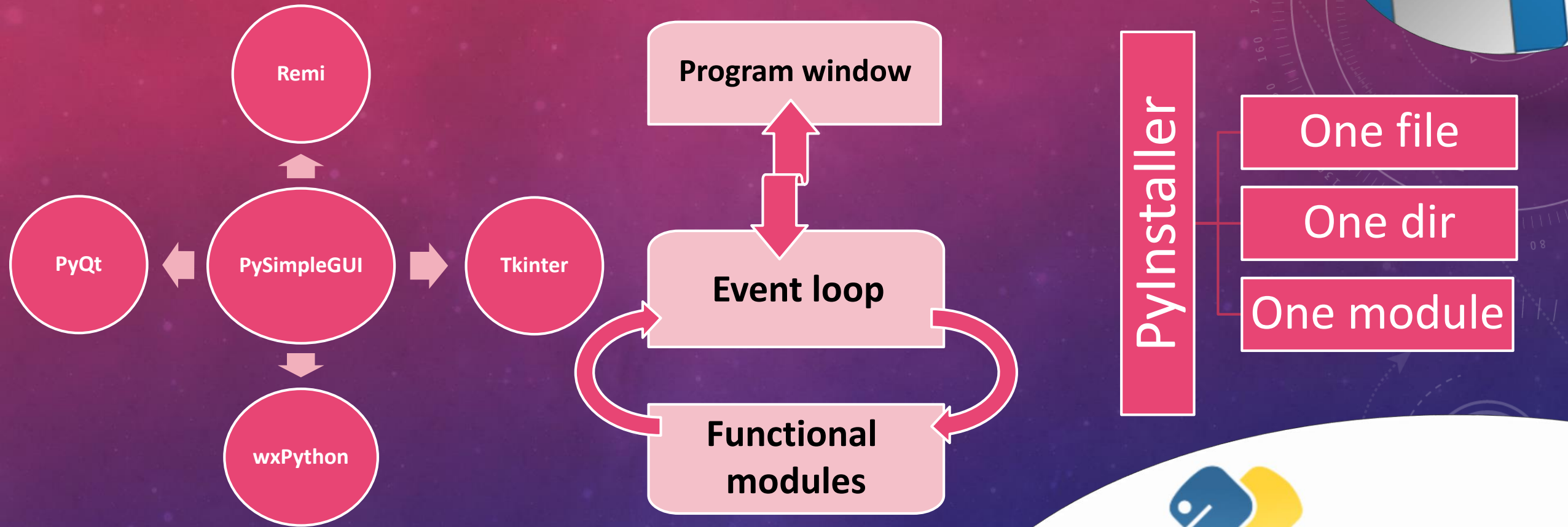
ТЕХНОЛОГІЯ РОЗРОБКИ БОТА



ЧАСТКА ПОПЕРЕДНІХ НАПРАЦЮВАНЬ У РОЗРОБЦІ НОВОГО ПРОЄКТА



ТЕХНОЛОГІЇ РОЗРОБКИ ВІКОННОГО ДОДАТОКА, ЗАБЕЗПЕЧЕННЯ КРОС-ПЛАТФОРМНОСТІ ПРОГРАМИ



PySimpleGUI обгортає tkinter, Qt, WxPython і Remi, аби ми отримували однакові віджети, але взаємодіяли з ними більш легшим способом, який є загальним для всіх портів.

ПРОГРАМНИЙ ІНТЕРФЕЙС ВЗАЄМОДІЇ З WINDOWS

User Interface and desktop

[Windows controls](#)

[Menus and other resources](#)

Graphics and gaming

[Windows GDI](#)

[Windows Imaging Component](#)

Audio and video

[Windows Multimedia](#)

[Core Audio APIs](#)

Windows API, неофіційно WinAPI, — це основний набір інтерфейсів прикладного програмування Microsoft, доступний в операційних системах Microsoft Windows.

PyWin32 - Python extension for using Win32 API

Controllers

- Keyboard control
- Mouse control
- Screen control

Win32 API

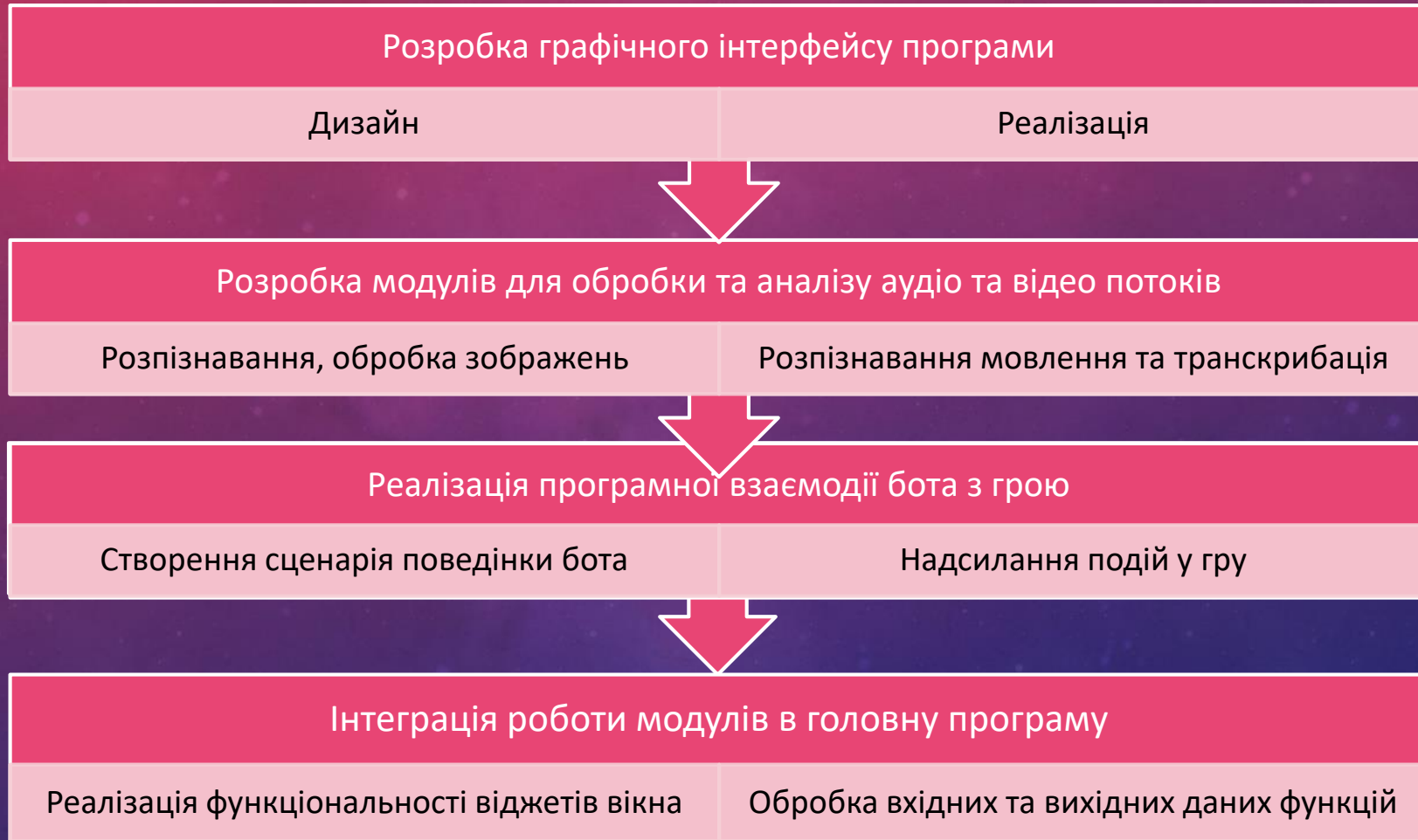
- User Interface and desktop
- Graphics and gaming
- Audio and video



PyWin 32

PyWin32 — це бібліотека розширень Python для Windows, яка дає змогу використовувати функції інтерфейса прикладного програмування Win32 (API) на Python.

РОЗРОБКА ПРОЄКТА



СТРУКТУРА ПРОЄКТА ТА АРХІТЕКТУРА ПРОГРАМИ

Модулі головної програми

Модуль з даними для
неймережі (дата сет,
модуль, конфігурації)

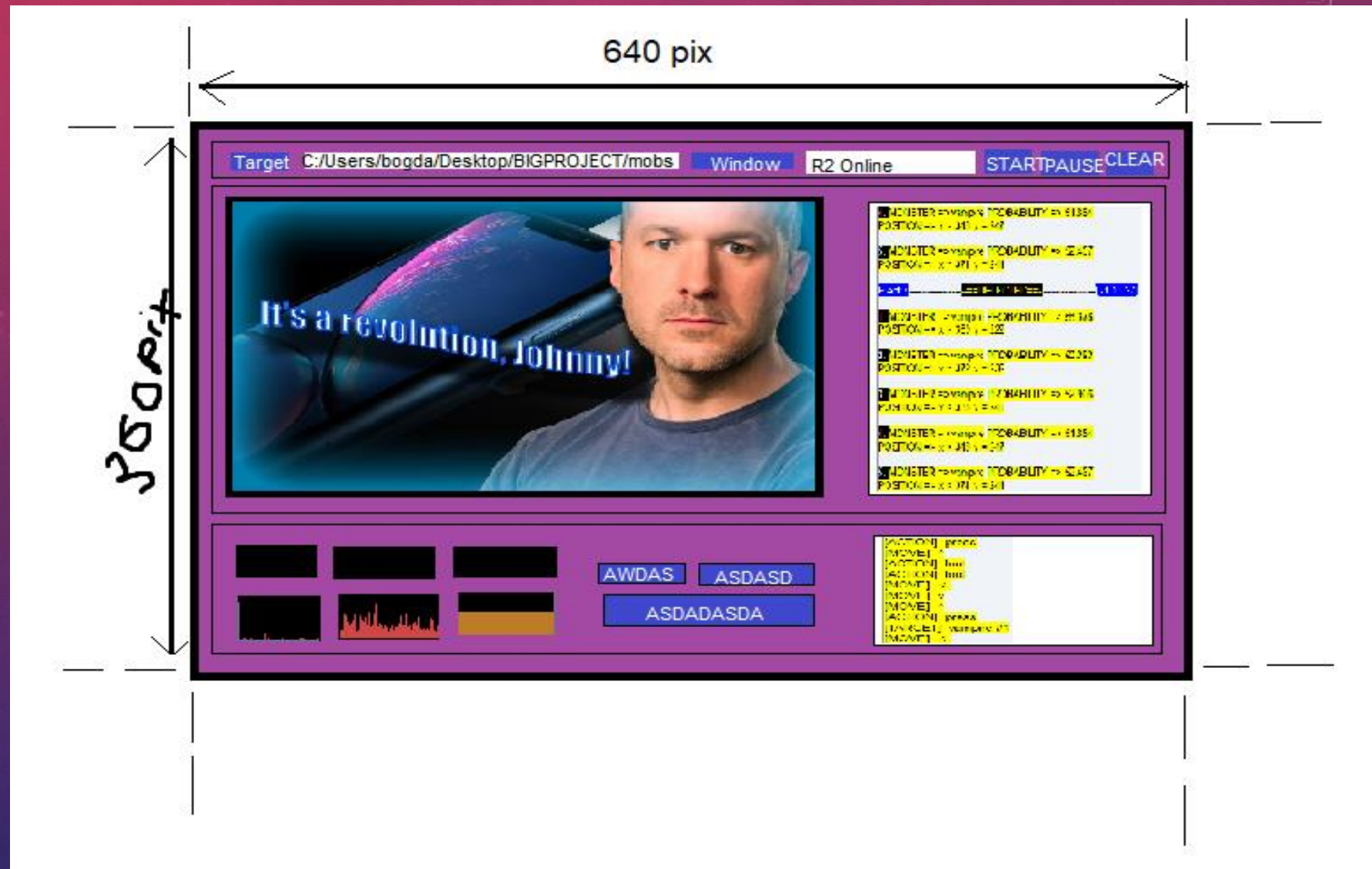
Модуль для запуску скриптів
тестування, зберігання
статисти та запуску програми

Модуль для збирання
вихідних даних програми

Модуль для опрацювання
вихідних даних програми

Функціональне програмування набуває популярності завдяки своїй ефективності та універсальності у вирішенні сучасних проблем. Програми створюються за допомогою послідовних функцій, кожна з яких приймає вхідне значення і повертає постійне вихідне значення, не змінюючи або не впливаючи на роботу самої програми. Саме такий підхід був використаний у цьому проєкті.

