

Галицький коледж імені В'ячеслава Чорновола
відділення комп'ютерних та видавничих технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділенням
комп'ютерних та видавничих
технологій

Чубей О.О. / _____ /

підпис

« ____ » _____ 2021 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту
освітньо-кваліфікаційного рівня «молодший спеціаліст»
зі спеціальності 122 «Комп'ютерні науки»

на тему: «Керування персональним комп'ютером рухами очей»

Студентка групи К-47

Касперська Л.А.

(підпис)

Керівник проєкту

Павлюс В.П.

(підпис)

Консультанти:

з техніко-економічного

обґрунтування

Меленчук Л.І.

(підпис)

нормоконтролер

Кульчинська Н.З

(підпис)

Тернопіль - 2021

Галицький коледж імені В'ячеслава Чорновола
відділення комп'ютерних та видавничих технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділенням
комп'ютерних та видавничих
технологій

Чубей О.О. / _____ /

підпис

« ___ » _____ 2020 р.

ЗАВДАННЯ

на дипломне проектування
на здобуття освітньо-кваліфікаційного рівня «молодший спеціаліст»
студентці Касперській Лілії Андріївній

(прізвище, ім'я та по-батькові студента)

1. Тема проєкту _____

затверджено наказом по коледжу від “ ___ ” _____ 202_ р., № ___

2. Термін здачі студентом завершеного проєкту “ ___ ” _____ 202_ р.

3. Вихідні дані до проєкту _____

4. Перелік питань, які повинні бути розроблені в проєкті:

а) основна частина _____

б) техніко-економічне обґрунтування _____

5. Перелік графічного матеріалу _____

6. Консультанти проєкту: _____

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко-економічного обґрунтування	(вчена ступінь, звання П.І.Б. консультанта)		

КАЛЕНДАРНИЙ ПЛАН
дипломного проєктування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до дипломного проєктування	16.11.20	30.11.20
2.	Огляд типових рішень та написання відповідного розділу ПЗ	01.12.20	26.01.21
3.	Дослідження технологій реалізації та написання відповідного розділу ПЗ	27.01.21	15.02.21
4.	Розробка функціональних вимог до проєкту та робота над структурою програмного продукту. Написання відповідного розділу ПЗ	15.02.21	02.03.21
5.	Встановлення та налаштування середовища реалізації та написання відповідного розділу ПЗ	02.03.21	16.03.21
6.	Проєктування програмного засобу та написання відповідного розділу ПЗ	16.03.21	16.04.21
7.	Реалізація та налаштування програмного засобу на написання відповідного розділу ПЗ	17.04.21	03.05.21
8.	Доопрацювання модулів	03.05.21	17.05.21
9.	Опрацювання економічного розділу дипломного проєкту та оформлення спеціального розділу	17.05.21	18.06.21
10.	Тестування та налагодження програмного продукту та написання відповідного розділу ПЗ	18.05.21	04.06.21
11.	Робота над оформленням пояснювальної записки	04.06.21	11.06.21
12.	Попередній захист дипломного проєкту, доопрацювання	11.06.21	
13.	Підготовка до захисту дипломного проєкту	18.06.21	22.06.21
14.	Захист дипломного проєкту	23.06.21	23.06.21

7. Дата видачі “ ___ ” _____ 2020р. Керівник _____/

Завдання прийняв до виконання _____/

Реферат

Керування персональним комп'ютером рухами очей. Дипломний проєкт. Касперська Лілія Андріївна. Галицький коледж імені В'ячеслава Чорновола, відділення комп'ютерних та видавничих технологій. Спеціальність 122 «Комп'ютерні науки». ГК, 2021. Сторінок – 52, рисунків – 28, додатків – 1.

Об'єкт дослідження – програми для керування персональним комп'ютером рухами очей, розпізнавання обличь.

Метою даного дипломного проєкту є створення програми для керування ПК за допомогою рухів очей. Це допоможе людям з обмеженими можливостями використовувати персональний комп'ютер, а також урізноманітнить способи його використання рештою користувачів.

В дипломному проєкті потрібно добре продумати систему так як вона орієнтована на осіб з інвалідністю. Вона повинна бути простою і зрозумілою, враховувати всі потреби користувачів, легкою у встановленні. Система повинна використовувати наявні у користувача пристрої. Це може бути вбудована камера ноутбука чи персонального комп'ютера, або веб-камера. Також варто врахувати людські рефлексії, а саме – кліпання очима. Рефлекторно людина кліпає з невеликим інтервалом, приблизно кожні 2 – 10 секунд, тому потрібно виключити можливість виконання якоїсь дії під час цього. Так як дана програма буде керувати курсором миші, варто забезпечити можливість вимкнення/ввімкнення даної функції.

Для реалізації було використано середовище розробки Visual Studio Code, мову програмування Python, бібліотеку OpenCV.

Результатом розробки стала програма для керування персональним комп'ютером рухами очей.

КЕРУВАННЯ ПЕРСОНАЛЬНИМ КОМП'ЮТЕРОМ,
РОЗПІЗНАВАННЯ РУХІВ ОЧЕЙ, PYTHON, OPENCV, КОМП'ЮТЕРНЕ
БАЧЕННЯ.

Abstract

Personal computer control by eye movements. Diploma project. Kasperska Lilia. Galytsky College named after Viacheslav Chornovil, Department of Computer and Publishing Technologies. Specialty 122 "Computer Science". GK, 2021. Pages – 52, figures – 28, appendixes – 1.

Object of study – programs for controlling a personal computer by eye movements.

The purpose of this project is to create a program for controlling a PC with eye movements. This will help people with disabilities to use a personal computer, as well as diversify the way the other users use it.

The diploma project requires a well-thought-out system as it is aimed at people with disabilities. It should be simple and clear, take into account all the needs of users, easy to install. The system must use the user's existing devices. This can be a laptop or PC built-in camera or a webcam. You should also take into account human reflexes, namely blinking eyes. Typically, a person blinks at short intervals, approximately every 2 – 10 seconds, so need to exclude the possibility of performing any action during this. Since this program will control the mouse cursor, need to be able to disable / enable this feature.

Visual Studio Code development environment, Python programming language, OpenCV library were used for implementation.

The result is a ready-to-use program for a personal computer control by eye movements.

PERSONAL COMPUTER CONTROL, EYE MOVEMENTS
RECOGNITION, PYTHON, OPENCV, COMPUTER VISION.

ЗМІСТ

Вступ	7
1 Аналіз існуючих рішень та постановка завдання.....	8
1.1 Обґрунтування доцільності створення системи.....	8
1.2 Огляд існуючих рішень.....	8
1.3 Постановка задачі	20
2 Проєктування системи	21
2.1 Формалізація вимог до системи	21
2.2 Проєктування структури системи.....	22
2.3 Проєктування алгоритму роботи системи	22
3 Реалізація та тестування системи.....	25
3.1 Вибір засобів реалізації.....	25
3.2 Програмна реалізація	28
3.3 Тестування готового продукту.....	31
4 Техніко-економічне обґрунтування.....	39
4.1 Аналіз ринку.....	39
4.2 Розрахунок витрат на проєктування.....	40
4.3 Обґрунтування необхідності розробки	43
Висновки.....	44
Перелік джерел посилання.....	45
Додатки	47

					<i>ДП. КН 21.437.16.000 ПЗ</i>		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Касперська Л.А			Літ.	Арк.	Аркушів
Перев.		Павлюс В.П.				5	52
Рецензет.		Посвятовська О.Б.			<i>ГК.КВТ.К-47</i>		
Н. Контр.		Кульчинська Н.З.					
Зав. від.		Чубей О.О.					
Керування персональним комп'ютером рухами очей							

СКОРОЧЕННЯ І УМОВНІ ПОЗНАКИ

ПК – Персональний комп'ютер

ПЗ – Програмне забезпечення

ОЗП – Оперативна пам'ять

VS – Visual Studio

ISO – Міжнародна організація зі стандартизації

ANSI – Американський інститут національних стандартів

USB – Universal Serial Bus

IDEF0 – Integration Definition for Function Modeling

IDEF3 – Integrated DEFinition for Process Description Capture Method

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

З кожним днем у нашому житті з'являється все більше і більше техніки. Вона появляється на робочих місцях, на вулицях, у наших домівках. Нові винаходи створюються з метою покращити та спростити життя людям.

Хоча є і негативні сторони цих модернізацій. Деякі люди лишуються своїх робочих місць. Також є певна категорія осіб з обмеженими можливостями, які просто не зможуть скористатися новими винаходами. Це призводить до їхнього віддалення від суспільства. Тому важливою є розробка нових пристроїв, якими вони зможуть повноцінно користуватися.

Сучасні технології повинні використовуватись для допомоги людям з обмеженими фізичними можливостями повноцінно реалізовувати себе в повсякденному житті. Наприклад, людям з вадами опорно-рухового апарату замінити відсутні кінцівки спеціальними протезами. Або ж адаптувати наявні пристрої для цих людей. Наприклад, для персональних комп'ютерів чи мобільних пристроїв таким рішенням може бути реалізація спеціального ПЗ.

Метою даного дипломного проєкту є створення програми для керування ПК за допомогою рухів очей. Це допоможе людям з обмеженими можливостями використовувати персональний комп'ютер, а також урізноманітнить способи його використання рештою користувачів.

Завданням дипломного проєкту є створення простої та зручної у користуванні програми керування ПК рухами очей. Простота використання є важливою вимогою, так як вона має бути орієнтована на людей будь-якого віку.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Обґрунтування доцільності створення системи

Чимало людей з обмеженими можливостями не мають змоги повноцінно використовувати персональний комп'ютер загальноприйнятим способом: натискаючи кнопки на клавіатурі і рухаючи мишкою. Основна причина цього – травми верхніх кінцівок. Таким людям дуже важко знайти роботу, яка пов'язана з використанням комп'ютерної техніки. Часто це може призводити до дистанціювання таких осіб від сучасного «цифрового» суспільства.

Допомогти в цій ситуації можуть спеціальні апаратно-програмні системи керування ПК без допомоги рук. Сьогодні на ринку існує чимало таких пристроїв, але найбільшим їхнім недоліком є доволі висока ціна, через яку не кожен може собі їх дозволити. Крім цього, як покаже аналіз нижче, більшість таких систем орієнтовані, в першу чергу, на ігрову сферу, а не на допомогу людям з обмеженими можливостями.

1.2 Огляд існуючих рішень

Складовою відомої на сьогодні операційної системи Windows 10 є функція Eye Control (рис. 1.1). Вона націлена на допомогу користувачам з порушеннями моторно-рухової системи в користуванні комп'ютером. Головною метою компанії Microsoft було зробити дану операційну систему максимально доступною для користування нею без традиційних засобів управління. Для активації даної функції необхідно відкрити «Налаштування», перейти у «Спеціальні можливості». В розділі «Взаємодія» вибрати «Керування очима».

Дана функція дозволяє використовувати технологію:

- відслідковування погляду для керування мишею;
- вводу з екранної клавіатури;
- спілкування з людьми за допомогою перетворення тексту у звук.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

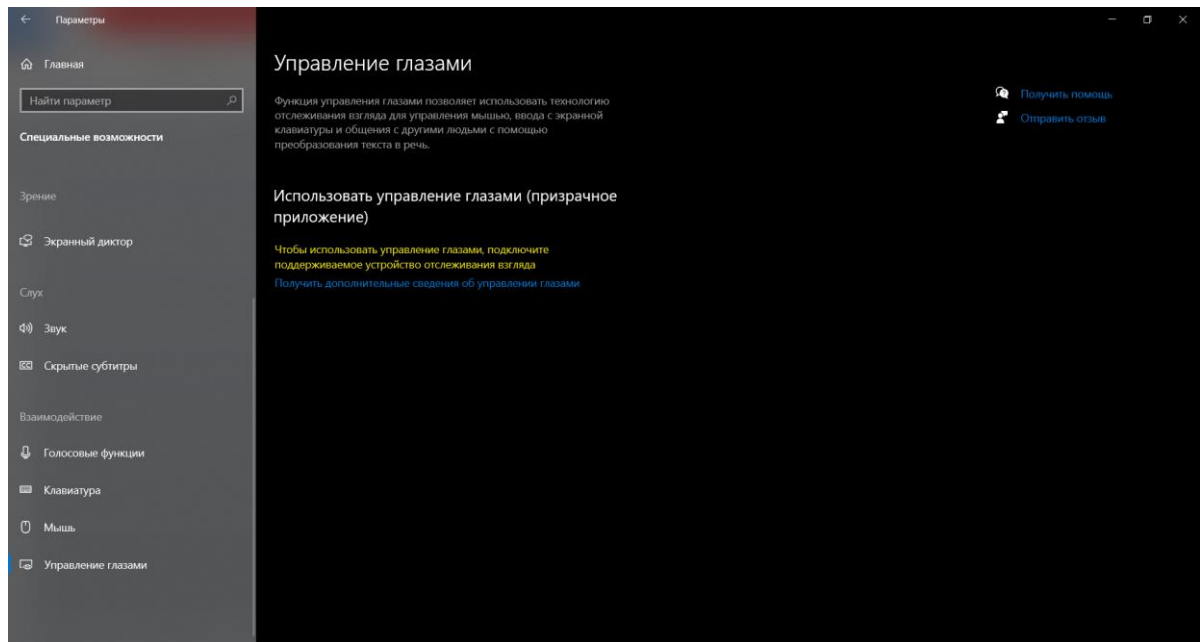


Рисунок 1.1 – Функція керування очима в ОС Windows

Людина може надрукувати текст за допомогою зручної віртуальної клавіатури, затримавши погляд на буквах та інших символах. Для користувачів Eye Control були розроблені підказки, які спрощують процес керування комп'ютером.

Функціонує ця технологія з допомогою спеціальної камери Eye Tracker компанії Tobii.

Tobii Eye Tracker 4C (рис. 1.2) – це унікальний гаджет. Ідея трекера полягає у можливості контролювати ігровий процес за допомогою погляду.

На даний момент даний гаджет знятий із виробництва, а його орієнтована ціна становила 250\$.

В основу покладено метод відстеження контрасту між зіницею і райдужною оболонкою, який виникає при інфрачервоному підсвічуванні очей.