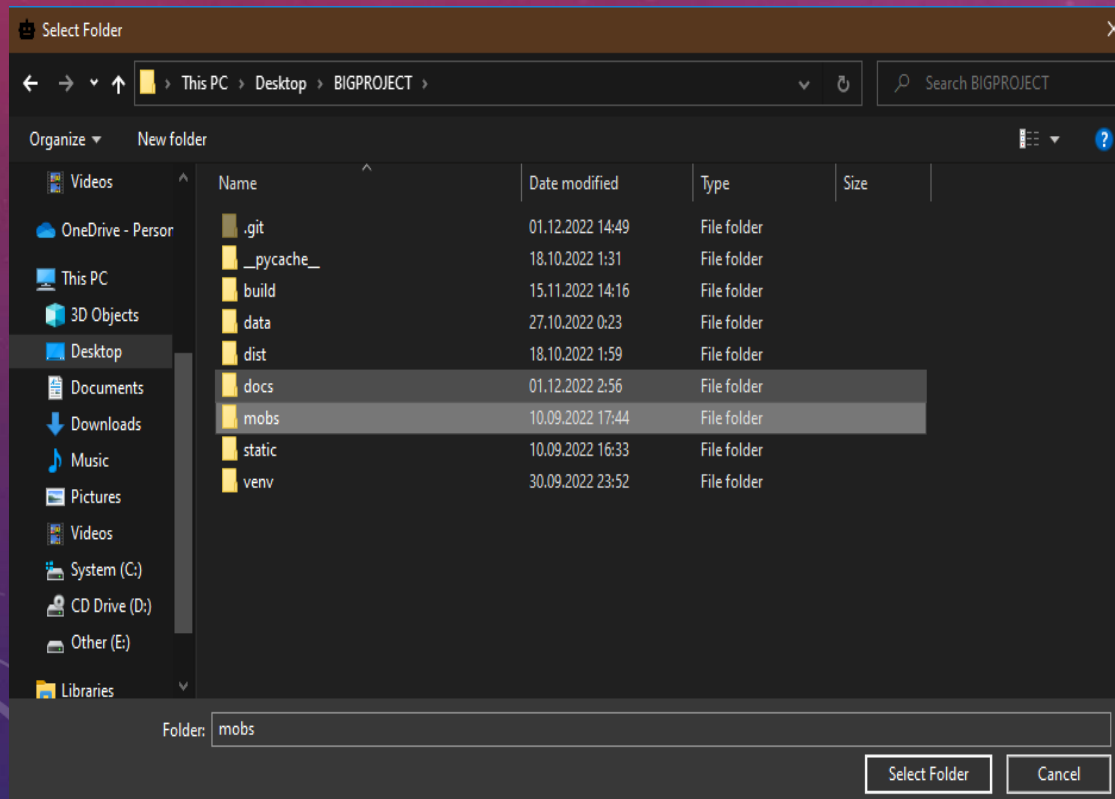
РОЗРОБКА МАКЕТА ВІКНА ПРОГРАМИ



ФОРМА ДЛЯ НАЛАШТУВАННЯ ТА КЕРУВАННЯ ПРОГРАМОЮ

Вибір цільової директорії

Target



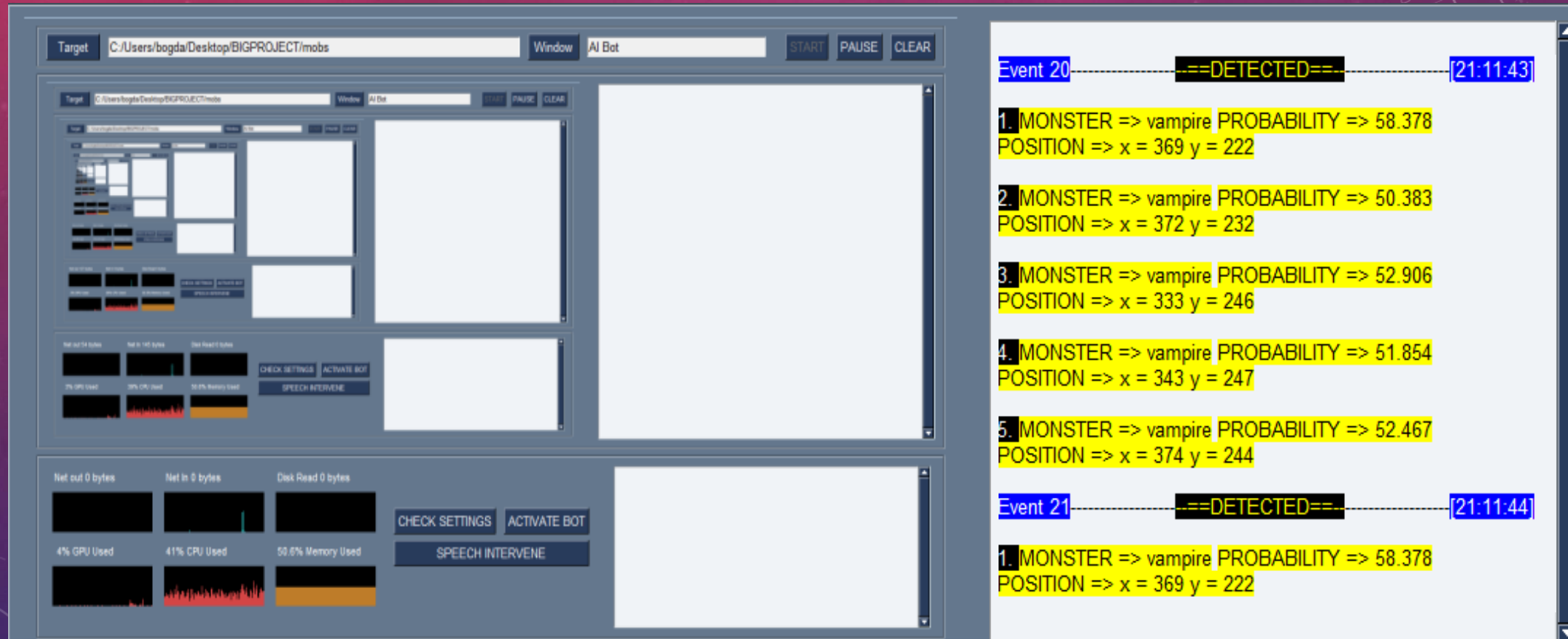
Кнопки керування програмою,
вибір цільового вікна

START PAUSE CLEAR

Window

- main.py - BIGPROJECT - Visual Studio Code
- BIGPROJECT
- Диплом.docx - Word (не вдалося активувати пр
- Здача.pptx - PowerPoint (не вдалося активуват
- Диплом.pdf and 11 more pages - Personal - Micr
- Settings
- Settings
- Microsoft Text Input Application
- NVIDIA GeForce Overlay
- docs
- Telegram (436)
- Steam
- Program Manager

ФОРМА ДЛЯ ВИВЕДЕННЯ ВІДЕОТРАНСЛЯЦІЇ ТА ТРАНСКРИБАЦІЇ ЦІЛЬОВО ВІКНА ПРОГРАМИ



The interface displays a target window and a list of detected events. The target window is titled "C:/Users/bogda/Desktop/BIGPROJECT/mobs" and contains a smaller window titled "AI Bot". The list of detected events is as follows:

Event 20 ---DETECTED--- [21:11:43]

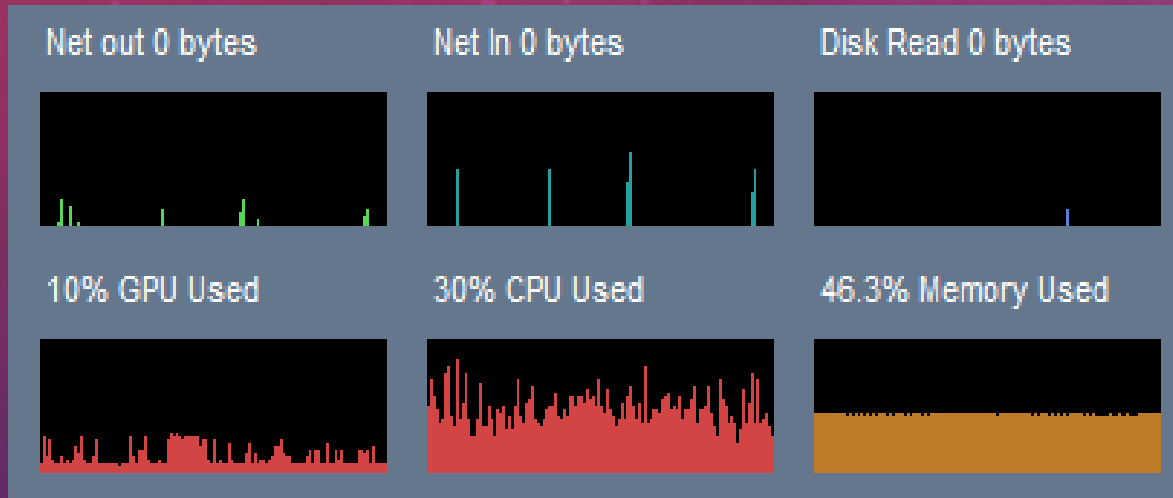
- 1. MONSTER => vampire PROBABILITY => 58.378
POSITION => x = 369 y = 222
- 2. MONSTER => vampire PROBABILITY => 50.383
POSITION => x = 372 y = 232
- 3. MONSTER => vampire PROBABILITY => 52.906
POSITION => x = 333 y = 246
- 4. MONSTER => vampire PROBABILITY => 51.854
POSITION => x = 343 y = 247
- 5. MONSTER => vampire PROBABILITY => 52.467
POSITION => x = 374 y = 244

Event 21 ---DETECTED--- [21:11:44]