Основні компоненти системи:

- інфрачервоні світлодіоди;
- камера із високою роздільною здатністю;
- ASIC-чіп для розрахунків.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.2 – Tobii Eye Tracker 4C

Світлодіоди 90 раз в секунду «обстрілюють» очі інфрачервоним випромінюванням, створюючи відблиски на сітківці і зіниці. Камера робить знімки, а чіп аналізує їх: обчислює значення різних параметрів на зразок взаємного розташування відблисків і таким чином визначає напрямок погляду.

Трекер пройшов сертифікацію за європейським стандартом EN 62471, який регламентує використання світлового випромінювання в світлодіодних джерелах освітлення. Розміри пристрою 17×15×335мм, маса 95г. Максимальна діагональ монітора ПК складає 27" при співвідношенні 16:9 або 30" при співвідношенні 21:9. Трекер працює на відстані 50-95см, розміри простору, який відслідковується 40×30см на відстані 75см. Підключається до одного роз'єму USB 2.0, довжина кабеля 80см. Сумісний з Windows 7, 8.1 і 10, але тільки 64-бітними. Інші системні вимоги: процесор Intel Core i5 або Core i7 з частотою 2,0 ГГц, 8 Гбайт ОЗП.

Гаджет упакований дуже ретельно (рис. 1.3). Всередині розташувалося наступне:

- Tobii Eye Tracker 4C з вбудованим кабелем довжиною 80 см;
- два кріплення для встановлення (одне запасне);

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

- інструкція зі встановлення;
- гарантійний талон і відомості про сертифікацію;
- USB-подовжувач, 1м.

Кріпиться трекер до монітора за допомогою 8-сантиметрової магнітної смужки – однією стороною вона міцно хапається за виїмку в тильній частині пристрою, а на іншу, яка контактує з монітором, нанесена самоклеюча плівка (рис. 1.4). Відповідно, місце для кріплення вибирається один раз і надовго, в іншому випадку доведеться віддирати самоклеючу плівку від монітора.



Рисунок 1.3 – Упаковка Tobii Eye Tracker 4C

При підключенні Windows 10 розпізнає трекер як EyeChip. Потрібно завантажити пакет Tobii Eye Tracking Core Software з офіційного сайту. Після встановлення відбувається калібрування.

Користувач поглядом знищує розкидані по екрану точки, а система в цей час підлаштовує трекер під особливості очей – образно кажучи, знімає

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

мірку і накладає її на внутрішню анатомічну 3D-модель очей, яка містить інформацію про різні оптичні властивості окремих складових очного яблука. Отриманий результат можна оцінити в космічній міні-грі, яка пропонує збивати астероїди, цілячись в них поглядом.

Якщо комп'ютером користуються кілька людей, дозволяється створити окремий профіль для кожного.



Рисунок 1.4 – Розташування камери на моніторі

Пристрій також можна використовувати для входу в обліковий запис Windows 10 за умови використання Windows Hello. Також програма Tobii Eye Tracking пропонує кілька можливостей, згрупованих в чотири підменю.

Управління живленням (рис. 1.5). Коли користувач залишає поле зору камери, яскравість монітора знижується. Ця корисна функція працює не з усіма моніторами; у вкладці є вказівка про те, що «більшість дисплеїв ноутбуків підтримуються». Монітор ПК виходить із сплячого режиму в той час коли користувач сідає за комп'ютер.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

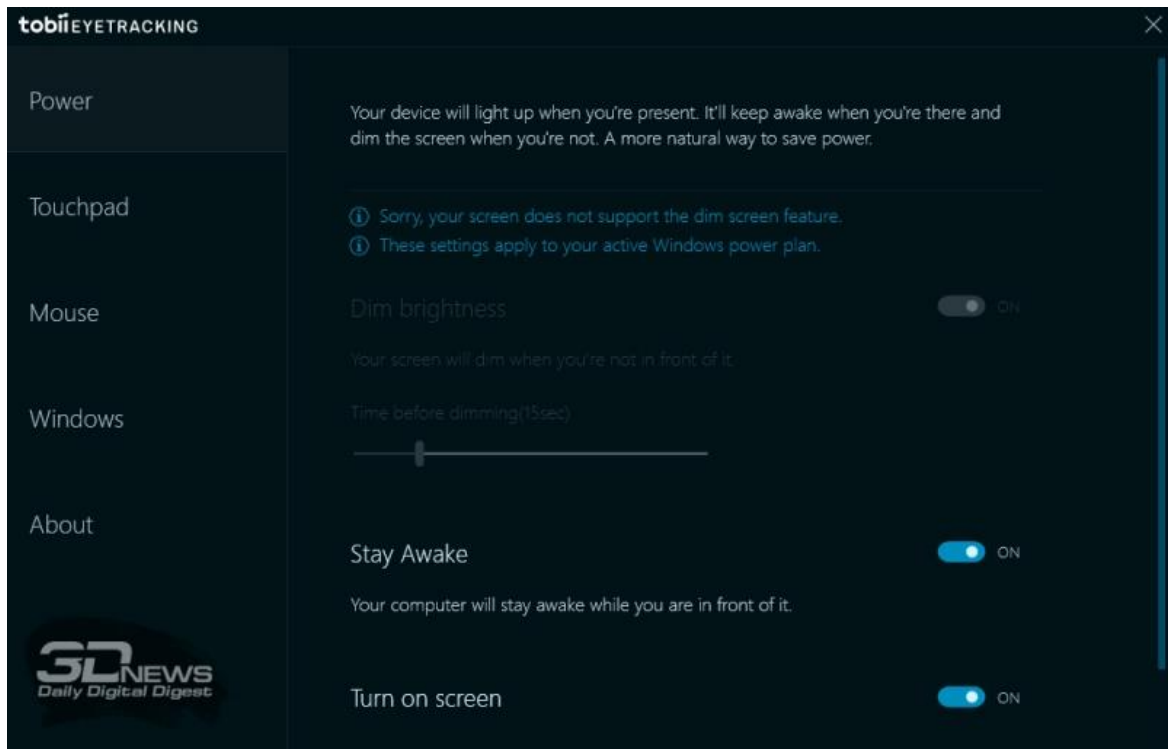


Рисунок 1.5 – Управління живленням

Робота з сенсорною панеллю ноутбука (рис. 1.6). Підтримуються тачпади Synaptics і Precision.

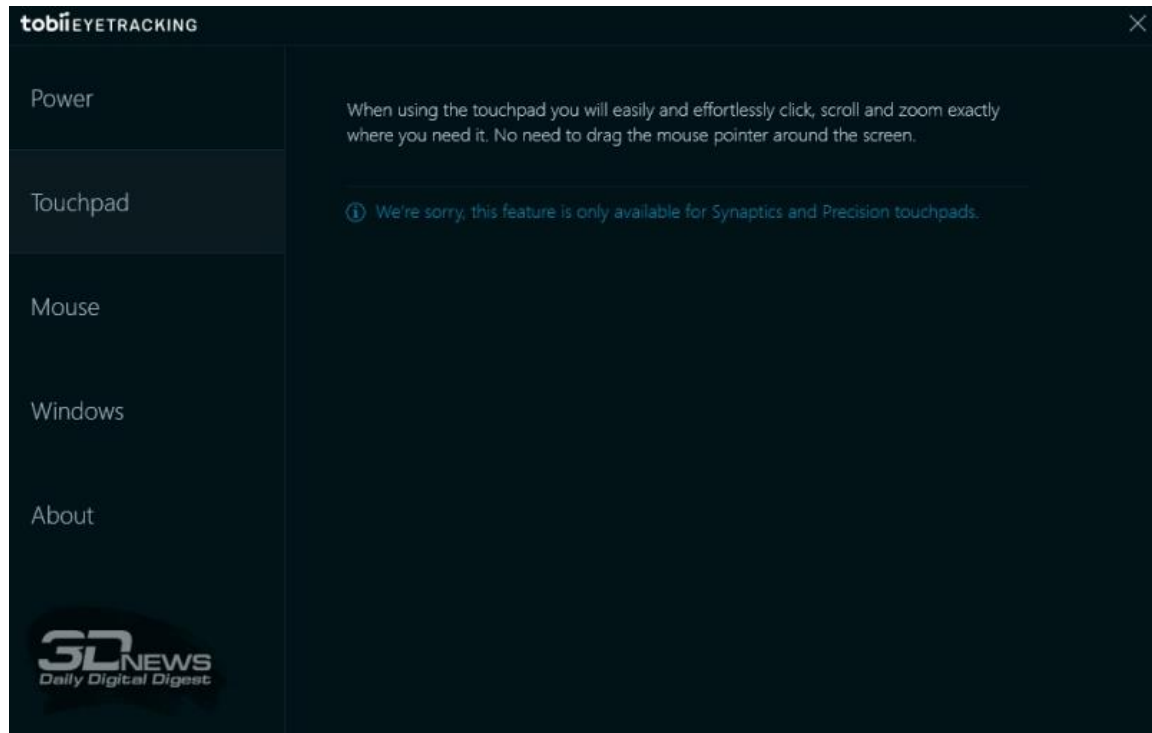


Рисунок 1.6 – Робота з сенсорною панеллю

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Робота з мишкою (рис. 1.7). Переміщення курсора слідом за поглядом:

- при переміщенні миші: невеликий рух руки – і курсор стрибає туди, куди користувач дивиться. Можна налаштувати поріг переміщення курсору, після якого функція спрацьовує (від 1 до 10 см);
- при натисканні на кнопку. Подивився, натиснув призначену кнопку – курсор перемістився на потрібне місце.

Обидва варіанти вимагають певного звикання, перебудови шаблону роботи з комп'ютером і вироблення нової взаємозв'язку між поглядом і рухами мишею.

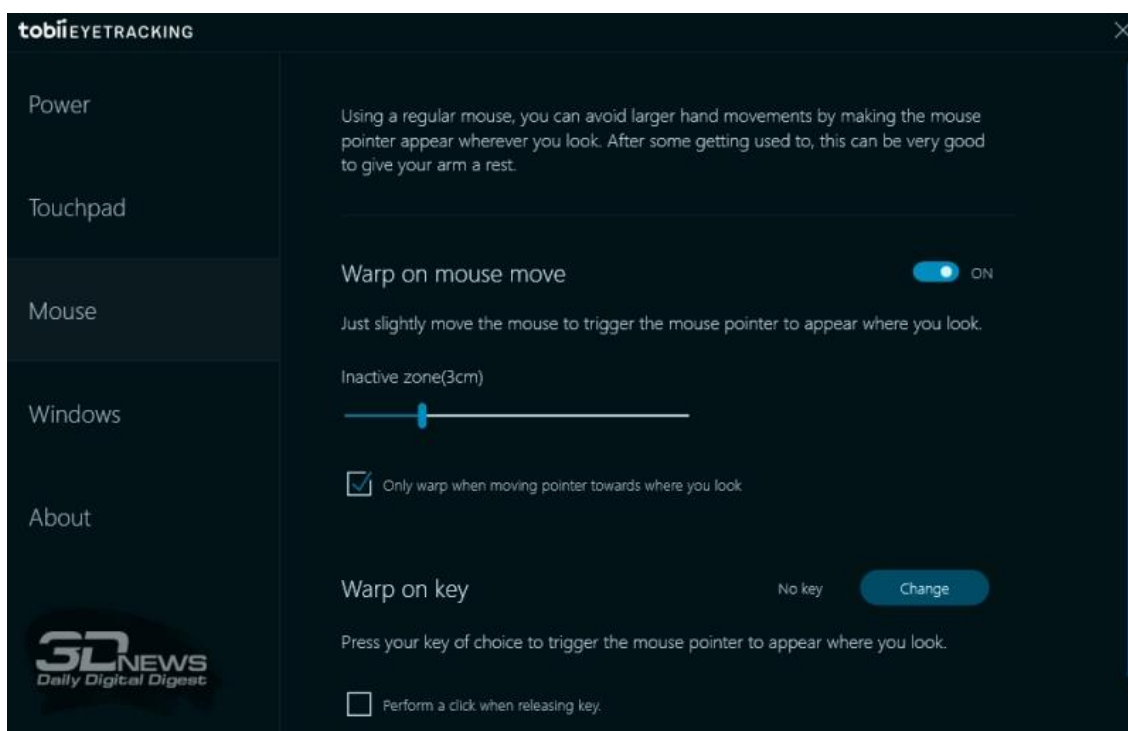


Рисунок 1.7 – Робота з мишкою

Робота з функціями Windows (рис. 1.8):

- перехід між вікнами за допомогою Alt + Tab;
- перехід між вікнами за допомогою Win + Tab;
- прикріплення вікон за допомогою Windows Snap Assist.