- 1. MONSTER => vampire PROBABILITY => 58.378
POSITION => x = 369 y = 222

ФОРМА МОНІТОРИНГУ ЗАВАНТАЖЕННЯ ПК ТА АКТИВНОСТІ БОТА

Графіки завантаження ресурсів ПК

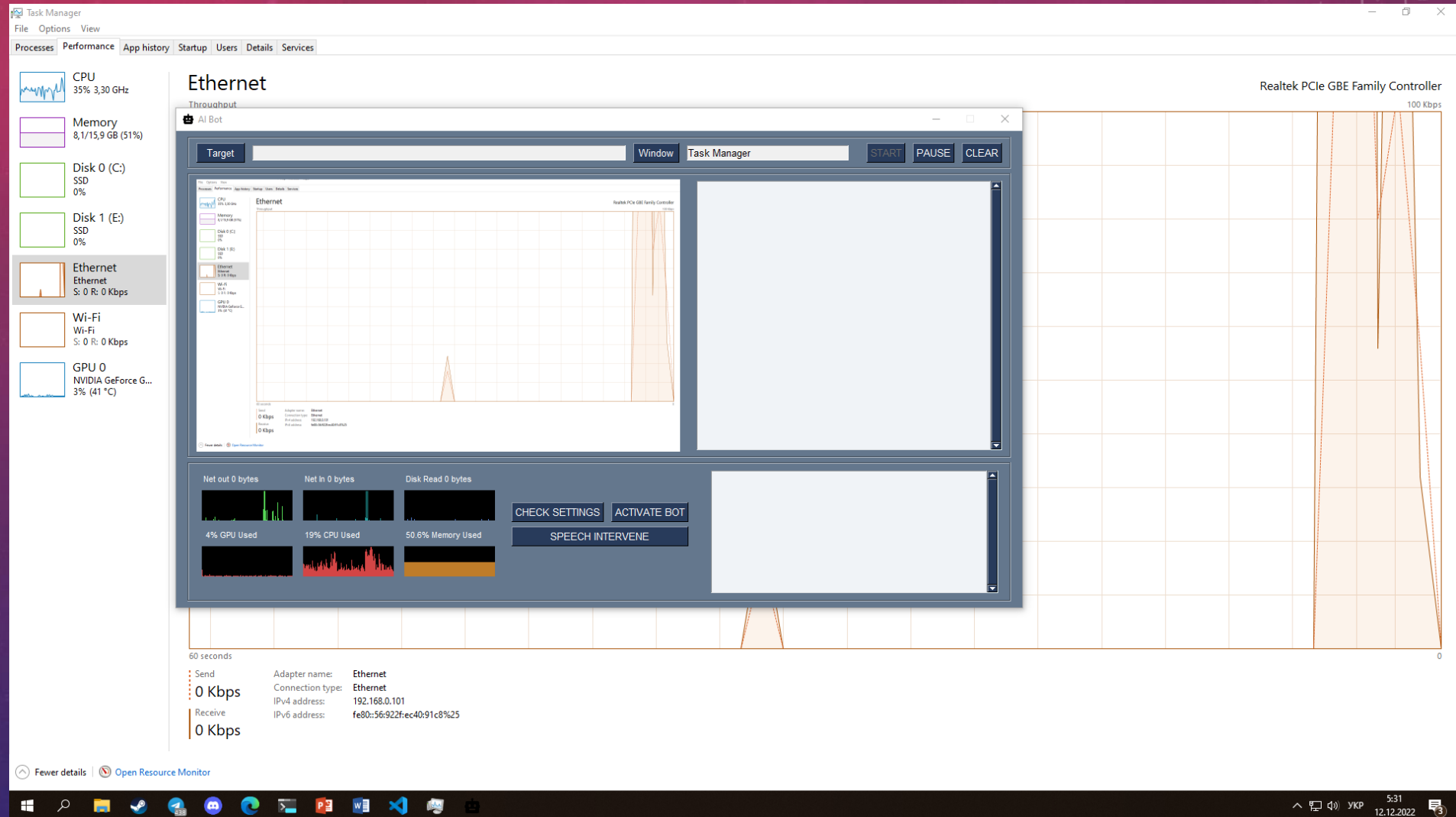


Кнопки перевірки налаштувань
та список активностей бота

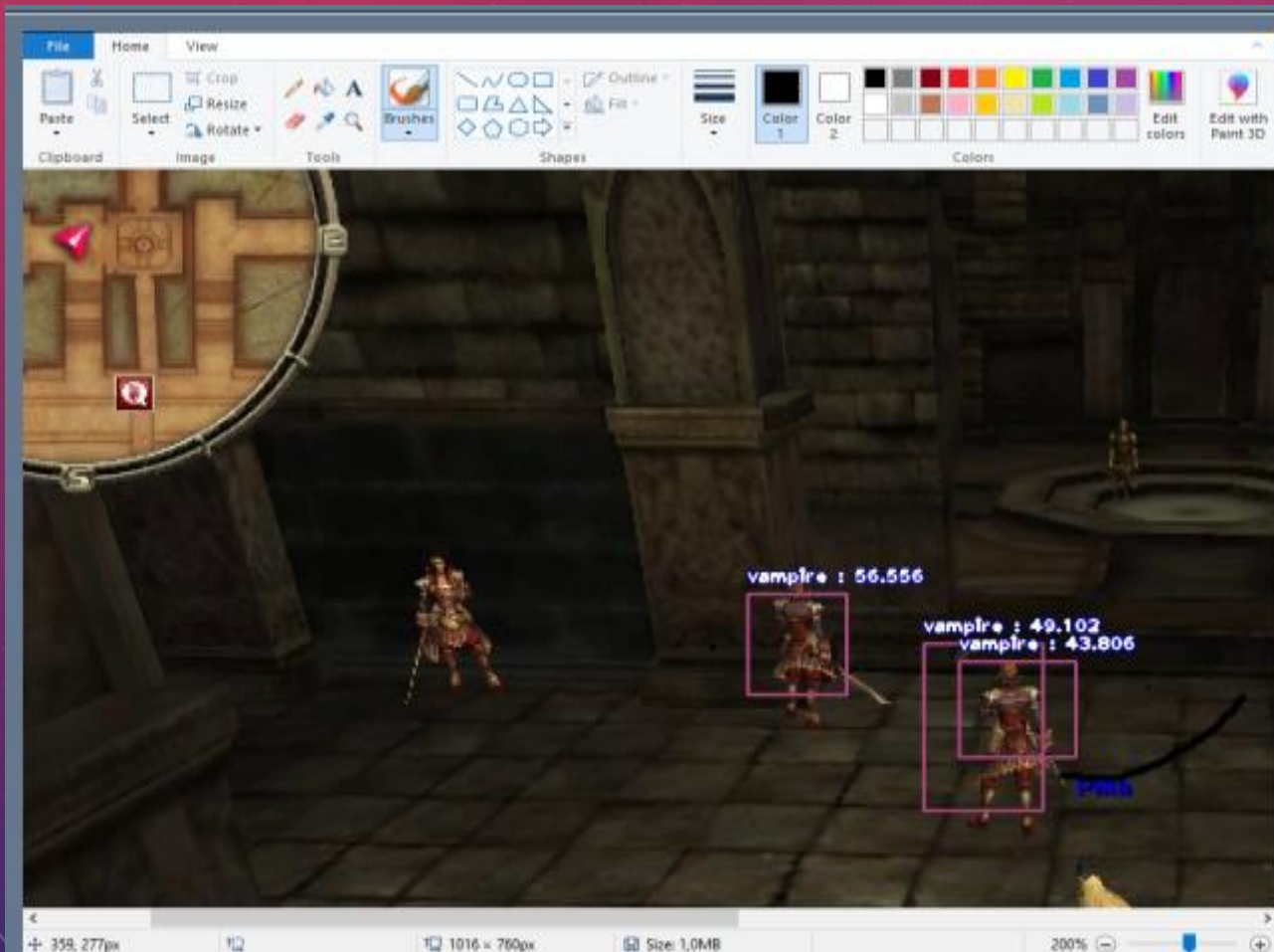
The interface includes three buttons on the left: "CHECK SETTINGS", "ACTIVATE BOT", and "SPEECH INTERVENE". To the right is a scrollable list of bot commands.

```
[MOVE] <-  
[MOVE] v  
[TARGET] vampire #1  
[MOVE] ->  
[MOVE] v  
[ACTION] press  
[MOVE] v  
[MOVE] ->  
[MOVE] ^  
[TARGET] vampire #2
```


РОЗРОБКА МОДУЛЯ ДЛЯ КОМП'ЮТЕРНОГО ЗОРУ ТА РОЗПІЗНАВАННЯ МОВЛЕННЯ



РОЗРОБКА МОДУЛЯ ДЛЯ КОМП'ЮТЕРНОГО ЗОРУ ТА РОЗПІЗНАВАННЯ МОВЛЕННЯ



Event 66 ---DETECTED--- [01:44:01]

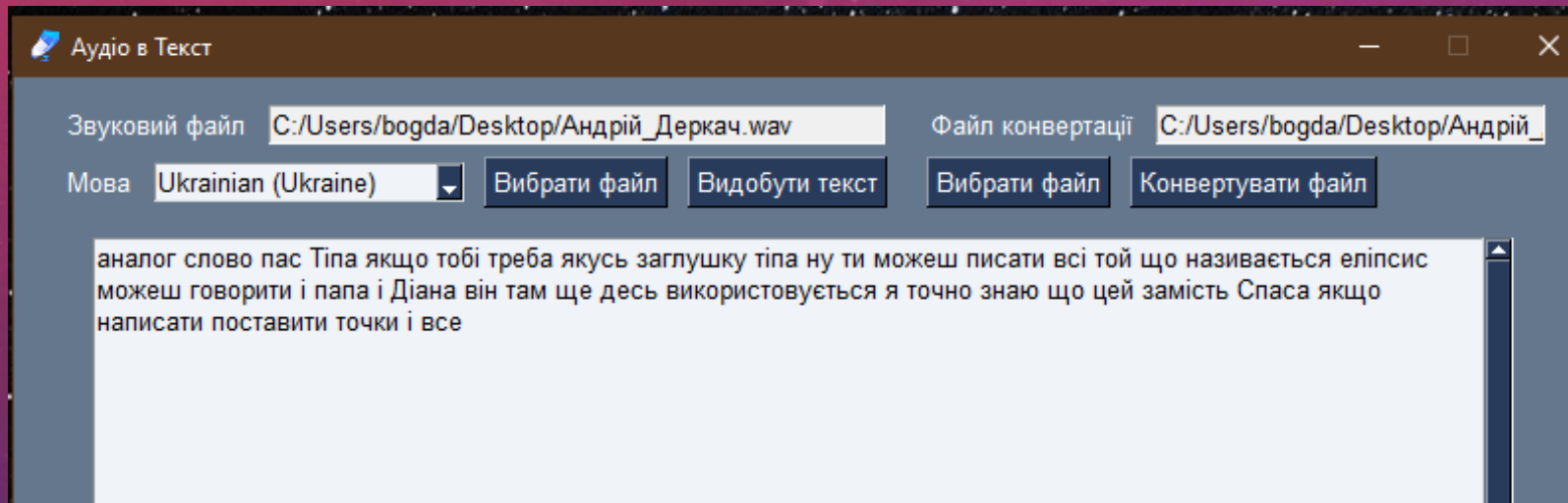
1. MONSTER => vampire PROBABILITY => 56.586
POSITION => x = 572 y = 476
2. MONSTER => vampire PROBABILITY => 49.082
POSITION => x = 711 y = 518
3. MONSTER => vampire PROBABILITY => 43.8
POSITION => x = 739 y = 533

Event 67 ---DETECTED--- [01:44:02]

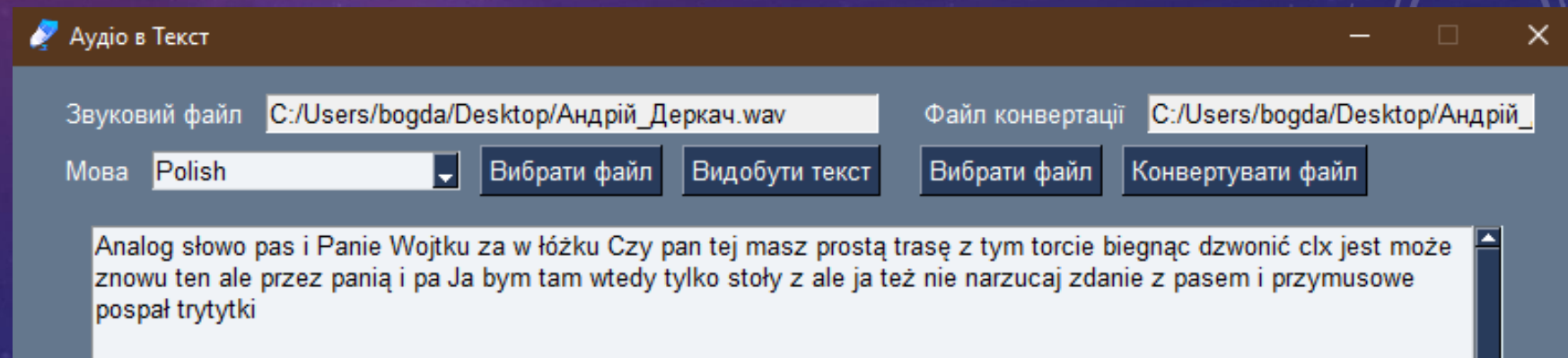
1. MONSTER => vampire PROBABILITY => 56.556
POSITION => x = 572 y = 476
2. MONSTER => vampire PROBABILITY => 49.102
POSITION => x = 711 y = 518
3. MONSTER => vampire PROBABILITY => 43.806
POSITION => x = 739 y = 533