Є ще дві можливості, але вони в статусі «beta»:

- автоматичне приховування повідомлень (зникають після того, як користувач на них подивиться);
- виділення адресного рядка в Edge.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

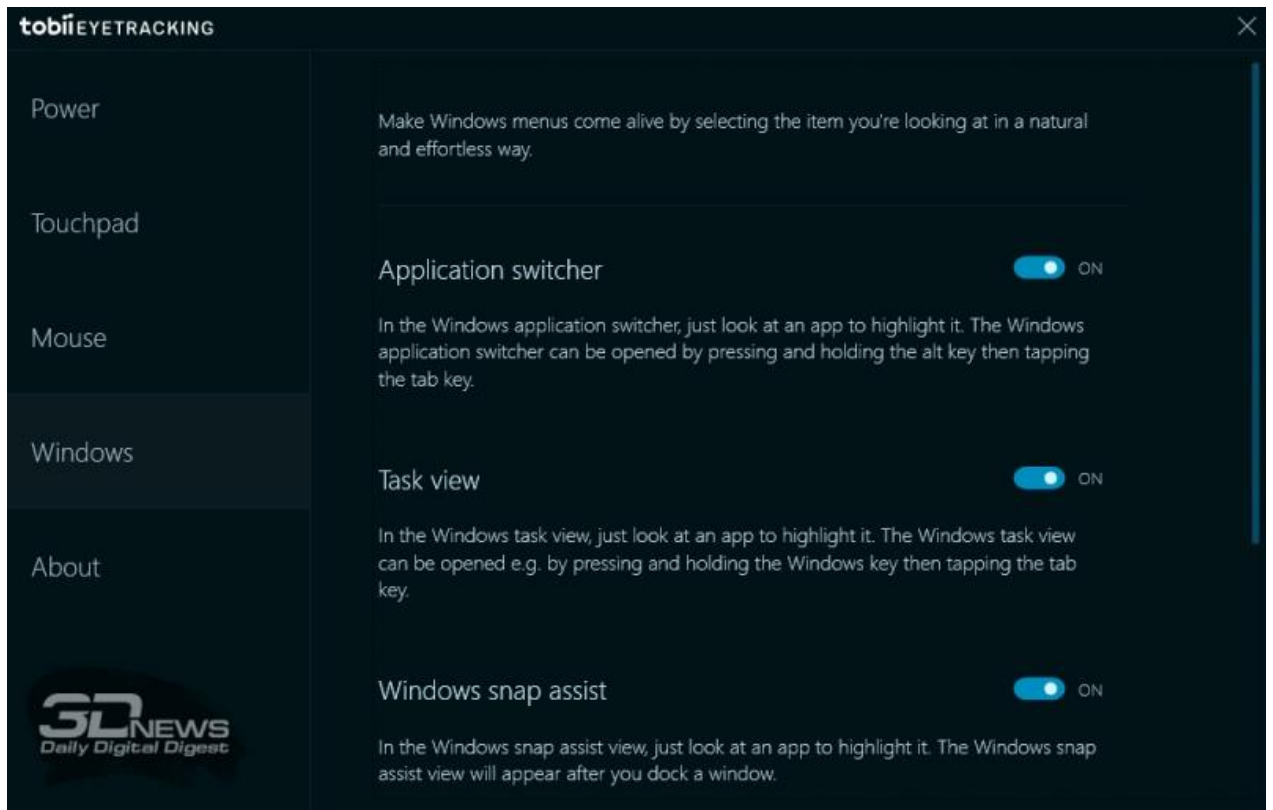


Рисунок 1.8 – Робота із функціями Windows

Окремим пунктом йде функція Gaze Trace, при активації якої на екрані з'являється «бульбашка», яка слідує за поглядом. Додатково можна скачати надбудову Tobii Streaming Gaze Overlay, яка розширює можливості цього інструменту.

Підбивши підсумок, можна зрозуміти, що в якості додаткової периферії для роботи в Windows у Tobii Eye Tracker 4С більше мінусів, ніж плюсів. Він більше використовується як периферія для ігор [11].

Актуальною версією є Tobii Eye Tracker 5 (рис. 1.9). Орієнтована ціна – 320\$. Tobii Eye Tracker 5, як і його попередник, створений для ігор на ПК [12].

В Tobii Eye Tracker 5 покращена чутливість: виготовлений спеціально для нього оптичний сенсор дозволив збільшити швидкість і чутливість, зменшена затримка. Завдяки широкому полю огляду сенсор легко зчитує кожен рух. Більш надійне відстеження голови: для того, щоб ще точніше відстежувати положення голови, був реалізований новий алгоритм.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.9 – Tobii Eye Tracker 5

Нове універсальне гнучке кріплення (рис. 1.10): даний пристрій став на 15% менший в порівнянні з попередньою версією. Тепер він укомплектований новим гнучким кріпленням, яке робилося з розрахунком на універсальність і адаптивність. Завдяки цьому стало набагато простіше прикріпити сенсор до зігнутих моніторів і екранів різних розмірів.

Портативність і надійність: завдяки тому, що Tobii Eye Tracker 5 зроблений з більш міцних і легких матеріалів і проектувався з прицілом на портативність, після установки кріплення сенсор можна легко знімати і встановлювати.

Реалізація нових потужних засобів забезпечення безпеки: гаджет дозволяє відмовитися у використанні паролів, так як сертифікований для роботи з Windows Hello як сканер біометричного розпізнавання осіб корпоративного рівня.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

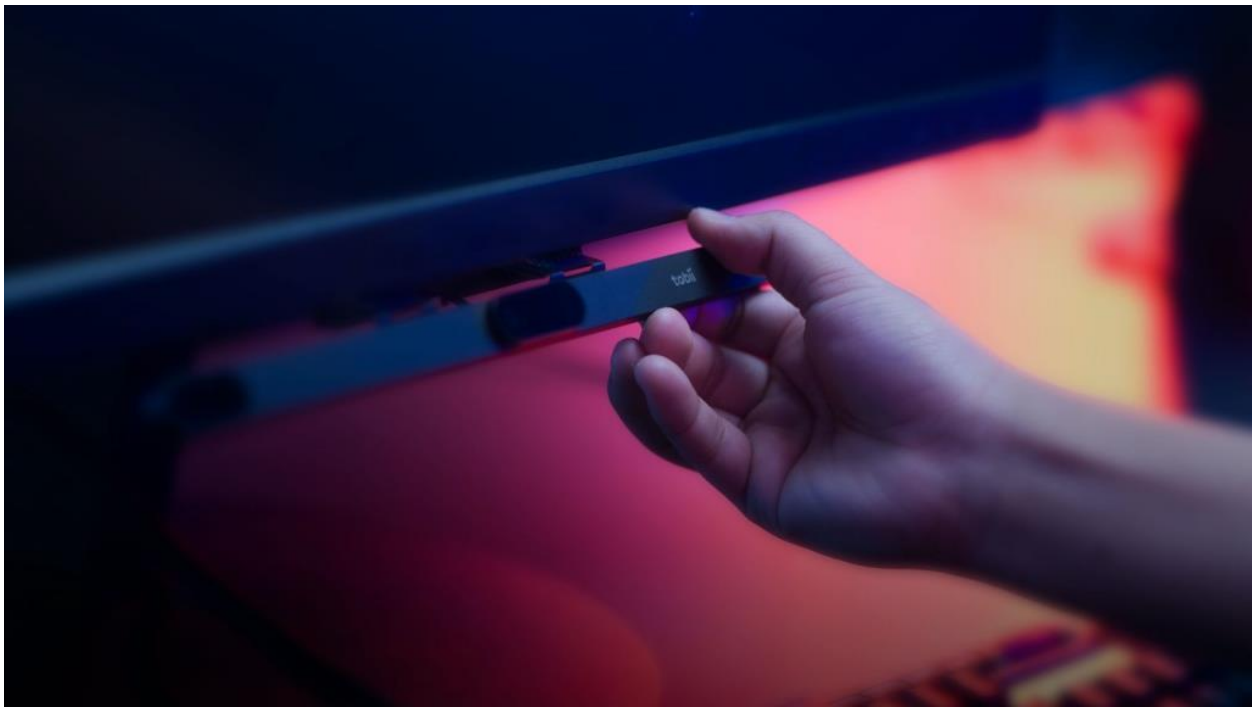


Рисунок 1.10 – Нове кріплення

Особливості Tobii Eye Tracker 5:

– технологія розумного сенсора: в основі системи – інфрачервоний оптичний сканер, спеціально спроектований під жорсткі параметри чутливості, точності і надійності. Коли включено відстеження, пристрій показує неймовірну продуктивність в іграх і тренуваннях, дозволяючи максимально покращити результативність;

– адаптованість до різних умов: прожектори Tobii Eye Tracker 5 випромінюють світло, близьке до інфрачервоного спектру, що дозволяє сенсору добре працювати в різних умовах;

– абсолютно новий спосіб використання Windows: взаємодіє з Windows 10 легко та інтуїтивно. Використовує такі функції, як "mouse warp", щоб телепортувати курсор миші туди, куди користувач дивиться, або перемикається між програмами і екранами за допомогою руху очей.

Периферійний пристрій легко приєднується до нижньої частини монітора, після чого можна підключити його до комп'ютера кабелем USB 2.0 (рис. 1.11).

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.11 – Eye Tracker 5

Технологія Tobii Eye Tracker дозволяє комп'ютеру відслідковувати напрямок погляду за допомогою безпечного інфрачервоного випромінювання, камер і алгоритмів наступним чином:

- 1) за допомогою променів ближнього інфрачервоного спектра створюються відблиски на поверхні очей;
- 2) камери роблять високошвидкісні знімки очей користувача і відблисків на їх поверхнях;
- 3) алгоритми розпізнавання образів розпізнає специфічні деталі очей користувача і шаблони відблисків;
- 4) ґрунтуючись на цих даних, математичні алгоритми вираховують точне положення очей, напрям погляду і точку, на яку ви дивитесь на екрані.

Технічні характеристики трохи змінились. Комплектація теж має незначні зміни:

- сам пристрій;
- подовжувач USB4;
- два магнітних плоских кріплення;

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

- одне гнучке кріплення;
- гарантія та інформація про запобіжні заходи;
- інструкція.

Приставка PCEye Mini (рис. 1.12) створена для керування поглядом ПК [13]. Вона також є доволі популярна. Mini PCEye дозволяє користувачеві:

- виконувати різні дії в інтернеті;
- виконувати різні дії в локальній мережі;
- займатися форматуванням документів;
- займатися редагуванням документів;
- грати в ігри тощо.

Mini PCEye встановлюється на ПК за допомогою магніту, підключається через USB. Орієнтована ціна – 2200\$.



Рисунок 1.12 – PCEye Mini

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

1.3 Постановка задачі

Наведені вище приклади існуючих рішень потребують як спеціального програмного забезпечення, так і складної технічної складової, що значно піднімає вартість продукту.

Варто також відзначити, що більшість таких продуктів створюються з метою урізноманітнити процес комп'ютерних ігор користувачами, а не допомогти людям з обмеженими можливостями.

Основною метою даного дипломного проєкту є створення програмного продукту, який зможуть використовувати усі зацікавлені.

В якості технічної складової програмний продукт використовуватиме наявні у користувача вбудовані в персональний комп'ютер засоби, такі як камера чи веб-камера.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Формалізація вимог до системи

Так як даний дипломний проєкт орієнтований на осіб з інвалідністю, потрібно добре продумати систему. Вона має бути простою і зрозумілою, враховувати всі потреби користувачів, легкою у встановлені.

Однією з головних вимог є простота системи. Користувачами програми можуть бути особи різного віку: від дітей до людей похилого віку. Тому варто виключити необхідність тривалого навчання.

Окрім простоти, система повинна використовувати наявні у користувача пристрої. Це може бути вбудована камера ноутбука чи персонального комп'ютера, або веб-камера. Якщо продавати камеру разом із програмою, це б призвело до підвищення ціни, тим самим зменшивши коло користувачів.

Також варто врахувати людські рефлексії, а саме – кліпання очима. Рефлекторно людина кліпає з невеликим інтервалом, приблизно кожні 2 – 10 секунд, тому потрібно виключити можливість виконання якоїсь дії під час цього.

Так як дана програма буде керувати курсором миші, варто забезпечити можливість вимкнення/ввімкнення даної функції. Постійний рух курсору викликає незручності та може призвести до небажаних дій.

Для забезпечення безпеки програмного коду та непередбачуваних поправок у програмному коді програму варто реалізовувати у форматі exe.

Рекомендовані мінімальні технічні вимоги:

- камера 5мп;
- Windows 7;
- процесор Intel Core i3;
- відеокарта RTX-1080;
- оперативна пам'ять 4 Гб;
- постійна пам'ять 500 Гб.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2 Проєктування структури системи

Система має виконувати такі функції: розпізнавання очей, фіксування зміни їх положення, керування курсором миші. Отже, дана система повинна містити такі модулі:

- модуль розпізнавання очей;
- модуль керування курсором.

Модуль розпізнавання очей є ядром системи, на якому базується роботи системи в цілому. Його завдання – визначити положення зіниці відносно всього ока. Також потрібно фіксувати їх попереднє положення для роботи наступного модуля.

Модуль керування курсором напряму залежить від модуля розпізнавання очей. Він має порівнювати два набори даних про положення зіниць. В результаті визначається, яку дію виконав користувач і тоді курсор миші виконує відповідну маніпуляцію. Це може бути рух курсора вгору, вниз, вліво або вправо по екрану, а також клік правою чи лівою кнопкою миші.

2.3 Проєктування алгоритму роботи системи

Алгоритм роботи програми відносно не є складним (рис. 2.1).

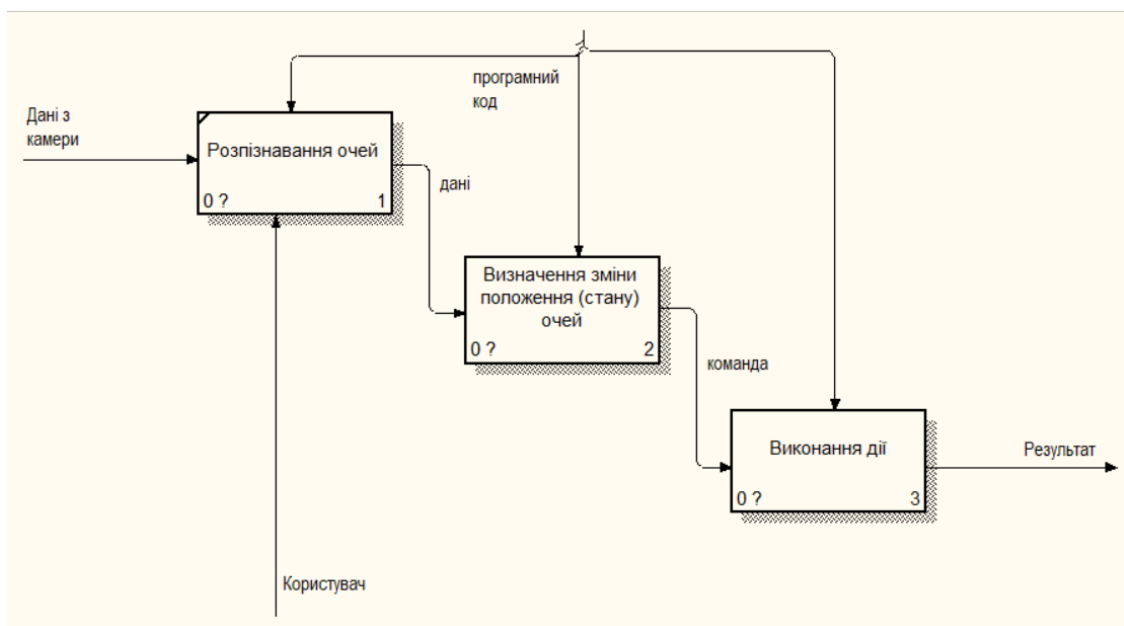


Рисунок 2.1 – Діаграма IDEF0 алгоритму роботи програми

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо коротко, то він включає наступні кроки:

- розпізнавання зіниць;
- визначення зміни положення зіниць;
- виконання певної дії (пересування курсора чи клік мишкою).