РОЗРОБКА МОДУЛЯ ДЛЯ КОМП'ЮТЕРНОГО ЗОРУ ТА РОЗПІЗНАВАННЯ МОВЛЕННЯ

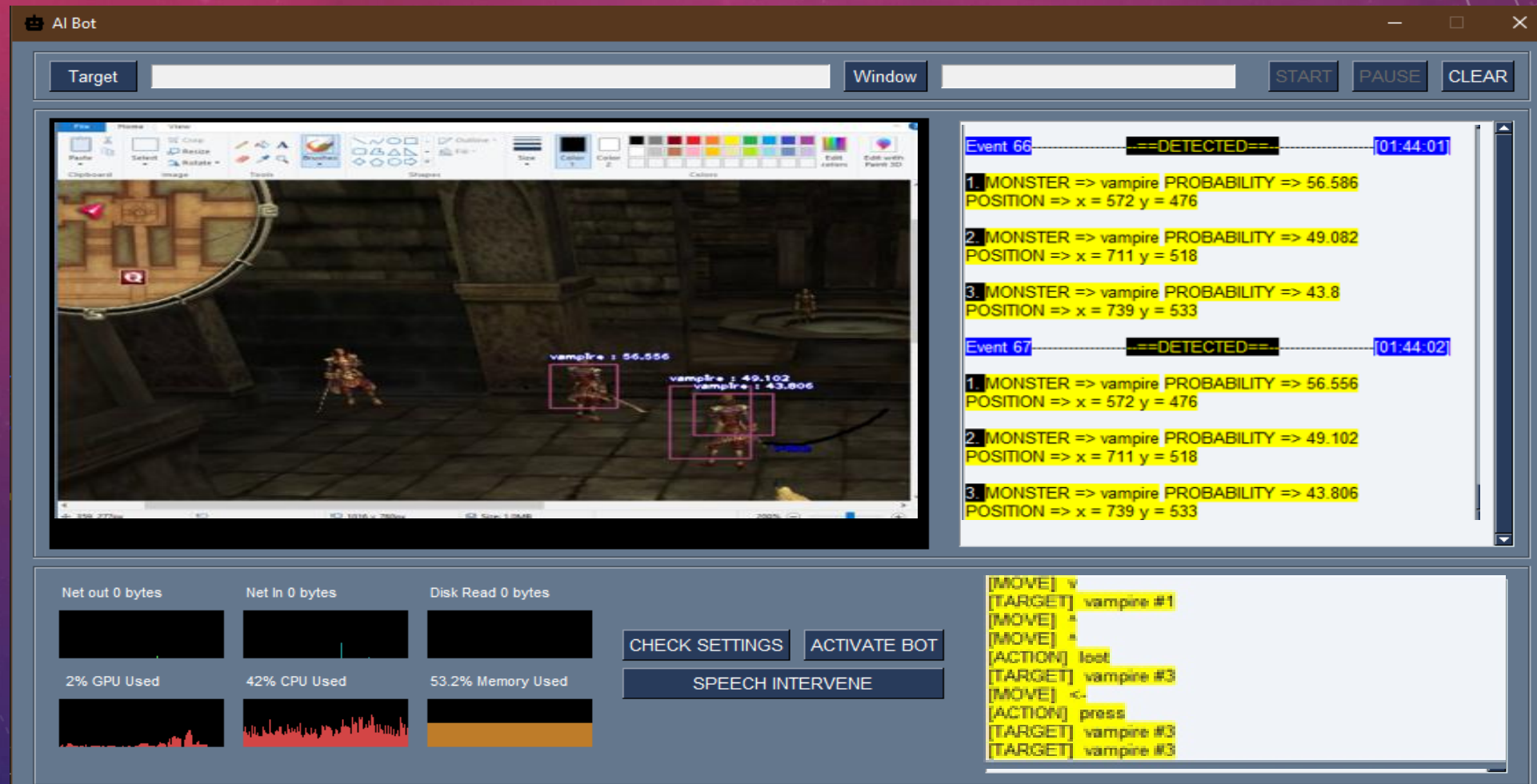
Мова українська



Мова польська



РЕАЛІЗАЦІЯ ВІКОННОЇ ПРОГРАМИ З ІНТЕГРАЦІЄЮ ВСІХ ФУНКЦІЙ



РЕАЛІЗАЦІЯ ВІКОННОЇ ПРОГРАМИ З ІНТЕГРАЦІЄЮ ВСІХ ФУНКЦІЙ

Підготовка роботи модуль з ШІ

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER COMMENTS

```
2022-12-07 01:04:44.602331: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library cusolver64_10.dll
2022-12-07 01:04:44.602472: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library cusparse64_11.dll
2022-12-07 01:04:44.602617: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library cudnn64_8.dll
2022-12-07 01:04:44.602792: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1862] Adding visible gpu devices: 0
2022-12-07 01:04:45.312604: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1261] Device interconnect StreamExecutor with strength 1 edge matrix:
2022-12-07 01:04:45.312958: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1267]      0
2022-12-07 01:04:45.313230: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1280] 0:  N
2022-12-07 01:04:45.313583: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1406] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 2976
ory) -> physical GPU (device: 0, name: NVIDIA GeForce GTX 1050 Ti, pci bus id: 0000:01:00.0, compute capability: 6.1)
2022-12-07 01:04:45.317972: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x259f9d2b530 initialized for platform CUDA (this does not guarantee that XLA
be used). Devices:
2022-12-07 01:04:45.318404: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): NVIDIA GeForce GTX 1050 Ti, Compute Capability 6.1
```

Файл після компіляції

AI Bot.exe
Application



Date modified: 07.12.2022 2:40
Size: 428 MB
Date created: 07.12.2022 2:40

Компіляція у виконавчий файл

```
128679 INFO: Updating resource type 24 name 1 language 0
128684 INFO: Appending PKG archive to EXE
129110 INFO: Fixing EXE headers
134928 INFO: Building EXE from EXE-00.toc completed successfully.
```


ТЕСТУВАННЯ РОБОТИ ПРОГРАМИ



Тестування функції трансляції вікна



Тестування функції розпізнавання

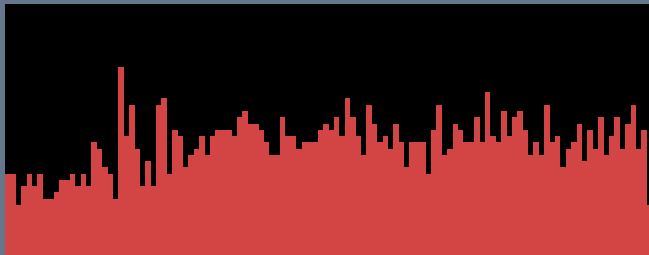


Випробовування та збір даних з довготривалого використання

ТЕСТУВАННЯ РОЗРОБЛЕНОЇ ПРОГРАМИ

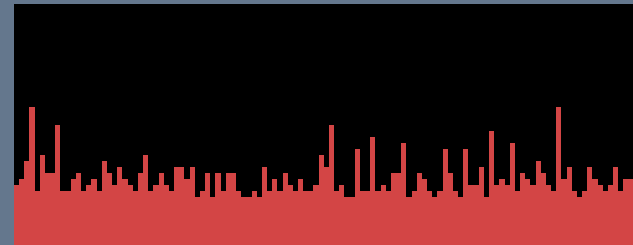
Графік завантаження CPU з
увімкненою трансляцією вікна

22% CPU Used



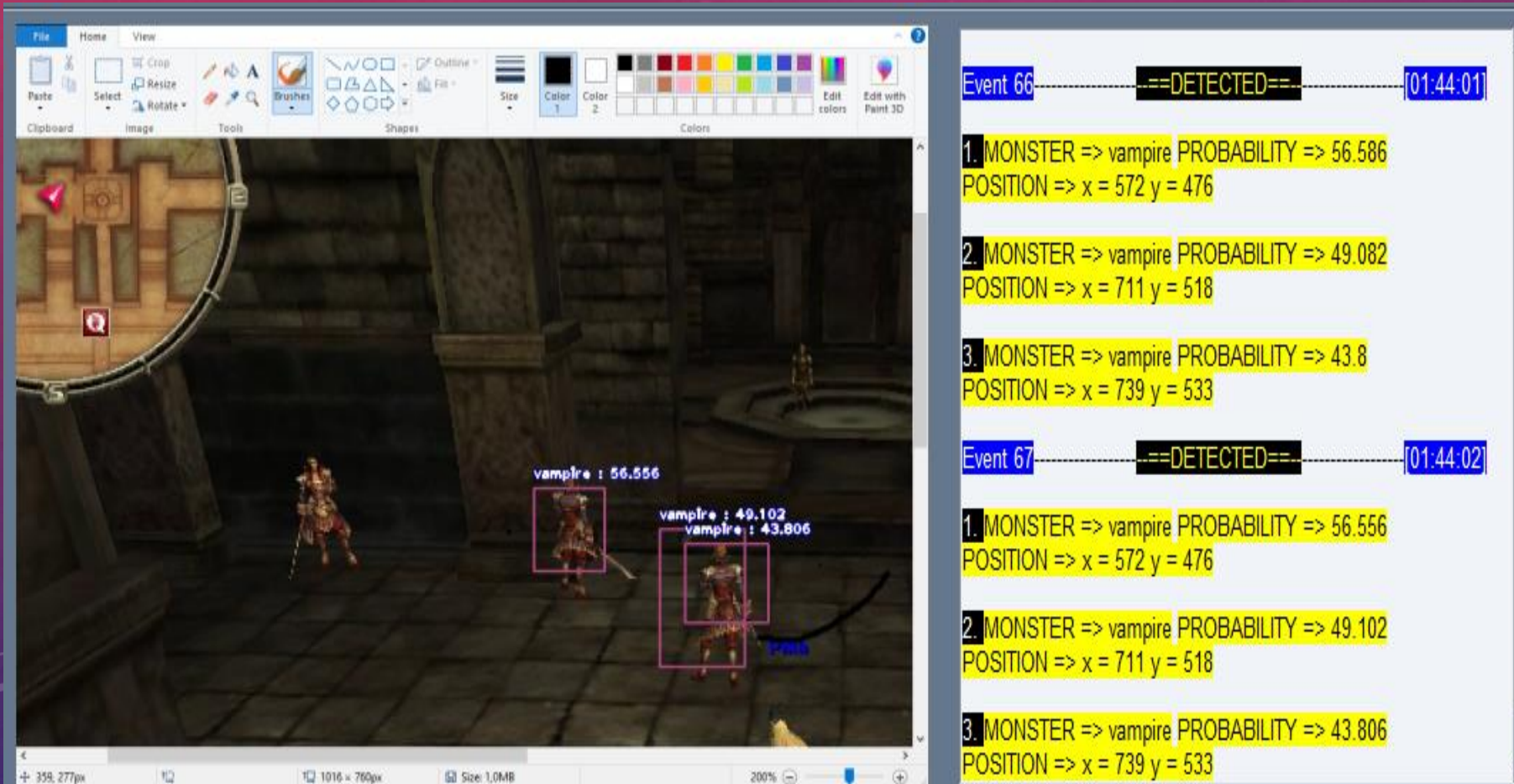
Графік завантаження CPU з
вимкненою трансляцією вікна

31% CPU Used



ТЕСТУВАННЯ РОЗРОБЛЕНОЇ ПРОГРАМИ

Робота програми в режимі
розпізнавання



The screenshot displays a game engine interface. On the left, a 3D scene shows a character in a dark, stone-walled environment. The character is labeled with a probability value: "vampire : 56.556". To the right, a UI overlay shows event logs for "Event 66" and "Event 67". Each event log lists three items with their probabilities and positions.