На рисунку 2.2 зображено діаграму IDEF3, яка представляє декомпозицію процесу визначення зміни положення (стану) очей.

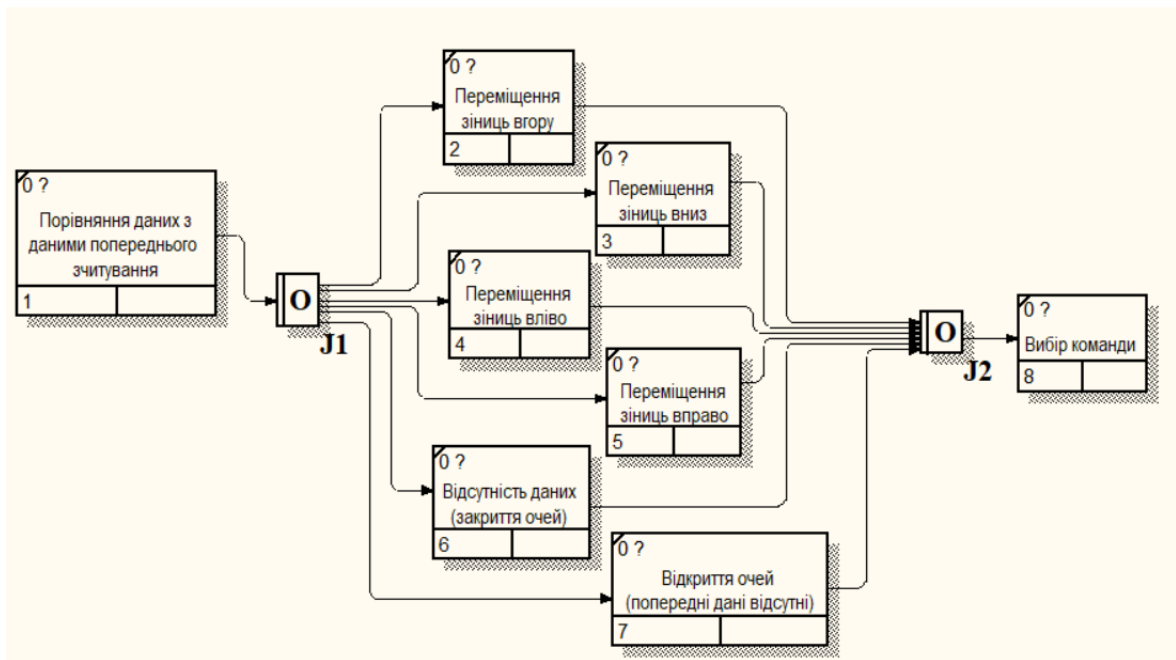


Рисунок 2.2 – Діаграма варіантів зміни стану очей

Щоб визначити зміну спершу потрібно порівняти теперішні дані з попередніми. Після цього можна робити висновки:

- зіниці перемістилися вгору;
- зіниці перемістилися вниз;
- зіниці перемістилися вліво;
- зіниці перемістилися вправо;
- очі закрились;
- очі відкрились.

Відповідно до дії користувача програма буде виконувати відповідну команду. В основному це буде переміщення курсора миші або клік (рис. 2.3).

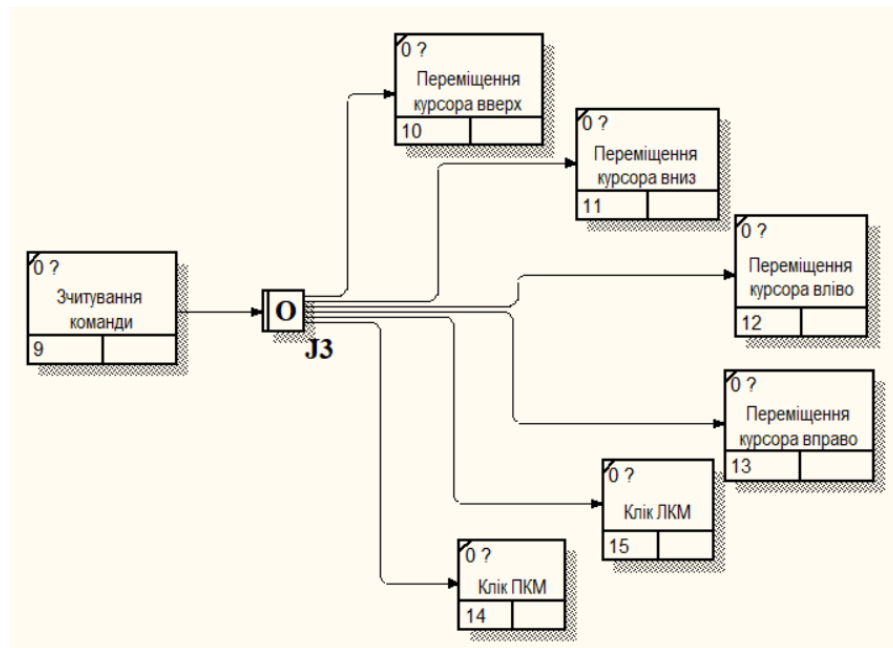


Рисунок 2.3 – Варіанти дій програми в IDEF3

Переміщення курсора доволі проста дія, а клік певною кнопкою миші може призвести до виконання ще інших дій. Це може бути:

- встановлення курсора у відповідне місце для написання тексту;
- відкриття програми чи файлу;
- виклик контекстного меню;
- прокрутка сторінки;
- друк тексту та інші.

При реалізації кліків мишки необхідно врахувати природне кліпання очима, що є звичним рефлексом людини. Так як людина кліпає з інтервалом від 2 до 10 секунд, потрібно запобігти виконанню будь-яких дій під час рефлекторного кліпання. Для цього для натискання кнопки миші користувач має тримати очі закритими деякий час. Для кліку лівою кнопкою миші це має бути короткий час, але довший від звичайного кліпання. Довше закриті очі призведуть до кліку правою кнопкою миші. Після виконання кліку можна відтворити звуковий сигнал, щоб сповістити про успішне виконання операції.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Вибір засобів реалізації

Перед початком програмної реалізації проекту потрібно вибрати редактор коду. Сьогодні є багато потужних і безкоштовних середовищ, наприклад, Atom, Sublime Text, Visual Studio Code, Notepad++ тощо.

Для даного проекту був вибраний Visual Studio Code. Це легкий, але потужний редактор коду, доступний для Windows, macOS та Linux.

Він має багату екосистему розширень для різних мов таких як C++, C#, Java, Python, PHP, Go та середовищ виконання (.NET та Unity).

При запуску редактора VS Code в перший раз відкривається початкова сторінка, на якій описані всі можливості редактора. Всі посилання інтерактивні і відкривають потрібну частину інтерфейсу або сторінку в браузері. Інтерфейс дуже зручний та інтуїтивно зрозумілий. Основну частину екрану займає вікно з текстовим вмістом файлу. У лівій частині екрана знаходяться вкладки активного меню, в якому знаходяться головні функції редактора. Вкладка провідника відкривається за замовчуванням при запуску програми. У ній виводяться список відкритих файлів і каталог відкритої папки.

Частиною активного меню є пошук. В полі можна ввести шукане значення, і редактор покаже, в яких місцях воно знаходиться. При бажанні можна поміняти значення через поле заміни. VS Code підтримує роботу з системою контролю версій Git. Можна виконати базові операції і подивитися, як змінилися файли з моменту останнього комміта.

VS Code відрізняється від конкурентів наявністю вбудованого відлагоджувача. Після невеликої настройки можна шукати баги в коді прямо з редактора – наприклад, поставити крапку зупинки і спостерігати за виконанням конкретної ділянки коду. Крім цього у редакторі є вбудована консоль, в яку може виводитися результат роботи або повідомлення про

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

помилку, якщо щось пішло не так. Відлагоджувач можна налаштувати під різні мови і різні завдання.

Останній пункт активного меню дозволяє встановлювати розширення. В VS Code є розширення для більшості мов програмування.

Наступним кроком є вибір мови програмування. Для реалізації проекту було обрано мову Python. Це сучасна мова програмування високого рівня. Її переваги – це висока продуктивність програмних рішень і структурований та легкий для розуміння програмний код. Синтаксис Python доволі простий, що дозволяє опанувати його за відносно коротким часом. Ядро має зручну структуру, а великий перелік вбудованих бібліотек дозволяє застосовувати широкий набір готових функцій.

Python може використовуватися для створення прикладних програм та розробки WEB-сервісів. До того ж, він активно розвивається: приблизно раз в 2 роки виходять оновлення. Головною особливістю Python є відсутність стандартів кодування (ANSI, ISO).

Python підтримує багато популярних операційних систем та може працювати на невеликих комп'ютерах і на великих серверах. Мова також підтримує роботу з віртуальною машиною Java.

У проекті буде використовуватись бібліотека OpenCV [14]. Це бібліотека Python для задач комп'ютерного зору та машинного навчання із відкритим вихідним кодом. Вона складається з понад 2500 алгоритмів, серед яких є як класичні, так і сучасні алгоритми для комп'ютерного бачення і машинного навчання. Основна увага приділяється обробці зображень, захопленню і аналізу відео з камер, включаючи такі функції, як виявлення осіб і виявлення об'єктів.

Computer Vision може бути визначене як дисципліна, яка пояснює, як реконструювати, переривати і розуміти тривимірну сцену з її 2D-зображень з точки зору властивостей структури, які присутні в сцені. Він займається

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

моделюванням і копіюванням людського зору з використанням комп'ютерного програмного та апаратного забезпечення [7].

Computer Vision має такі функції:

- обробка зображень – фокусується на маніпулюванні зображеннями;
- розпізнавання образів – пояснює різні методи класифікації зразків;
- фотограмметрія – отримання точних вимірювань із зображень.

Деякі з основних областей, де інтенсивно використовується Computer Vision:

- застосування в робототехніці (локалізація, навігація, уникнення перешкод, маніпуляції, Human Robot Interaction);
- застосування в медицині (класифікація і виявлення, 2D/3D сегментація, 3D реконструкція людського органа, візуальна робототехніка);
- застосування в промисловій автоматизації (промисловий контроль, складальний, зчитування штрих-коду і етикетки на упаковці, сортування об'єктів, розуміння документа);
- застосування в системах безпеки (біометрія, спостереження);
- застосування в автомобільних системах (автономний транспортний засіб, безпека).

Використовуючи бібліотеку OpenCV, можна [2]:

- читати зображення;
- писати зображення;
- захоплювати і зберігати відео;
- обробляти зображення (фільтрувати, перетворювати);
- виявляти певні об'єкти (такі як особи, очі, автомобілі в відео або на зображеннях) тощо.

OpenCV спочатку розроблявся на C++ [8]. На додаток до цього були надані прив'язки Python і Java. OpenCV працює у різноманітних операційних системах, таких як Windows, Linux, OSx, FreeBSD, Net BSD, Open BSD тощо.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Програмна реалізація

Спершу потрібно підключити усі необхідні бібліотеки. Це:

- OpenCV (cv2);
- Numpy;
- Pyautogui;
- Tensorflow;
- Os.path;
- Datetime.

Сама програма включає в себе 11 функцій. Також є цикл while, який весь час зчитує дані з камери та викликає всі необхідні функції. Нижче буде коротко описано кожну функцію.

Перша функція `get_face_detector` використовується для отримання моделі виявлення обличчя Caffe model з модуля DNN. Для цієї функції з пристрою підвантажуються два файли: `res10_300x300_ssd_iter_140000.caffemodel` та `deploy.prototxt`. Фрагмент коду в лістингу 3.1.

Лістинг 3.1 – Функція `get_face_detector`

```
if model_file == None:
    model_file = join(dirname(__file__), "res10_300x300_ssd_iter_140000.caffemodel")
if config_file == None:
    config_file = join(dirname(__file__), "deploy.prototxt")
model = cv2.dnn.readNetFromCaffe(config_file, model_file)
```

Наступна функція це `find_faces`. З назви зрозуміло, що вона шукає обличчя на отриманому зображенні. Використовує для цього модель, отриману з попередньої функції. Повертає координати виявленого обличчя. Програмний код в лістингу 3.2.

Лістинг 3.2 – Функція `find_faces`

```
hh, ww = img.shape[:2]
blob = cv2.dnn.blobFromImage(cv2.resize(images, (300, 300)), 1.0, (300, 300), (104.0, 177.0, 123.0))
model.setInput(blob)
res = model.forward()
faces = []
```

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження лістингу 3.2

```
for i in range(res.shape[2]):
    conf = res[0, 0, i, 2]
    if conf > 0.5:
        box = res[0, 0, i, 3:7] * np.array([ww, hh, ww, hh])
        (x, y, x1, y1) = box.astype("int")
        faces.append([x, y, x1, y1])
return faces
```

Наступна функція `get_landmark_model`. Вона обробляє `save model`, тобто будує модель ключових точок на обличчі. Для цього вона використовує бібліотеку `Tensorflow`. Програмний код в лістингу 3.3.

Лістинг 3.3 – Функція `get_landmark_model`

```
model = keras.models.load_model(saved_model)
return model
```

Функції `get_square_box`, `move_box` та `detect_marks` в результаті спільної роботи видають ключові точки на наявном зображенні. Тобто, визначають ключові точки в реальному часі. Фрагмент коду в лістингу 3.4:

Лістинг 3.4 – Функція `detect_marks`

```
faceImg = img[face_box[1]: face_box[3], face_box[0]: face_box[2]]
faceImg = cv2.resize(faceImg, (128, 128))
faceImg = cv2.cvtColor(faceImg, cv2.COLOR_BGR2RGB)
predictions = model.signatures["predict"](
    tf.constant([faceImg], dtype=tf.uint8))
marks = np.array(predictions['output']).flatten()[:136]
marks = np.reshape(marks, (-1, 2))
marks *= (face_box[2] - face_box[0])
marks[:, 0] += face_box[0]
marks[:, 1] += face_box[1]
marks = marks.astype(np.uint)
```

Для визначення очей на зображенні використовується функція `eye_on_mask`. Повертаються крайні точки очей. Це верхня, нижня, ліва та праві точки. Код в лістингу 3.5.

Лістинг 3.5 – Функція `eye_on_mask`

```
points = [shape[i] for i in side]
points = np.array(points, dtype=np.int32)
mask = cv2.fillConvexPoly(mask, points, 255)
l = points[0][0]
t = (points[1][1]+points[2][1])/2
r = points[3][0]
```

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження лістингу 3.5

```
b = (pointS[4][1]+pointS[5][1])/2  
return masK, [1, t, r, b]
```

Функція `find_eyeball_position` шукає позиції зіниць та повертає їх. Самі позиції кодуються цифрами (1, 2, 3, 0). Програмний код в лістингу 3.6:

Лістинг 3.6 – Функція `find_eyeball_position`

```
x_ratio = (end_points[0] - cx)/(cx - end_points[2])  
y_ratio = (cy - end_points[1])/(end_points[3] - cy)  
if x_ratio > 3:  
    return 1  
elif x_ratio < 0.33:  
    return 2  
elif y_ratio < 0.33:  
    return 3  
else:  
    return 0
```

Функція `contouring` обробляє вхідні дані та надсилає їх попередній функції `find_eyeball_position`. Також вона малює кола навколо зіниць користувача червоного кольору. Код функції в лістингу 3.7:

Лістинг 3.7 – Функція `contouring`

```
cntS, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)  
try:  
    cntT = max(cntS, key = cv2.contourArea)  
    m = cv2.moments(cntT)  
    x = int(m['m10']/m['m00'])  
    y = int(m['m01']/m['m00'])  
    if right:  
        x += mid  
    cv2.circle(img, (x, y), 4, (0, 0, 255), 2)  
    pos = find_eyeball_position(end_points, x, y)  
    return pos  
except:  
    pass
```

Функція `process_thresh` допомагає в обробці вхідного зображення. Наступна функція `mouse_func` є однією з ключових. Вона керує рухами курсора миші та виконує кліки. Для того, щоб рухався курсор миші потрібно подивитись у відповідну сторону. З кліканням все трохи інакше.

Для кліку правою кнопкою миші потрібно закрити очі на 3-4 секунди. Для кліку лівою кнопкою миші – 1-2 секунди. Також було реалізовано

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

можливість вимикати/вмикати функцію керування курсором миші. Для цього потрібно закрити очі на 5-7 секунд. Фрагмент коду для руху курсора в лістингу 3.8:

Лістинг 3.8 – Функція mouse_func

```
now2 = datetime.datetime.now()
time_now[0]=now2.hour
time_now[1]=now2.minute
time_now[2]=now2.second
if eye == 1 and move_mouse==1:
    pyautogui.moveRel(-20,0)
elif eye == 2 and move_mouse==1:
    pyautogui.moveRel(20,0)
elif eye ==
3 and move_mouse==1:
    pyautogui.moveRel(0,-20)
elif eye == 4 and move_mouse==1:
    pyautogui.moveRel(0,20)
```

3.3 Тестування готового продукту

Тестування будь якого програмного продукту починається з його запуску. Щоб запустити систему потрібно виконати файл eye_tracker.exe (рис. 3.1). Даний файл знаходиться в папці dist.

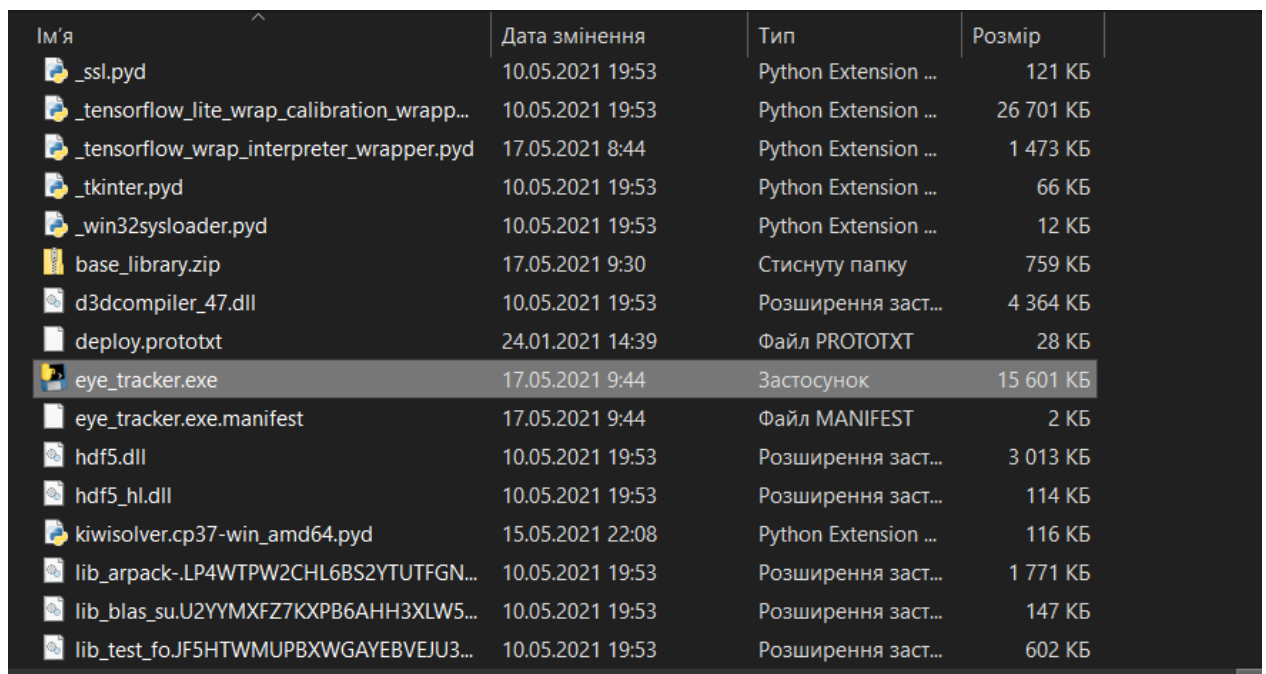


Рисунок 3.1 – Розташування файлу

Після подвійного кліку на файл майже відразу появляється вікно ініціалізації, в якому в подальшому будуть виводитись інформаційні сповіщення, зображений на рисунку 3.2.

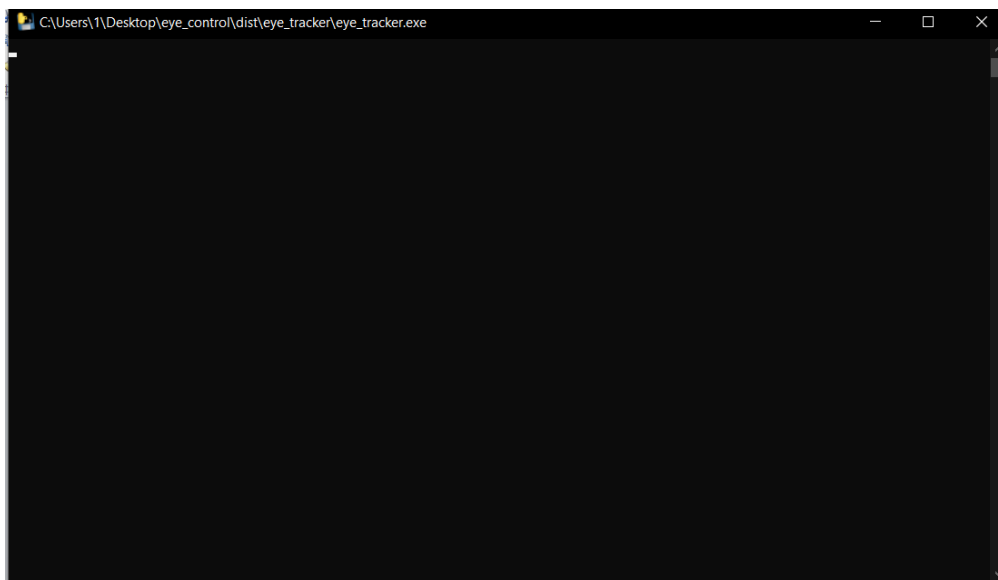


Рисунок 3.2 – Запуск програми

Якщо у вікні ініціалізації з'являється сповіщення про помилки (наприклад, не виявлено камеру), то їх необхідно усунути та перезапустити програму.

Якщо вікно ініціалізації виглядає як показано на рисунку 3.3, то програма запускається без помилок. Дані повідомлення означають запуск бібліотеки tensorflow.

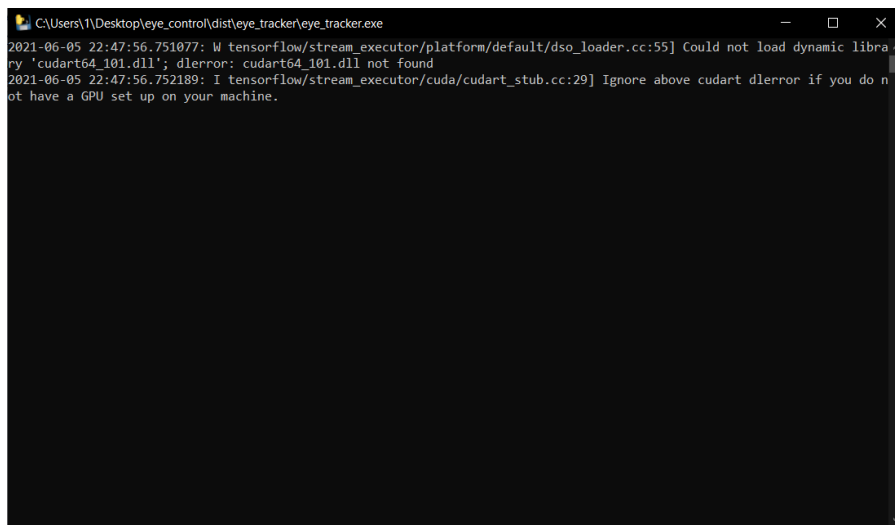


Рисунок 3.3 – Повідомлення при запуску програми

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

Запуск програми може тривати до 30 секунд, тому варто трохи зачекати. Обов'язково перед запуском потрібно перевірити підключення камери, якщо це не вбудована.

```

C:\Users\1\Desktop\eye_control\dist\eye_tracker\eye_tracker.exe
WARNING:tensorflow:Unable to create a python object for variable <tf.Variable 'layer1/conv2d/kernel:0' shape=(3, 3, 3, 3) dtype=float32_ref> because it is a reference variable. It may not be visible to training APIs. If this is a problem, consider rebuilding the SavedModel after running tf.compat.v1.enable_resource_variables().
WARNING:tensorflow:Unable to create a python object for variable <tf.Variable 'layer1/conv2d/bias:0' shape=(32,) dtype=float32_ref> because it is a reference variable. It may not be visible to training APIs. If this is a problem, consider rebuilding the SavedModel after running tf.compat.v1.enable_resource_variables().
WARNING:tensorflow:Unable to create a python object for variable <tf.Variable 'layer2/conv2d/kernel:0' shape=(3, 3, 32, 4) dtype=float32_ref> because it is a reference variable. It may not be visible to training APIs. If this is a problem, consider rebuilding the SavedModel after running tf.compat.v1.enable_resource_variables().
WARNING:tensorflow:Unable to create a python object for variable <tf.Variable 'layer2/conv2d/bias:0' shape=(64,) dtype=float32_ref> because it is a reference variable. It may not be visible to training APIs. If this is a problem, consider rebuilding the SavedModel after running tf.compat.v1.enable_resource_variables().
WARNING:tensorflow:Unable to create a python object for variable <tf.Variable 'global_step:0' shape=() dtype=int64_ref> because it is a reference variable. It may not be visible to training APIs. If this is a problem, consider rebuilding the SavedModel after running tf.compat.v1.enable_resource_variables().
WARNING:tensorflow:Unable to create a python object for variable <tf.Variable 'layer1/conv2d/kernel:0' shape=(3, 3, 3, 3) dtype=float32_ref> because it is a reference variable. It may not be visible to training APIs. If this is a problem, consider rebuilding the SavedModel after running tf.compat.v1.enable_resource_variables().
WARNING:tensorflow:Unable to create a python object for variable <tf.Variable 'layer1/conv2d/bias:0' shape=(32,) dtype=float32_ref> because it is a reference variable. It may not be visible to training APIs. If this is a problem, consider rebuilding the SavedModel after running tf.compat.v1.enable_resource_variables().
WARNING:tensorflow:Unable to create a python object for variable <tf.Variable 'layer2/conv2d/kernel:0' shape=(3, 3, 32, 4) dtype=float32_ref> because it is a reference variable. It may not be visible to training APIs. If this is a problem, consider rebuilding the SavedModel after running tf.compat.v1.enable_resource_variables().
WARNING:tensorflow:Unable to create a python object for variable <tf.Variable 'layer2/conv2d/bias:0' shape=(64,) dtype=float32_ref> because it is a reference variable. It may not be visible to training APIs. If this is a problem, consider rebuilding the SavedModel after running tf.compat.v1.enable_resource_variables().

```

Рисунок 3.4 – Інформаційні сповіщення при запуску програми

Після запуску на панелі з'явиться значок, зображений на рисунку 3.5. Тут видно, що запустилось два вікна.

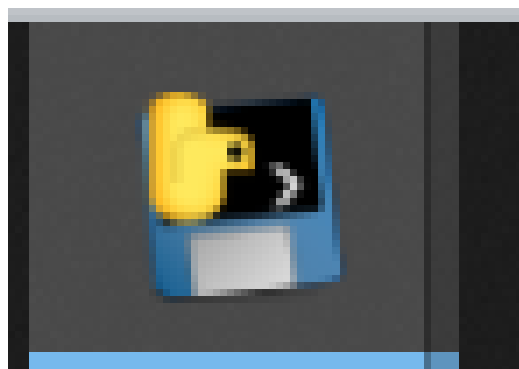


Рисунок 3.5 – Іконка програми

Після запуску програми відкривається ще одне вікно (рис. 3.6), яке достатньо просто згорнути. Причина його відкриття – необхідність визначення розташування очей на зображенні.