Event 66 [01:44:01]

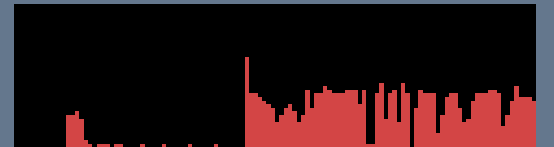
- 1. MONSTER => vampire PROBABILITY => 56.586
POSITION => x = 572 y = 476
- 2. MONSTER => vampire PROBABILITY => 49.082
POSITION => x = 711 y = 518
- 3. MONSTER => vampire PROBABILITY => 43.8
POSITION => x = 739 y = 533

Event 67 [01:44:02]

- 1. MONSTER => vampire PROBABILITY => 56.556
POSITION => x = 572 y = 476
- 2. MONSTER => vampire PROBABILITY => 49.102
POSITION => x = 711 y = 518
- 3. MONSTER => vampire PROBABILITY => 43.806
POSITION => x = 739 y = 533

Зріст навантаження GPU

35% GPU Used



ТЕСТУВАННЯ РОЗРОБЛЕНОЇ ПРОГРАМИ В «БОЙОВОМУ» РЕЖИМІ

Вікно гри



Ігровий момент: атака цілі



Ігровий момент:
нейтралізація цілі



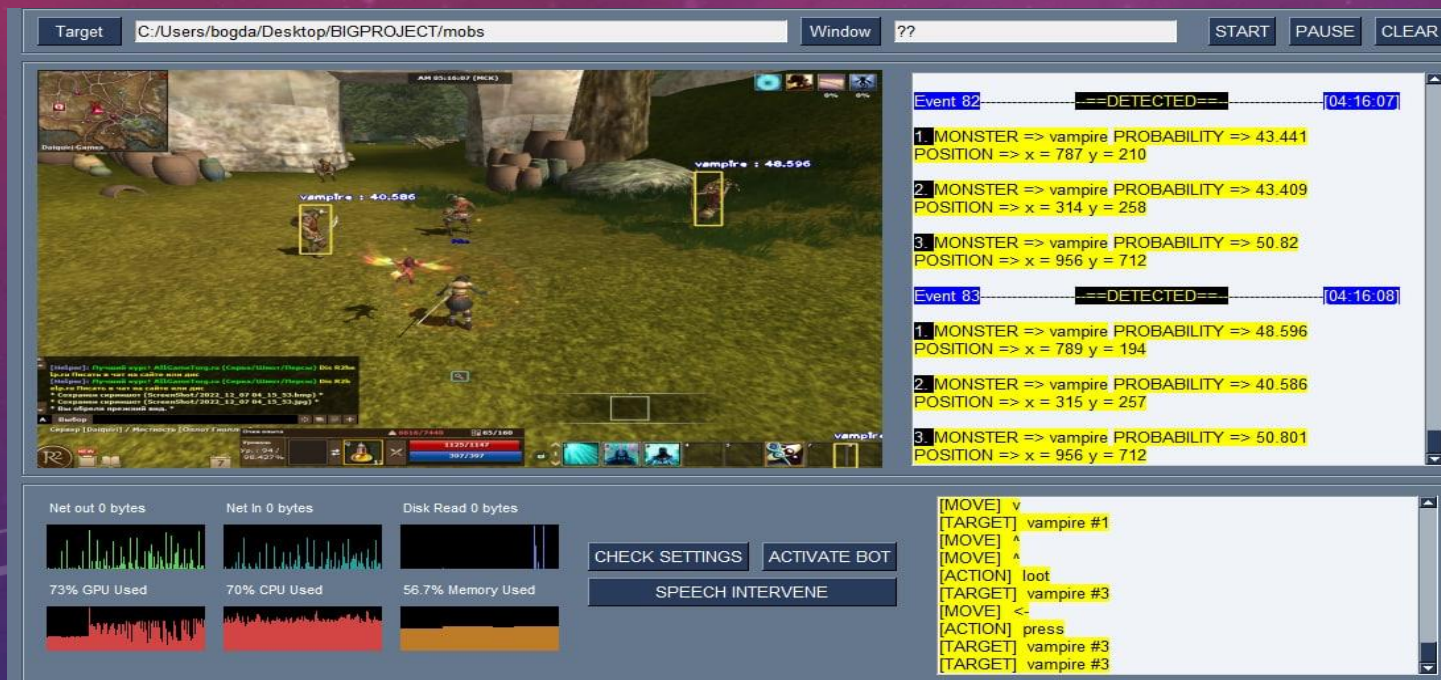


ТЕСТУВАННЯ РОЗРОБЛЕНОЇ ПРОГРАМИ В «БОЙОВОМУ» РЕЖИМІ

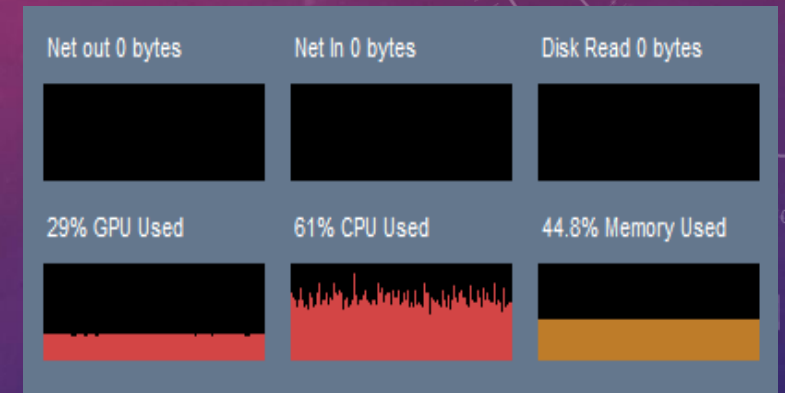
Запуск скрипта для збору даних

```
(venv) C:\Users\bogda\Desktop\BIGPROJECT\data>py monitor.py
```