						<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			33

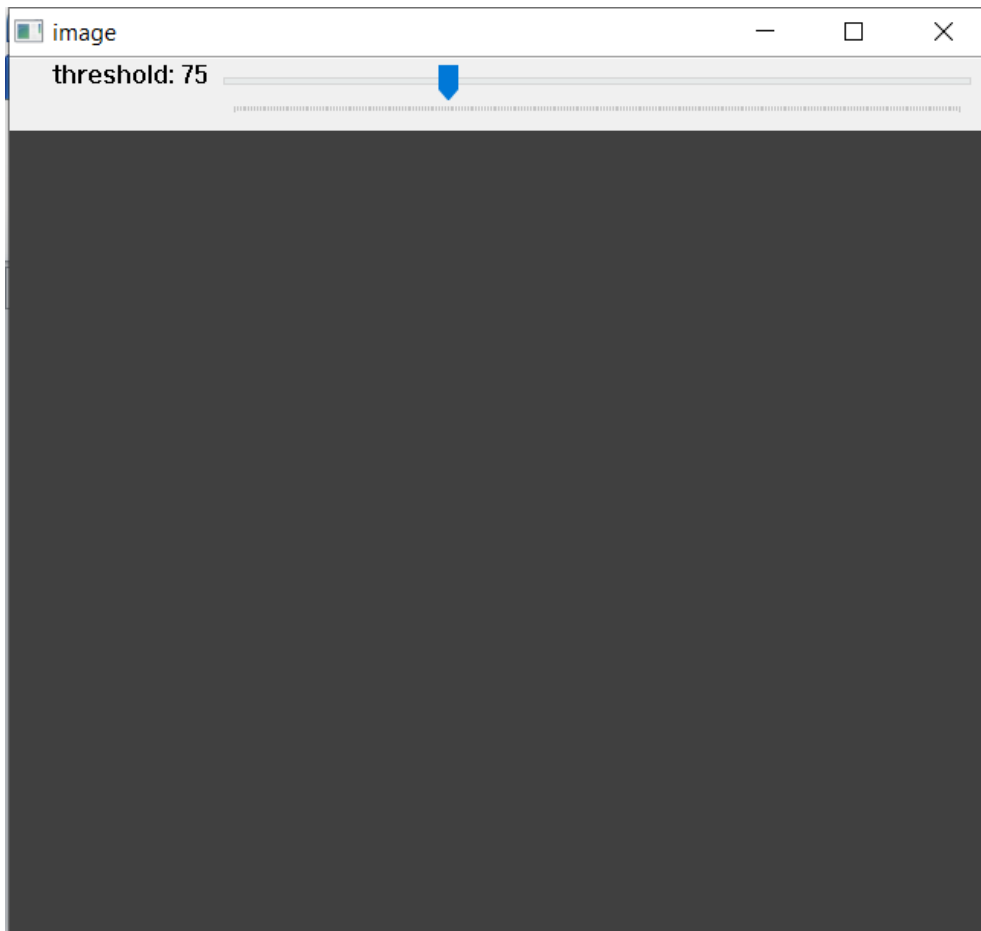


Рисунок 3.6 – Вікно визначення розташування очей

Керування курсором миші включається відразу після запуску програми.

На рисунку 3.7 зображено початкове положення курсора.

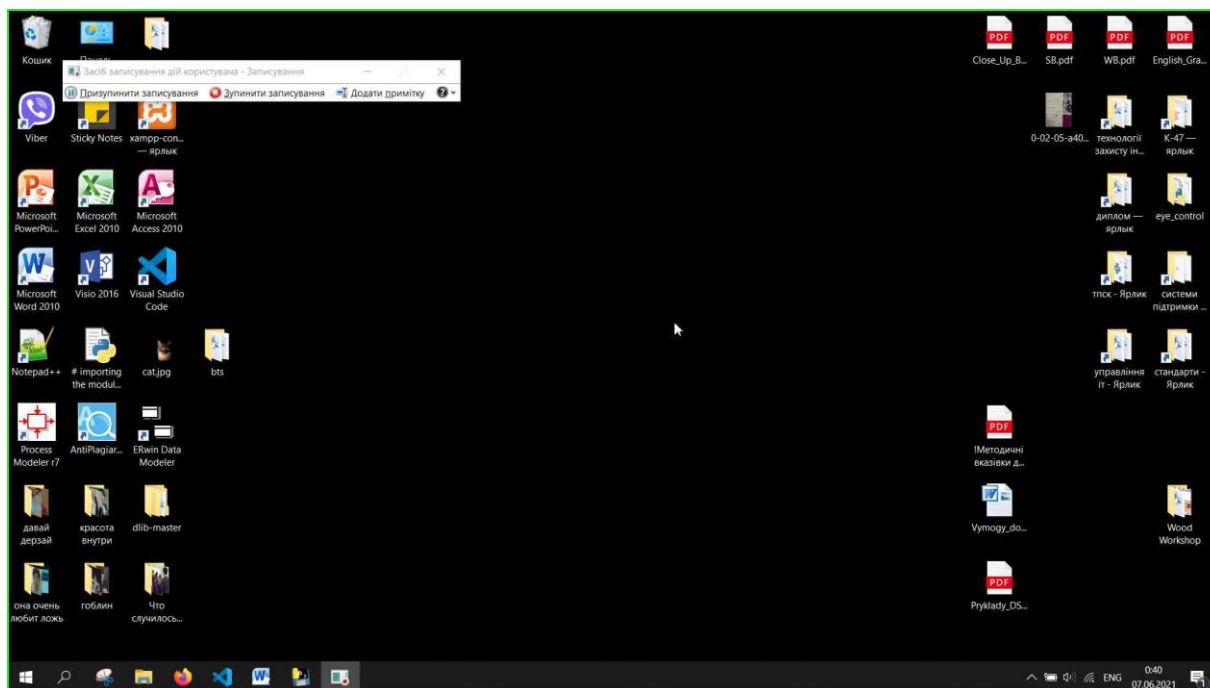


Рисунок 3.7 – Початкове положення курсору

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Якщо користувач деякий час буде дивитися вліво, курсор миші почне рухатись в цю сторону. Результат зображено на рисунку 3.8.

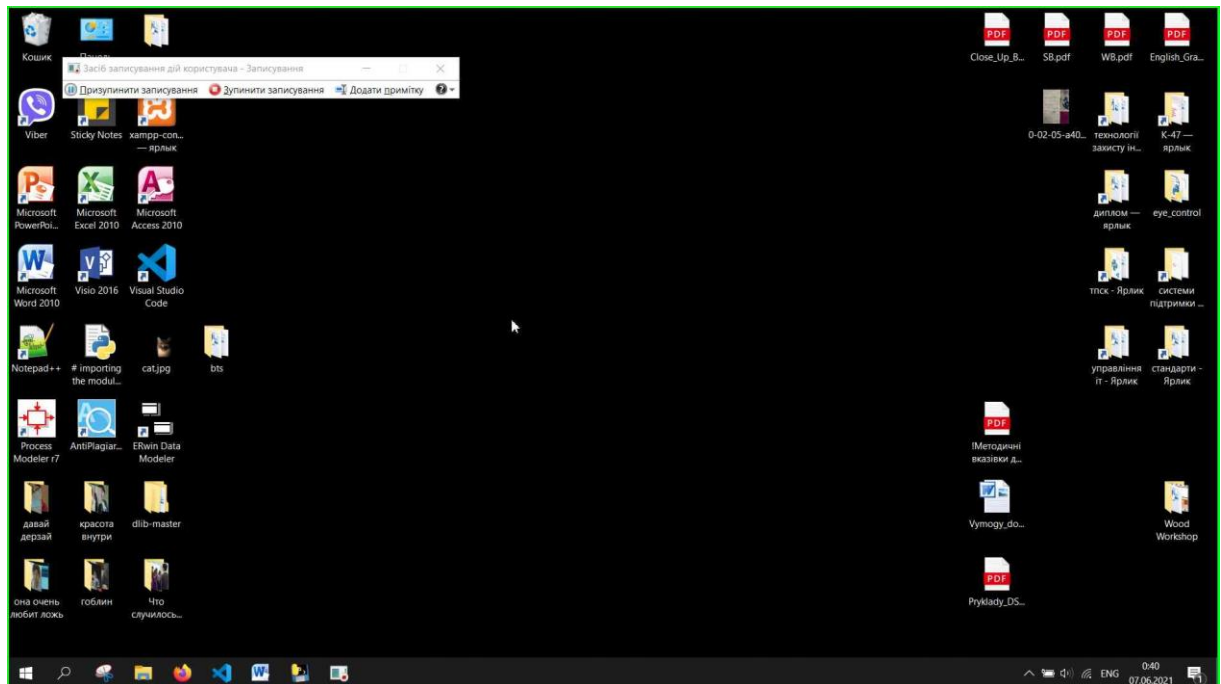


Рисунок 3.8 – Рух курсора вліво

Отже, курсор справно виконує запрограмований рух вліво. Далі, якщо користувач дивитиметься вправо деякий час, то курсор на екрані зміститься вправо (рис. 3.9).

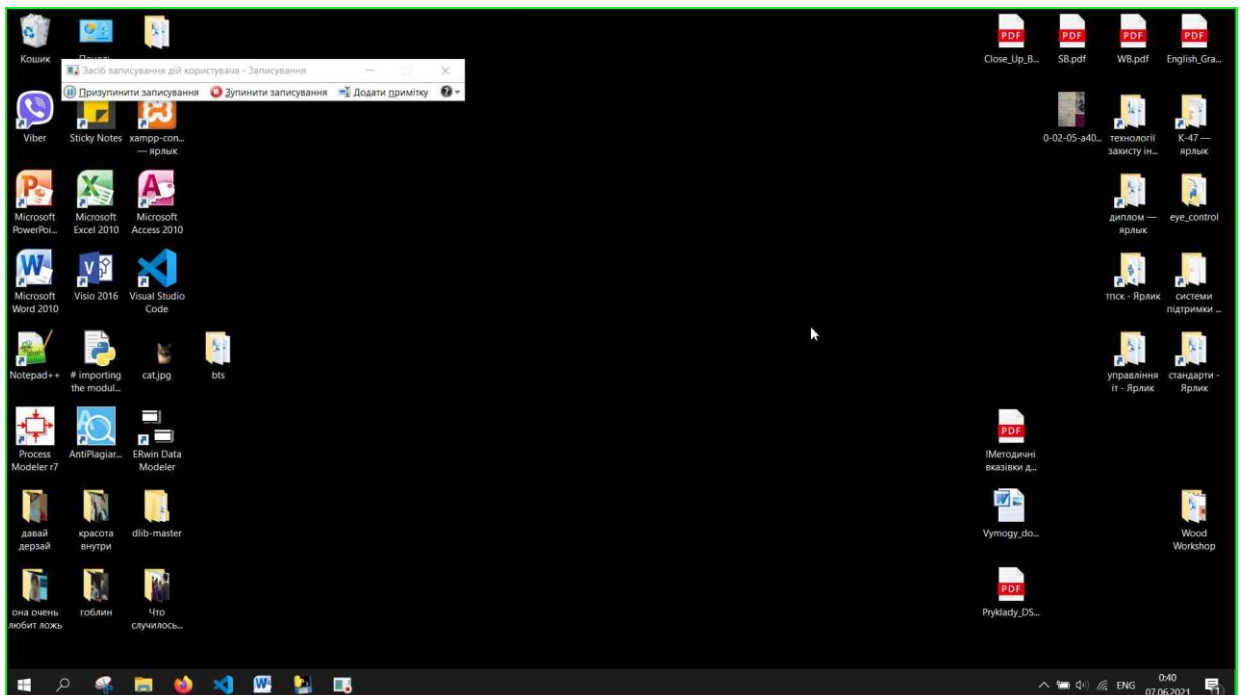


Рисунок 3.9 – Рух курсора вправо

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

Залишилися рухи вниз та вгору. На рисунку 3.10 зображено результат руху курсора миші коли користувач дивився вниз, а на рисунку 3.11 – коли користувач дивиться вгору.

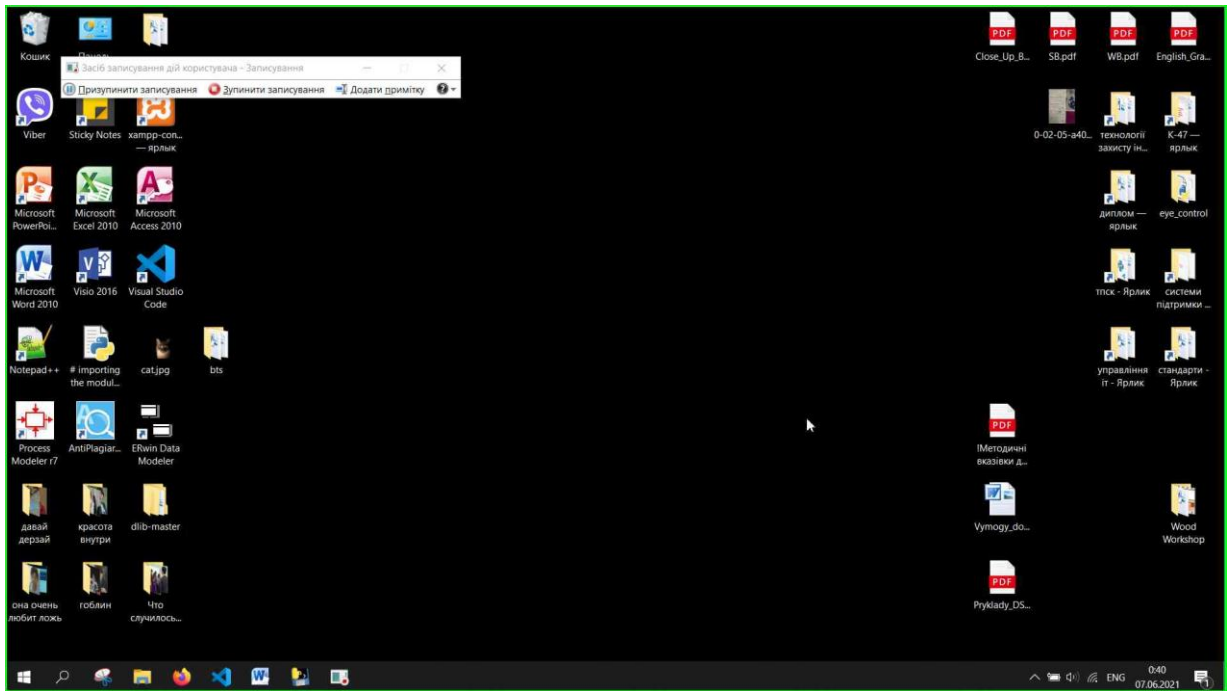


Рисунок 3.10 – Рух вниз

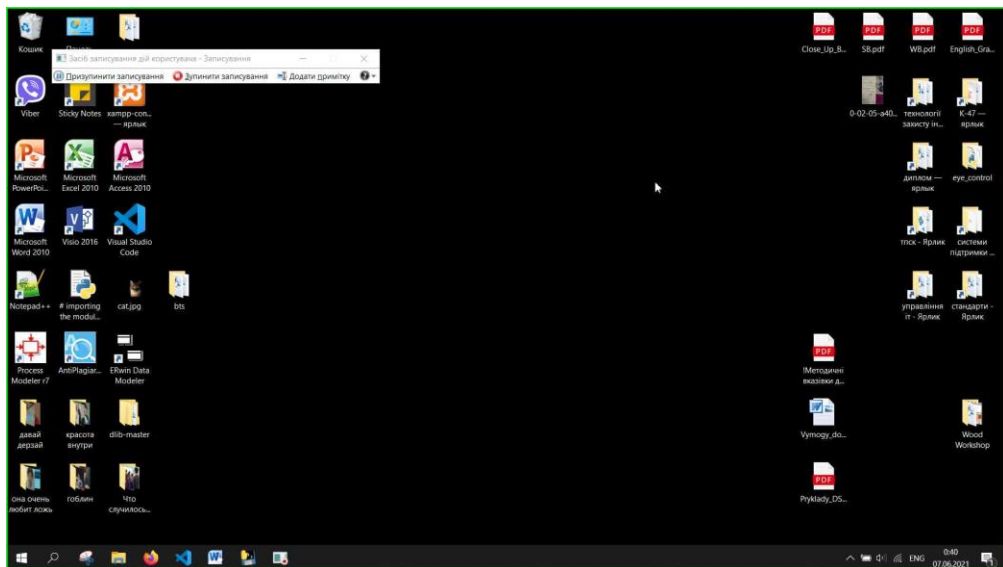


Рисунок 3.11 – Рух вгору

Наступним етапом буде тестування кліків мишею та включення/виключення рухів курсором. Щоб досконало використовувати дані функції потрібно трохи потренуватись.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Щоб протестувати клік лівою кнопкою миші потрібно вибрати одну програму на робочому столі. Користувач повинен закрити очі на 1-2 секунди. Результат зображено на рисунку 3.12.