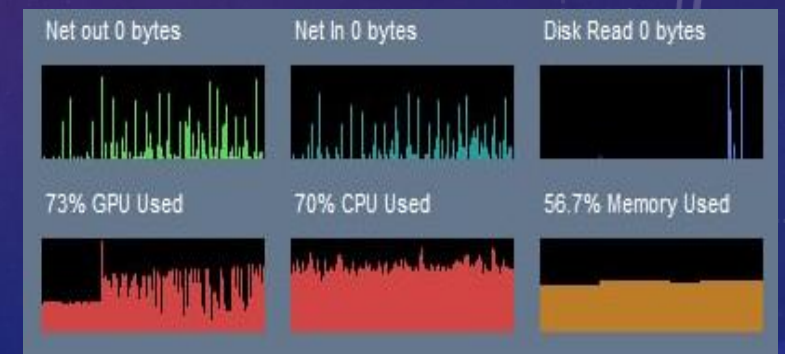
Бот виконує задачі



Моніторинг стану ПК з
увімкненою трансляцією



Активна робота бота в грі



ДОСЛІДЖЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

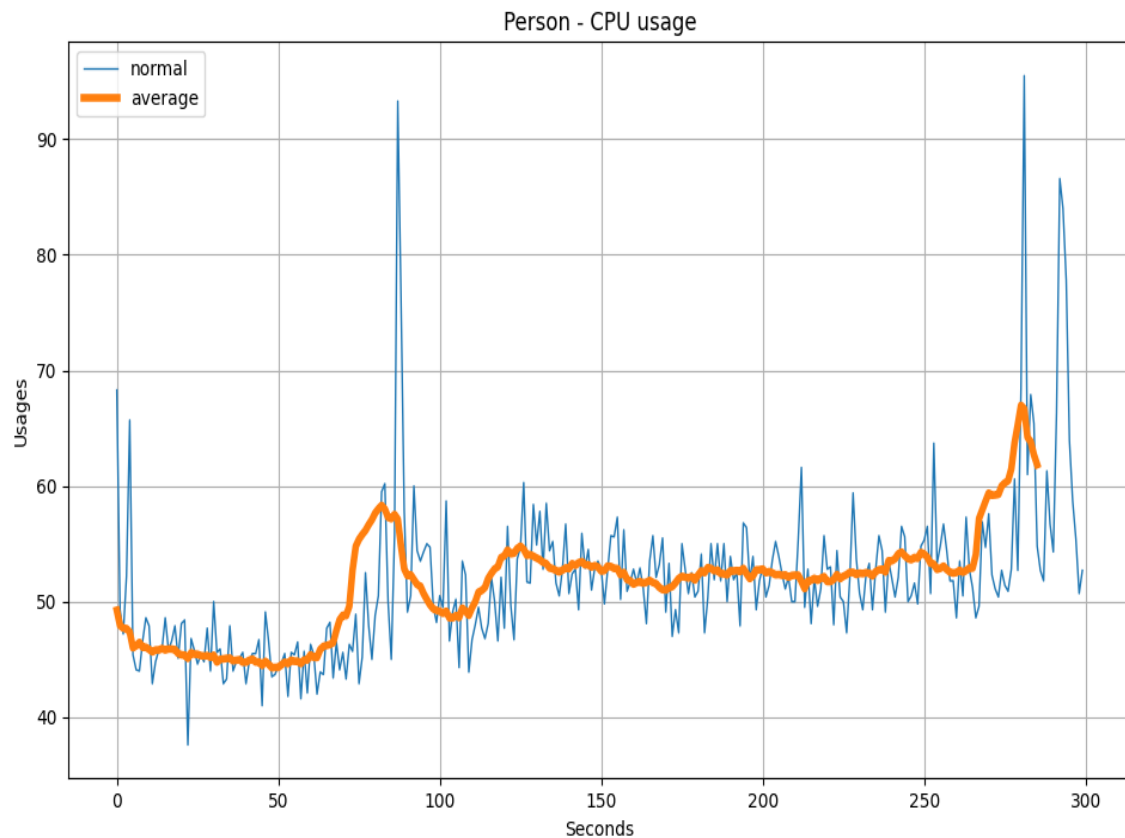
Порівняння завантаження ПК з і без бота

Ефективність виконання поставлених задач

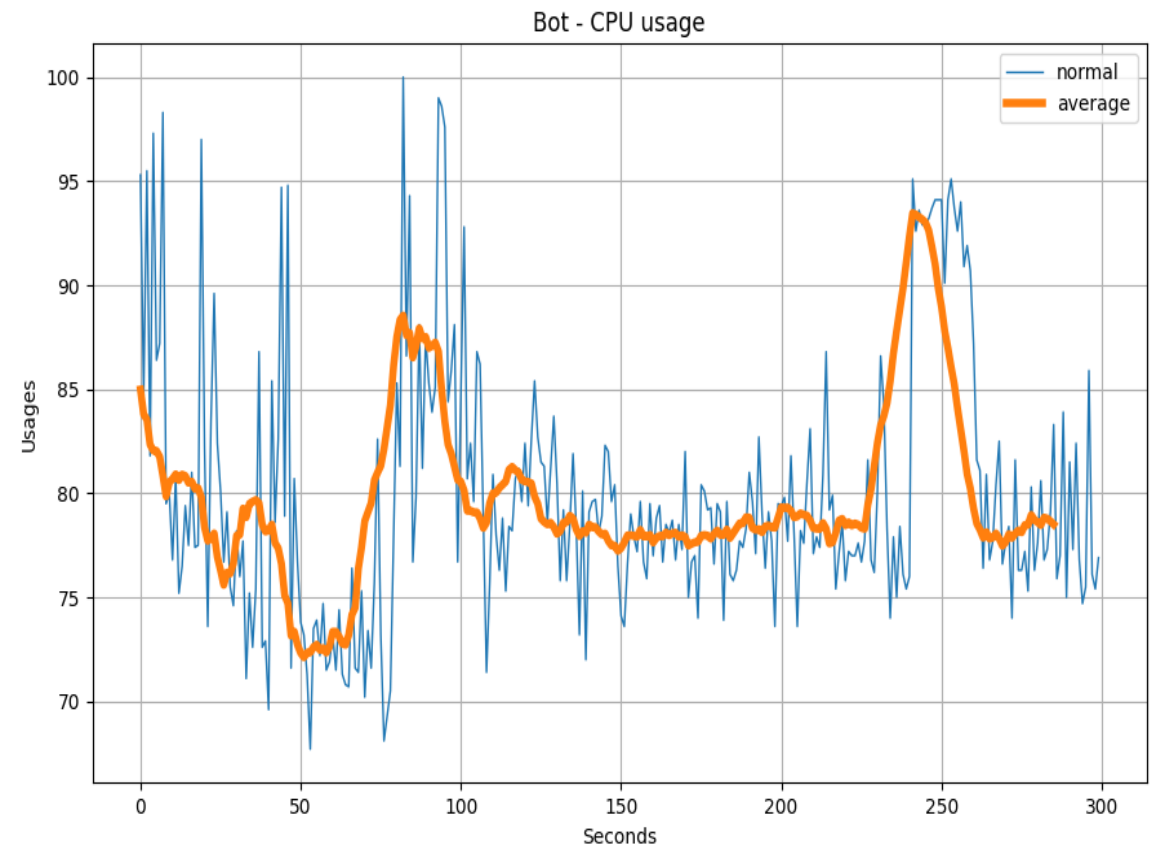
Оцінка оптимізації бота щодо попередньої розробки

ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ

Використання CPU при участі людини



Використання CPU при участі бота

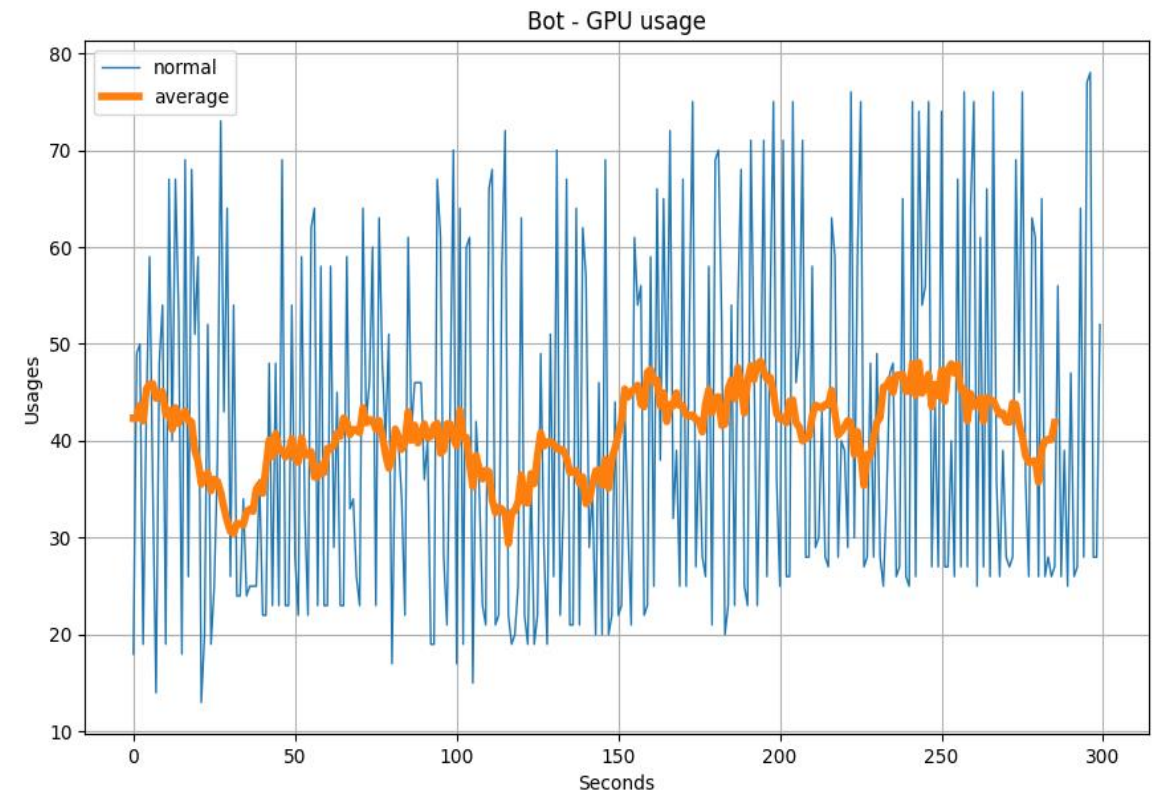


ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ

Використання GPU при участі людини

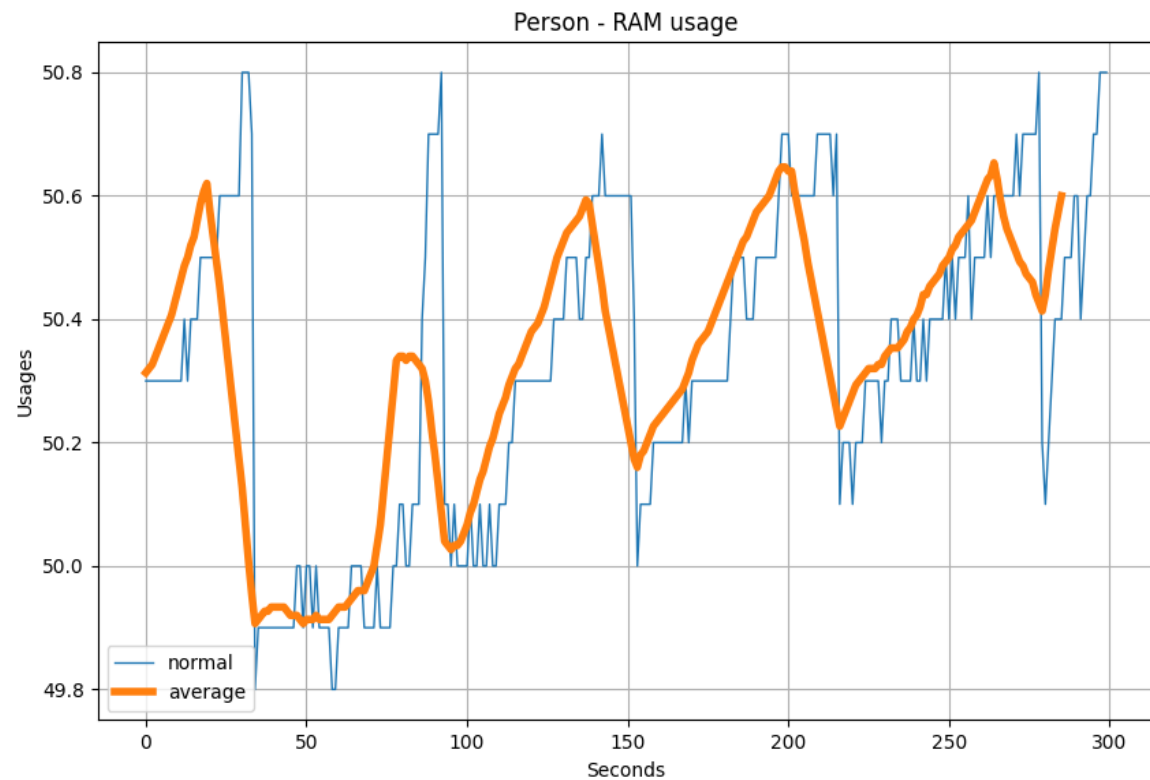


Використання GPU при участі людини

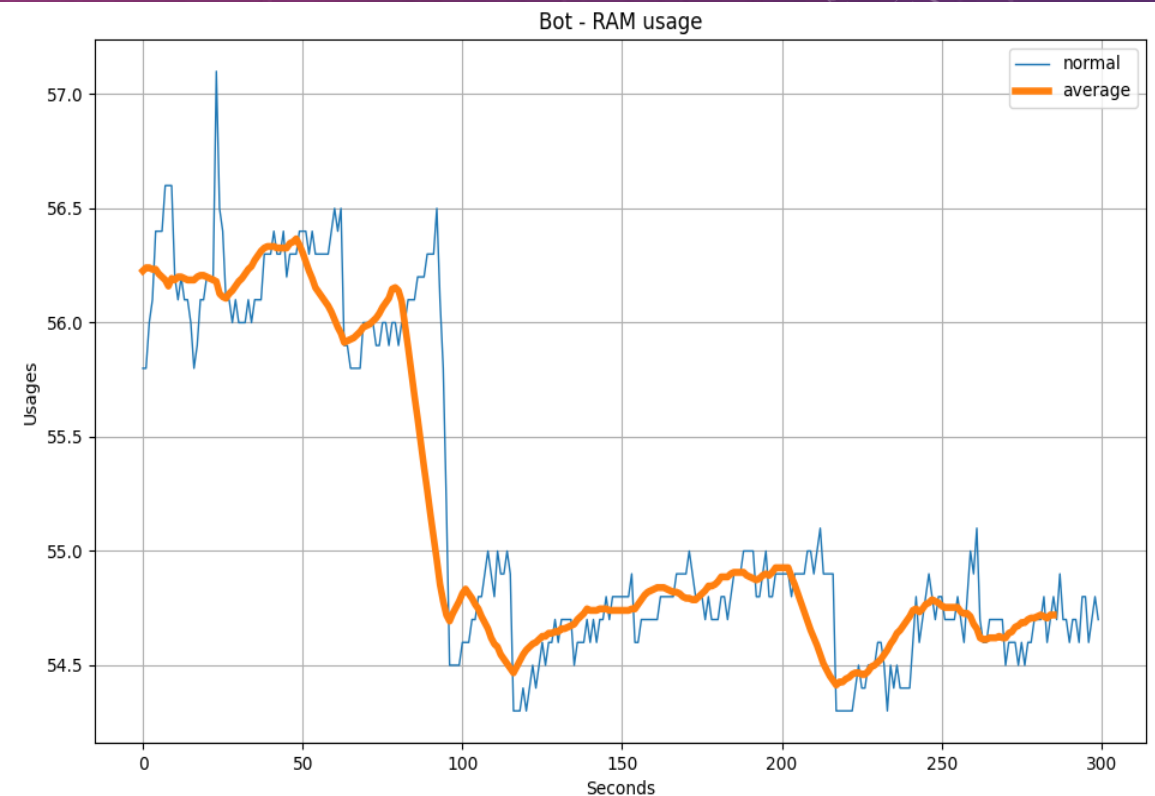


ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ

Використання RAM при участі людини



Використання RAM при участі бота

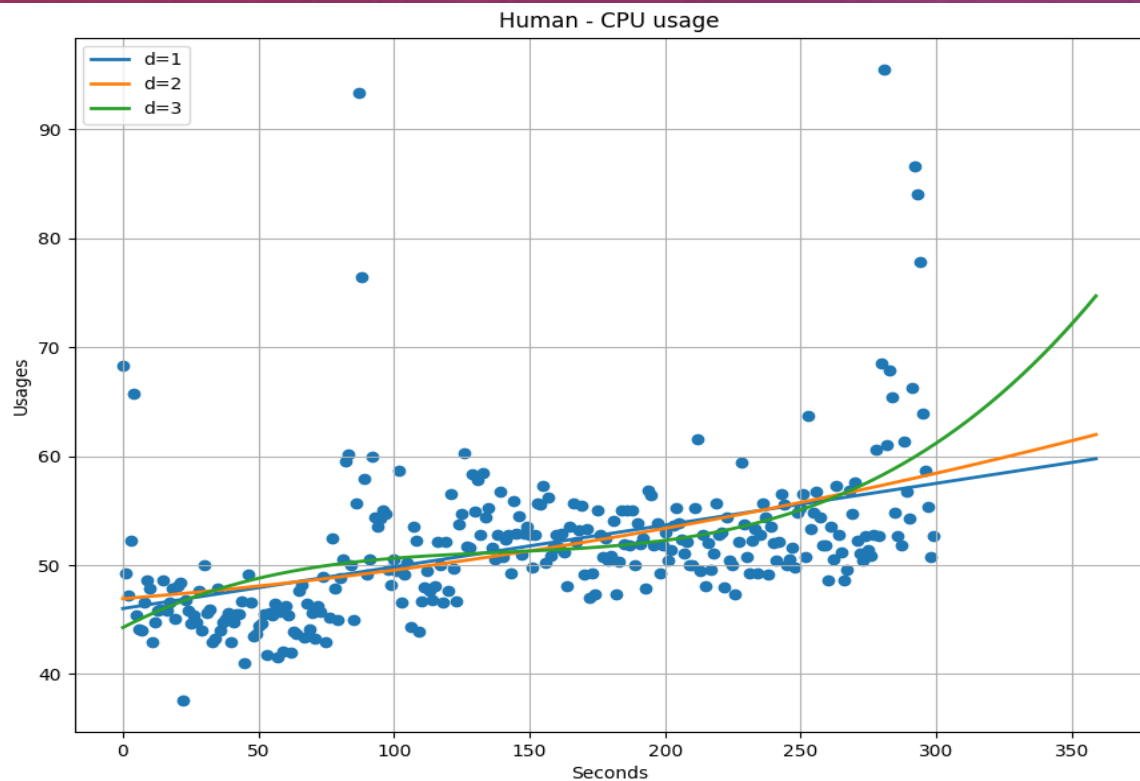


ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ

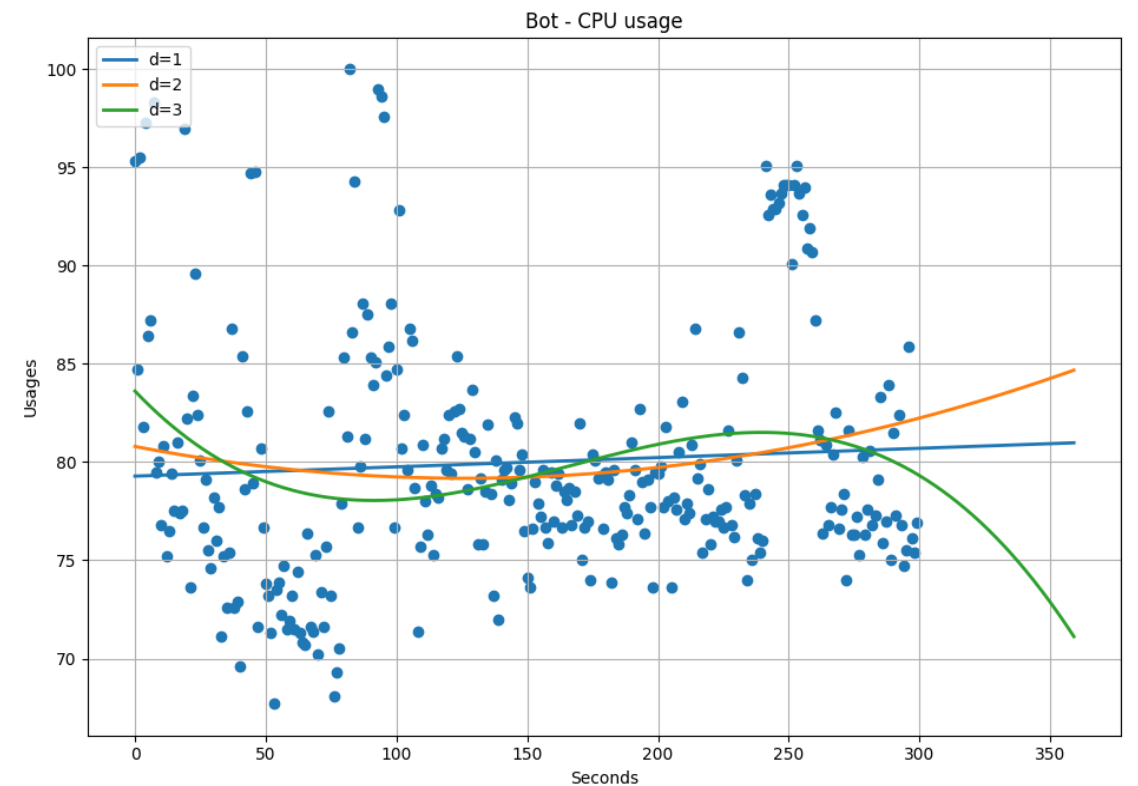
CPU		GPU		RAM	
Human	Bot	Human	Bot	Human	Bot
51.7 %	80.0 %	33.0 %	40.9 %	50.3 %	55.2 %
на 28.3 %		на 7.9		на 1.9 %	

ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ

Апроксимація даних навантаження
CPU при участі людини

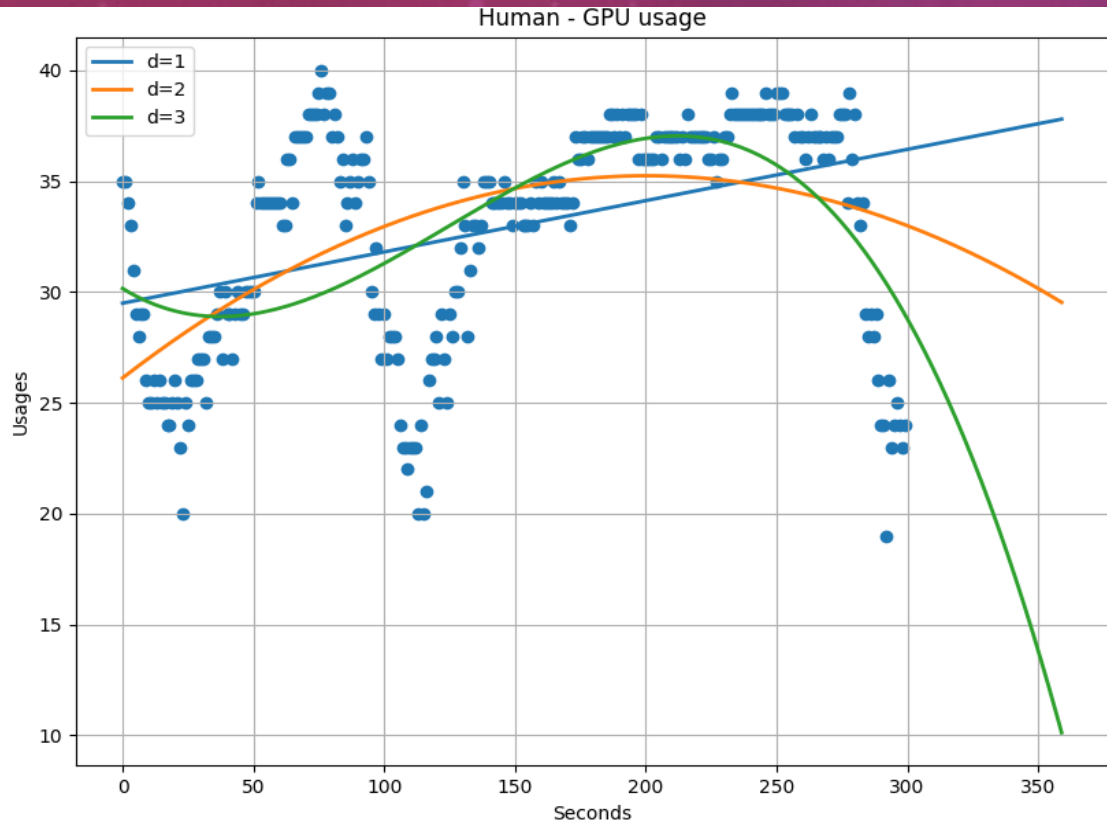


Апроксимація даних навантаження CPU
при участі бота

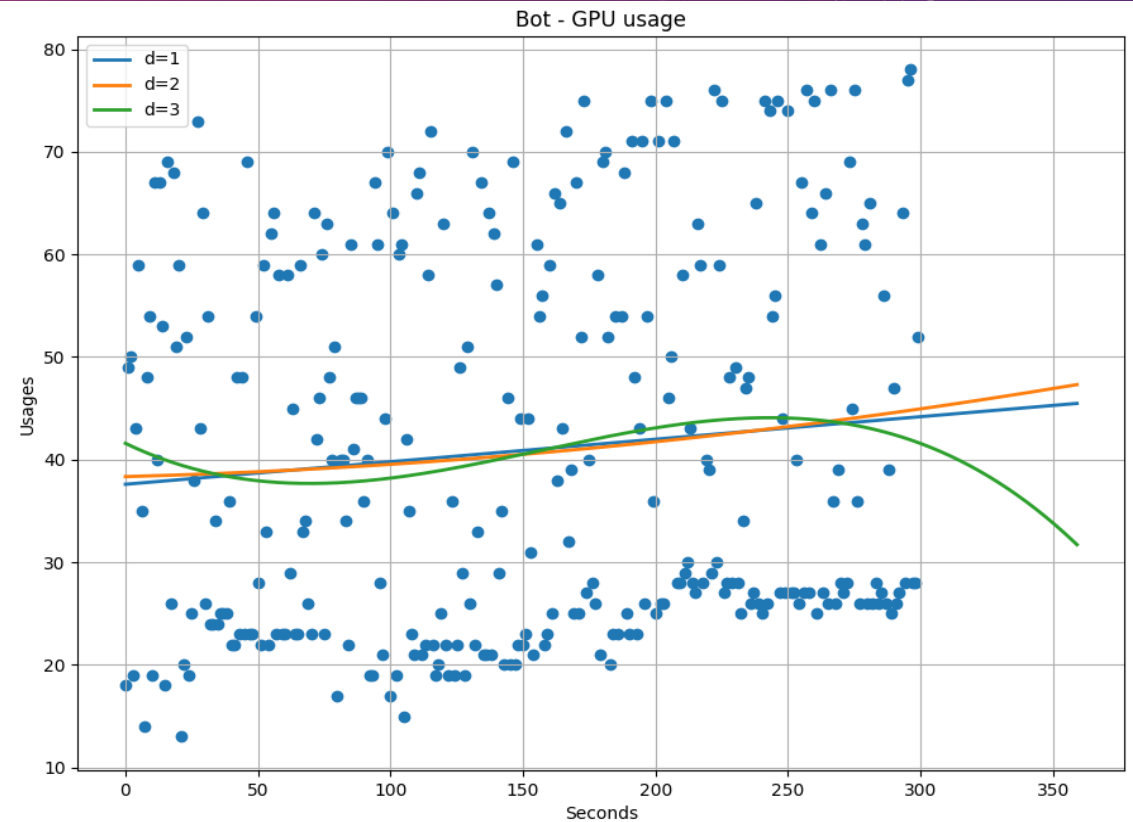


ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ

Апроксимація даних навантаження
GPU при участі людини

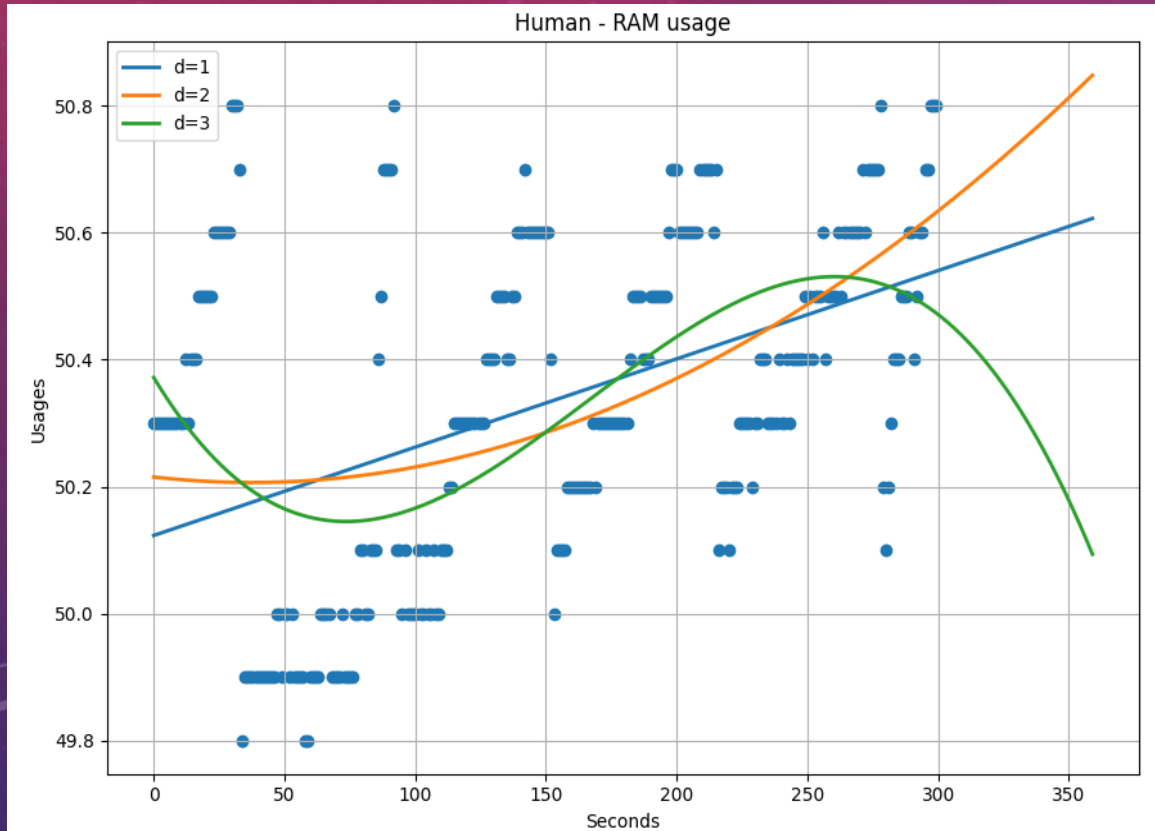


Апроксимація даних навантаження
GPU при участі бота

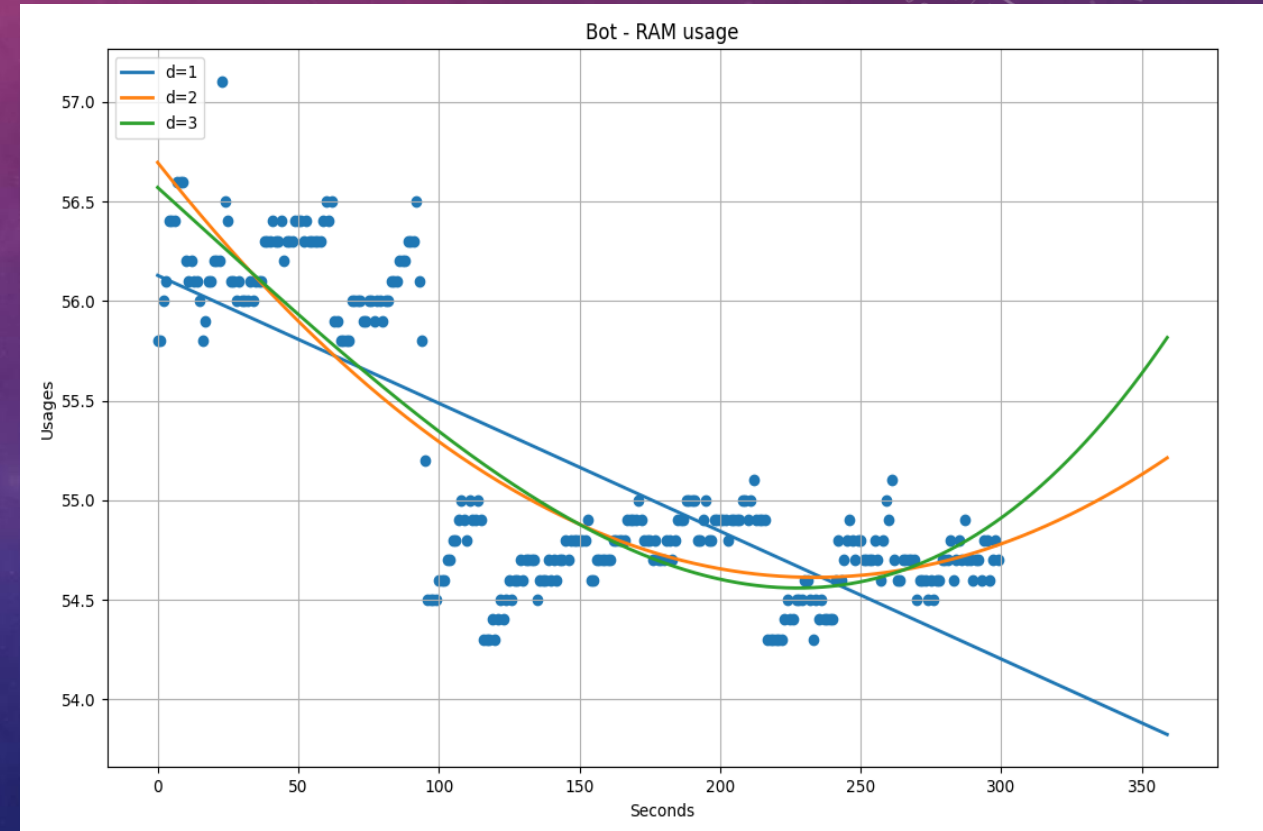


ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ

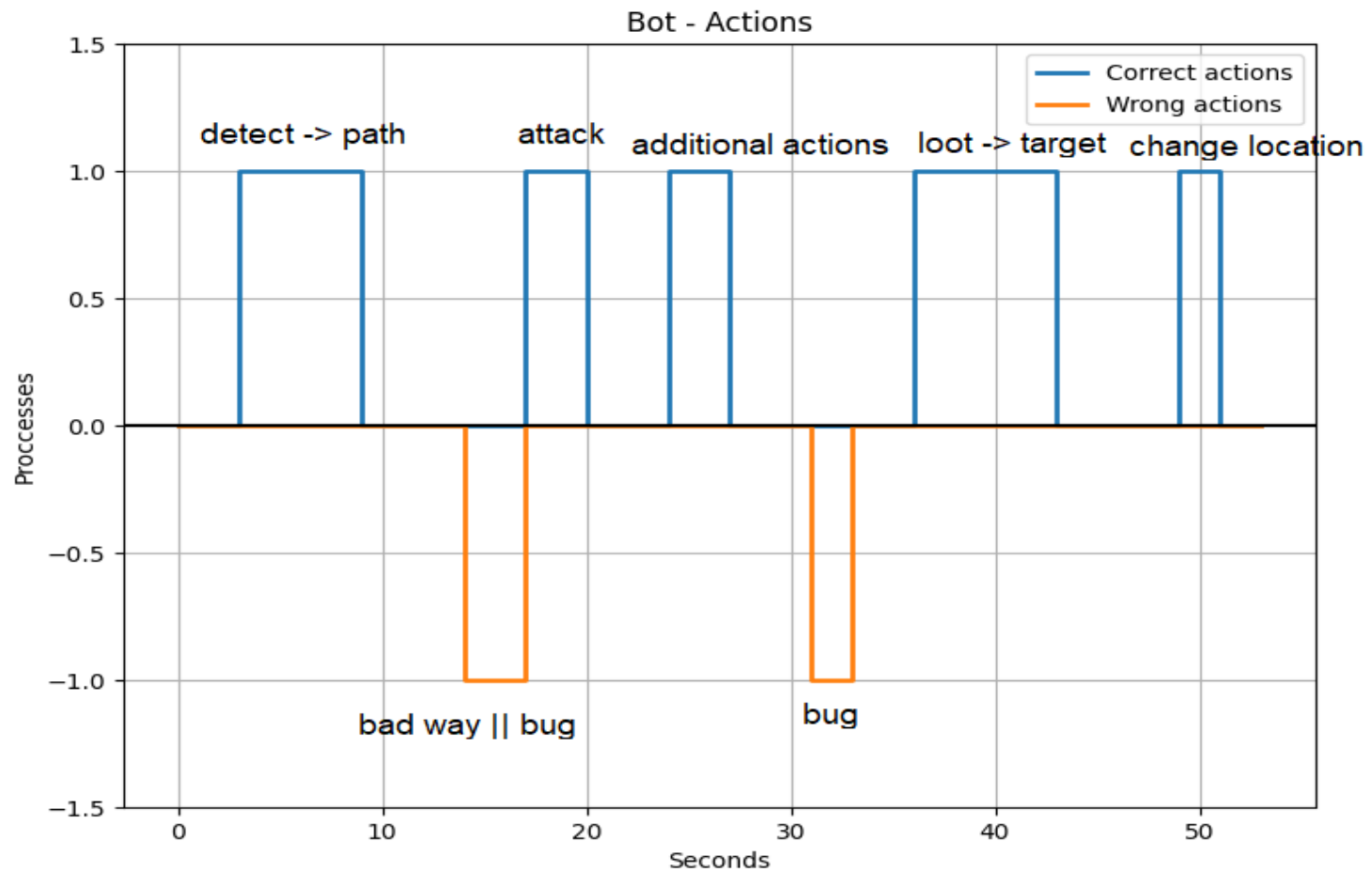
Апроксимація даних навантаження
RAM при участі людини



Апроксимація даних навантаження
RAM при участі бота



ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ



ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ

BOT 2			BOT 1		
CPU	GPU	RAM	CPU	GPU	RAM
80%	50%	55%	99%	25%	40%

ВИСНОВКИ

У заключенні, на основі отриманих даних, можна прийти до висновку, що програма успішно оптимізована — навантаження ефективно розподілено між ресурсами ПК (відклик/затримка програми зменшені). Отримано досвід в реалізації складних віконних додатків з інтеграцією функцій: обробки та аналізу відео/аудіоматеріалів.

Реалізовано віконний додаток з інтеграцією таких функцій, як: обробка зображень, розпізнавання зображення та мовлення, програмна взаємодія з ОС Windows та аналіз вихідних даних програми. Створено інтерфейс для: постійного моніторингу активності та управління програмою користувачем. Забезпечено крос-платформність застосунку, протестовано на роботоздатність функціонала.

Досліджено зміни показників навантаження системи ПК (в залежності від оптимізації), в тому числі часткове перенесення задач з CPU на GPU. Опрацьовано та проаналізовано вихідні дані, забезпечено візуальне представлення.



ДЯКУЮ ЗА УВАГУ!

СТУДЕНТ ГРУПИ ФЕІМ-22 ТОВКАЧ Б. М.