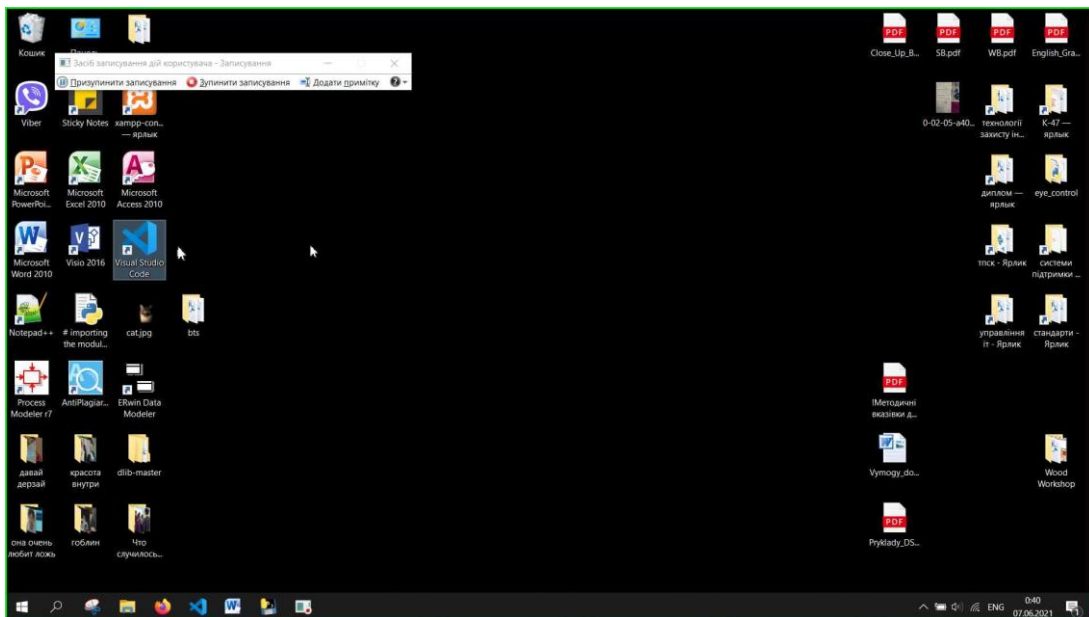
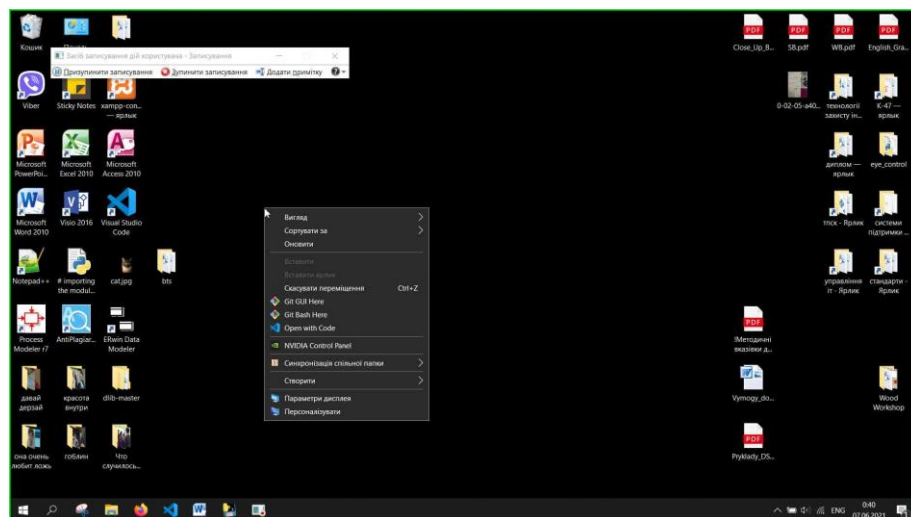


Рисунок 3.12 – Клік лівою кнопкою миші

Щоб протестувати клік правою кнопкою миші потрібно відкрити контекстне меню робочого стола або будь якої програми. Очі мають бути закриті 3-4 секунди. Результат зображено на рисунку 3.13.



Рисунку 3.13 – Виклик контекстного меню

Щоб виключити/включити рухи курсором миші очі потрібно тримати закритими 5-7 секунд. Дана функція також працює коректно. При вимкненні рухів курсором неможливо реалізувати кліки мишею.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

Якщо користувач буде тримати очі закритими довше зазначеного часу, то не буде нічого відбуватись. Якщо людина відійде від монітору на деякий час, то теж нічого не відбуватиметься, програма буде просто «очікувати» на координати зіниць ока.

Якщо потрібно буде виключити програму, то треба натиснути букву Q при включеній англійській розкладці клавіатури.

Варто також зауважити, що приміщення має бути добре освітлене. Бажано, щоб задній фон контрастував з кольорами на користувачеві. Також, не рекомендується працювати з програмою в окулярах. Усі ці фактори впливають на якість роботи програми. Вона може просто не коректно визначити положення обличчя, очей чи зіниць або ж взагалі не розпізнати їх.

Отже, тестування програми пройшло успішно. Виконуються усі заплановані функції без помилок. Щоб добре оволодіти програмою потрібно трохи потренуватись у виконанні простих дій з пристроєм, наприклад, відкриття файлів чи запуск програм.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

4.1 Аналіз ринку

Результатом дипломного проєкту є програма для керування ПК за допомогою рухів очей. Даний готовий продукт орієнтований, в основному на людей з обмеженими можливостями але його може використовувати будь-хто.

Сьогодні на ринку є такі продукти з програмною і технічною реалізацією. Дана програма є їх модифікацією, яка реалізована тільки програмно. Замість технічної складової продуктів на ринку ця програма використовує наявні у користувачів пристрої.

Найважливішою потребою замовників є оптимальна ціна і хороша якість продукту. Також він повинен бути простим у встановленні і користуванні. Потенційними замовниками є люди з обмеженими можливостями. А саме люди, які не мають можливості повноцінно використовувати свої верхні кінцівки. Також, можливий такий варіант, що даним продуктом можуть зацікавитися геймери. Даний продукт зможе урізноманітнити ігровий процес.

Для більшої доступності програмний продукт доцільно реалізовувати на просторах Інтернету. Також можна реалізовувати його в різних соціальних установах. Також, при високій потребі, можливий випуск нових, покращених версій продукту.

Головним конкурентом є пристрій Tobii Eye Tracker 5, який зараз є дуже популярним на ринку в Україні. Сам пристрій містить в собі інфрачервоні світлодіоди, камеру з високою роздільною здатністю і ASIC-чип для розрахунків. Також в комплекті є кріплення, інструкція, USB-подовжувач. Технічні характеристики:

- розміри 285×15×8,2мм;
- дистанція роботи 45-95см;
- розміри підтримуваних екранів 27" або 30";

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

- USB 2.0;
- вбудований кабель 0,8м, подовжувач 1м;
- 6 покоління IntelCore (i3/i5/i7) або 64-розрядний процесор AMD;
- операційна система Windows 8.1/7/10 64 bits;
- оперативна пам'ять 8 ГБ.

Пристрій Tobii Eye Tracker 5 коштує понад 8500 гривень. Його попередник Tobii Eye Tracker 4С буд дещо дешевий, до 7000 гривень. Це відносно дешеві варіанти. Дорожчою буде приставка PCEye Mini, ціна якої може досягнути 60000 гривень Також не менш дорогими будуть пристрої із закордонних магазинів, від 500\$ до 2 000\$.

4.2 Розрахунок витрат на проектування

В таблиці 4.1 зображено розрахунок витрат на розробку продукту.

Таблиця 4.1 – Кошторис витрат

Найменування статей витрат	Сума, грн	Обґрунтування
Зарплата проєктувальників	24922,8	
Відрахування на соціальні потреби	5483,02	
Контрагентські роботи і послуги	0	Немає потреби
Витрати на відрядження	0	Не передбачено
Інші прямі витрати	9969,12	
Усього прямих витрат	40374,94	
Накладні витрати	12112,48	
Планові накопичення	10497,48	
Усього, кошторисна вартість проєкту	62984,9	

Продовження таблиці 4.1

Податок на додану вартість	12596,98	
Загалом, договірна ціна розробки Зп	75581,88	

Зарплата проєктувальникам розраховується на двох осіб: науковий співробітник та інженер. У таблиці 4.2 наведено розрахунок їхньої заробітної плати.

Таблиця 4.2 – Розрахунок заробітної плати

N п/п	Посада виконавця	Оклад, грн/міс	Відрахування, грн/міс	Кількість		Сума з/п, грн.
				чол.	місяців	
1	Науковий співробітник	6461,4	1259,97	1	2	12922,8
2	Інженер	6000	1170	1	2	12000
Усього заробітної плати:						24922,8

Оклад працівника обчислюється так: ставка першого розряду * коефіцієнт посади. Ставка першого розряду станом на 1 січня 2021 року становить 2670 гривень. Науковий співробітник має тарифний коефіцієнт 2,42, тому його оклад становитиме $2670 * 2,42$, що дорівнює 6461,4 гривень. Тарифний коефіцієнт інженера 1,82, його оклад за місяць $2670 * 1,82$. Даний добуток дорівнює 4859,4 гривні і він менший за мінімальну зарплату. Отже, оклад інженера становитиме 6000 гривень, тобто мінімальна заробітна плата станом на 1 січня 2021 року.

Відрахування включають в себе податок на доходи фізичних осіб та військовий збір, які відповідно становлять 18% та 1,5% від окладу. Для наукового працівника сума відрахувань

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

становить $(6461,4*0,18)+(461,4*0,015)=1259,97$ гривень. Для інженера – $(6000*0,18)+(6000*0,015)=1170$ гривень.

Відрахування на соціальні потреби становить 22% від заробітної плати. Це виходить $24922,8*0,22=5483,02$ гривні.

У даному проекті немає потреби у контрагентських роботах і послугах та відрядженнях.

Інші прямі витрати становитимуть 40% від видатків на заробітну плату. Тобто $24922,8*0,4=9969,12$ гривень. Вони можуть включати вартість ліцензійного програмного забезпечення та літератури (за потреби).

Усього прямих витрат є 40374,94 гривні. Ця сума включає:

- зарплата проєктувальників (24922,8 гривень);
- відрахування на соціальні потреби (5483,02 гривні);
- контрагентські роботи і послуги;
- витрати на відрядження;
- інші прямі витрати (9969,12 гривень).

Накладні витрати включають витрати на електроенергію, опалення, оренду приміщення. Вони складають 30% від прямих витрат. Тобто $40374,94*0,3=12112,48$ гривень.

Планові накопичення даного проєкту становлять 20% від суми накладних і прямих витрат: $(40374,94+12112,48)*0,2=10497,48$ гривень.

Кошторисна вартість проєкту включає в себе вартість прямих і накладних витрат, планових накопичень. Це буде $40374,94+12112,48+10497,48=62984,9$ гривні.

Податок на додану вартість становить 20% від кошторисної вартості проєкту. Тобто $62984,9*0,2=12596,98$ гривень.

В результаті усіх цих обчислень загальна сума проєкту становить $62984,9+12596,98=75581,88$ гривню.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3 Обґрунтування необхідності розробки

Користуватись готовим продуктом даного проекту можуть усі бажаючі різної вікової категорії. Але основною метою створення було допомогти особам з обмеженими можливостями. Надати їм можливість повноцінно користуватись різними пристроями і не відставати від технічного прогресу людства. Не виключено, також, що даним продуктом зацікавляться ігромани. Він зможе урізноманітнити сам ігровий процес.

Люди з обмеженими можливостями зможуть повноцінно або майже повноцінно користуватись персональним комп'ютером. Тобто, при їхньому бажанні, влаштуватись на роботу. Це знизить рівень безробіття в країні, в казну буде приходити більше коштів від податків. Якщо ці люди зможуть працювати з ПК, тоді вони будуть мати можливість реалізовувати свої ідеї, тим самим піднести рівень економіки держави. Результатом такої діяльності може бути розвиток туристичного бізнесу, науково-технічний прогрес, підвищення прибутку підприємства, запобігти дистанціюванню таких людей від сучасного «цифрового» суспільства тощо.

Соціальний ефект може проявлятися у зменшенні безробіття, тобто створення нових робочих місць для людей з обмеженими можливостями. Це може запобігти відгородженню цих людей від суспільства.

Готовий продукт даного проекту може спричинити й екологічний ефект. Аналоги продукту використовують ще й технічні пристрої з пластику та інших матеріалів, які не підлягають переробці і мають довгий період розпаду. Отже, зменшиться шкідливий вплив на довкілля Землі.

Економічний ефект теж буде. Після збільшенн кількості робочих мість, збільшиться продуктивність підприємства. Це призведе і до збільшення прибутку.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В результаті виконання завдань дипломного проєкту було розроблено програму керування персональним комп'ютером за допомогою рухів очей. В процесі роботи над проєктом було проаналізовано наявні аналоги, визначено їх переваги та недоліки. На основі цих даних відбулося проєктування нової програми.

Після цього було сформульовано вимоги до системи, спроектовано структуру та алгоритм її роботи.

Після усіх цих етапів відбулася реалізація самого програмного продукту та його подальше тестування.

В економічній частині розраховано орієнтовані затрати на розробку проєкту та описано його доцільність. Поставлена мета даного проєкту досягнута, а основне завдання виконано.

Перевагою реалізованого програмного продукту є те, що він використовує вбудовану в персональний комп'ютер камеру. Це сильно відрізняє його від розглянутих аналогів, так як вони крім програмної складової передбачають ще й додаткову апаратну.

Для подальшого вдосконалення продукту можна використовувати більш точніші моделі розпізнавання зображень. Це збільшить точність розпізнавання та покращить роботу програми загалом. Також можна реалізувати простий графічний користувацький інтерфейс програмного продукту для швидкого виклику певних програм, наприклад, екранної клавіатури чи блокноту.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. PyAutoGUI. *PyPI*: вебсайт. URL: <https://pypi.org/project/PyAutoGUI/> (дата звернення: 21.01.2021);
2. Шпаргалка по OpenCV — Python. *Tproger*: вебсайт. URL: <https://tproger.ru/translations/opencv-python-guide/> (дата звернення: 17.12.2020);
3. OpenCV Course - Full Tutorial with Python. *Youtube*: веб сайт. URL: <https://www.youtube.com/watch?v=oXlwWbU8l2o> (дата звернення: 28.01.2021);
4. Урок #1 Python/Распознавание лиц OpenCV. *Youtube*: вебсайт. URL: <https://www.youtube.com/watch?v=i3MQZgS8BOM> (дата звернення: 15.01.2021);
5. Урок #2 Python/Распознавание глаз OpenCV. *Youtube*: вебсайт. URL: <https://www.youtube.com/watch?v=AmnWuw3z9jA&t=46s> (дата звернення: 19.01.2021);
6. [Python] Введение в OpenCV. Работа с камерой. *Youtube*: вебсайт. URL: <https://www.youtube.com/watch?v=1Yi3irfC8BA> (дата звернення: 19.02.2021);
7. OpenCV в Python. Часть 1. *Хабр*: вебсайт. URL: <https://habr.com/ru/post/519454/> (дата звернення: 02.02.2021);
8. OpenCV в Python. Часть 2. *Хабр*: вебсайт. URL: <https://habr.com/ru/post/528144/> (дата звернення: 04.02.2021);
9. OpenCV в Python. Часть 3. *Хабр*: вебсайт. URL: <https://habr.com/ru/post/539228/> (дата звернення: 04.02.2021);
10. OpenCV в Python. Часть 4. *Хабр*: вебсайт. URL: <https://habr.com/ru/post/549385/> (дата звернення: 08.02.2021);
11. Tobii Eye Tracker 4С. *Smart gadget club*: вебсайт. URL: <https://smart-gadget.club/fun/tobii-eye-tracker-4c> (дата звернення: 20.03.2021);
12. Tobii Eye Tracker 5 для игр. *Smart gadget club*: вебсайт. URL: <https://smart-gadget.club/fun/tobii-eye-tracker-5> (дата звернення: 25.03.2021);

					ДП. КН 21.437.16.000 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

13. Приставка PCEye Mini для управління поглядом на ПК. *Trosti*: вебсайт. URL: <http://www.trosti.com.ua/ua/pceye-mini.html> (дата звернення: 23.04.2021);

14. Касперська Л. А. Дослідження базових функцій для комп'ютерного бачення бібліотеки OpenCV. *Збірник наукових тез*: за матеріалами студентських наукових читань. Тернопіль: Навчально-практична майстерня редакційно-видавничих технологій Галицького коледжу імені В'ячеслава Чорновола, 2021р., С. 178-183.

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ
Додаток А
Програмний код

Лістинг А1 – Код функції get_face_detector

```
def get_face_detector(model_file=None, config_file=None, quantized=False):
    if quantized:
        if model_file == None:
            model_file = "opencv_face_detector_uint8.pb"
        if config_file == None:
            config_file = "opencv_face_detector.pbtxt"

        model = cv2.dnn.readNetFromTensorflow(model_file, config_file)
    else:
        if model_file == None:
            model_file = join(dirname(__file__), "res10_300x300_ssd_iter_140000.caffemodel")
        if config_file == None:
            config_file = join(dirname(__file__), "deploy.prototxt")

        model = cv2.dnn.readNetFromCaffe(config_file, model_file)
    return model
```

Лістинг А2 – Код функції find_faces

```
def find_faces(images, model):
    hh, ww = img.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(images, (300, 300)), 1.0, (300, 300), (104.0, 177.0, 123.0))
    model.setInput(blob)
    res = model.forward()
    faces = []

    for i in range(res.shape[2]):
        conf = res[0, 0, i, 2]
        if conf > 0.5:
            box = res[0, 0, i, 3:7] * np.array([ww, hh, ww, hh])
            (x, y, x1, y1) = box.astype("int")
            faces.append([x, y, x1, y1])

    return faces
```

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг А3 – Код функції get_landmark_model

```
def get_landmark_model(saved_model=join(dirname(__file__), 'pose_
model')):
    model = keras.models.load_model(saved_model)
    return model
```

Лістинг А4 – Код функції get_square_box

```
def get_square_box(box):
    left_x = box[0]
    top_y = box[1]
    right_x = box[2]
    bottom_y = box[3]
    box_width = right_x - left_x
    box_height = bottom_y - top_y
    diff = box_height - box_width
    delta = int(abs(diff) / 2)

    if diff == 0:
        return box
    elif diff > 0:
        left_x -= delta
        right_x += delta
        if diff % 2 == 1:
            right_x += 1
    else:
        top_y -= delta
        bottom_y += delta
        if diff % 2 == 1:
            bottom_y += 1

    assert ((right_x - left_x) == (bottom_y -
top_y)), 'Box is not square.'
    return [left_x, top_y, right_x, bottom_y]
```

Лістинг А5 – Код функції move_box

```
def move_box(box, offset):
    left_x = box[0] + offset[0]
    top_y = box[1] + offset[1]
    right_x = box[2] + offset[0]
    bottom_y = box[3] + offset[1]

    return [left_x, top_y, right_x, bottom_y]
```

Лістинг А6 – Код функції detect_marks

```
def detect_marks(img, model, face):
    offset_y = int(abs((face[3] - face[1]) * 0.1))
    box_moved = move_box(face, [0, offset_y])
    face_box = get_square_box(box_moved)
```

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження лістингу А6

```
h, w = img.shape[:2]

if face_box[0] < 0:
    face_box[0] = 0
if face_box[1] < 0:
    face_box[1] = 0
if face_box[2] > w:
    face_box[2] = w
if face_box[3] > h:
    face_box[3] = h

faceImg = img[face_box[1]: face_box[3], face_box[0]: face_box[
2]]
faceImg = cv2.resize(faceImg, (128, 128))
faceImg = cv2.cvtColor(faceImg, cv2.COLOR_BGR2RGB)
predictions = model.signatures["predict"](tf.constant([faceIm
g], dtype=tf.uint8))

marks = np.array(predictions['output']).flatten()[:136]
marks = np.reshape(marks, (-1, 2))
marks *= (face_box[2] - face_box[0])
marks[:, 0] += face_box[0]
marks[:, 1] += face_box[1]
marks = marks.astype(np.uint)

return marks
```

Лістинг А7 – Код функції eye_on_mask

```
def eye_on_mask(mask, side, shape):
    points = [shape[i] for i in side]
    points = np.array(points, dtype=np.int32)
    mask = cv2.fillConvexPoly(mask, points, 255)

    l = points[0][0]
    t = (points[1][1]+points[2][1])//2
    r = points[3][0]
    b = (points[4][1]+points[5][1])//2

    return mask, [l, t, r, b]
```

Лістинг А8 – Код функції find_eyeball_position

```
def find_eyeball_position(end_points, cx, cy):
    x_ratio = (end_points[0] - cx)/(cx - end_points[2])
    y_ratio = (cy - end_points[1])/(end_points[3] - cy)

    if x_ratio > 3:
        return 1
```

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження лістингу А8

```
elif x_ratio < 0.33:
    return 2
elif y_ratio < 0.33:
    return 3
else:
    return 0
```

Лістинг А9 – Код функції `contouring`

```
def contouring(thresh, mid, img, end_points, right=False):
    cnts, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)

    try:
        cntT = max(cnts, key = cv2.contourArea)
        m = cv2.moments(cntT)
        x = int(m['m10']/m['m00'])
        y = int(m['m01']/m['m00'])

        if right:
            x += mid

        cv2.circle(img, (x, y), 4, (0, 0, 255), 2)
        pos = find_eyeball_position(end_points, x, y)
        return pos
    except:
        pass
```

Лістинг А10 – Код функції `process_thresh`

```
def process_thresh(thresh):
    thresh = cv2.erode(thresh, None, iterations=2)
    thresh = cv2.dilate(thresh, None, iterations=4)
    thresh = cv2.medianBlur(thresh, 3)
    thresh = cv2.bitwise_not(thresh)

    return thresh
```

Лістинг А11 – Код функції `mouse_func`

```
def mouse_func(eye, move_mouse, time_now):
    if eye != 0:
        if time_now[3] == 0:
            now1=datetime.datetime.now()
            blink_time=now1-
datetime.timedelta(hours=time_now[0], minutes=time_now[1], seconds=time_now[2])
```

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження лістингу A11

```
        if blink_time.second>=1 and blink_time.second<=2 and
move_mouse==1:
            pyautogui.click(button='left')
        elif blink_time.second>=3 and blink_time.second<=4 an
d move_mouse==1:
            pyautogui.click(button='right')
        elif blink_time.second>=5 and blink_time.second<=7:
            if move_mouse == 1:
                move_mouse = 0
            elif move_mouse == 0:
                move_mouse = 1

        now2 = datetime.datetime.now()
        time_now=[now2.hour, now2.minute, now2.second, 1]
    elif time_now[3] == 1:
        now2 = datetime.datetime.now()
        time_now[0]=now2.hour
        time_now[1]=now2.minute
        time_now[2]=now2.second

        if eye == 1 and move_mouse==1:
            pyautogui.moveRel(-20,0)
        elif eye == 2 and move_mouse==1:
            pyautogui.moveRel(20,0)
        elif eye == 3 and move_mouse==1:
            pyautogui.moveRel(0,-20)
        elif eye == 4 and move_mouse==1:
            pyautogui.moveRel(0,20)
    elif eye == 0:
        time_now[3]=0

    e=[move_mouse, time_now]
    return e
```

Лістинг A12 – Основний код

```
face_model = get_face_detector()
landmark_model = get_landmark_model()

left = [36, 37, 38, 39, 40, 41]
right = [42, 43, 44, 45, 46, 47]

cap = cv2.VideoCapture(0)
ret, img = cap.read()

thresh = img.copy()
cv2.namedWindow('image')

kernel = np.ones((9, 9), np.uint8)
cv2.createTrackbar('threshold', 'image', 75, 255, nothing)
```

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження лістингу А12

```
now = datetime.datetime.now()
time_now=[now.hour, now.minute, now.second, 1]
move_mouse = 1

while(True):
    ret, img = cap.read()
    rects = find_faces(img, face_model)

    for rect in rects:
        shape = detect_marks(img, landmark_model, rect)
        mask = np.zeros(img.shape[:2], dtype=np.uint8)
        mask, end_points_left = eye_on_mask(mask, left, shape)
        mask, end_points_right = eye_on_mask(mask, right, shape)
        mask = cv2.dilate(mask, kernel, 5)
        eyes = cv2.bitwise_and(img, img, mask=mask)
        mask = (eyes == [0, 0, 0]).all(axis=2)
        eyes[mask] = [255, 255, 255]
        mid = int((shape[42][0] + shape[39][0]) // 2)
        eyes_gray = cv2.cvtColor(eyes, cv2.COLOR_BGR2GRAY)
        threshold = cv2.getTrackbarPos('threshold', 'image')
        _, thresh = cv2.threshold(eyes_gray, threshold, 255, cv2.
THRESH_BINARY)
        thresh = process_thresh(thresh)

        eyeball_pos_left = contouring(thresh[:, 0:mid], mid, img,
end_points_left)
        eyeball_pos_right = contouring(thresh[:, mid:], mid, img,
end_points_right, True)
        a=print_eye_pos(img, eyeball_pos_left, eyeball_pos_right,
move_mouse, time_now)
        move_mouse=a[0]
        time_now=a[1]

        for (x, y) in shape[36:48]:
            cv2.circle(img, (x, y), 2, (255, 0, 0), -1)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

Лістинг А13 – Код функції print_eye_pos

```
def print_eye_pos(img, left, right, move_mouse, time_now):
    if left == right:
        text = ''
        if left == 1:
            print('Looking left')
            move=mouse_func(1, move_mouse, time_now)
            text="Looking left"
```

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження лістингу А13

```
elif left == 2:
    print('Looking right')
    move=mouse_func(2, move_mouse, time_now)
    text="Looking right"
elif left == 3:
    print('Looking up')
    move=mouse_func(3, move_mouse, time_now)
    text="Looking up"
elif left == 0:
    print('Looking down')
    move=mouse_func(4, move_mouse, time_now)
    text="Looking down"
else:
    print("zero")
    move=mouse_func(0, move_mouse, time_now)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img, text, (30, 30), font,1, (0, 0, 255), 2,
cv2.LINE_AA)
else:
    move=[move_mouse, time_now]

return move
```

					<i>ДП. КН 21.437.16.000 ПЗ</i>	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		