

Virtual Mouse

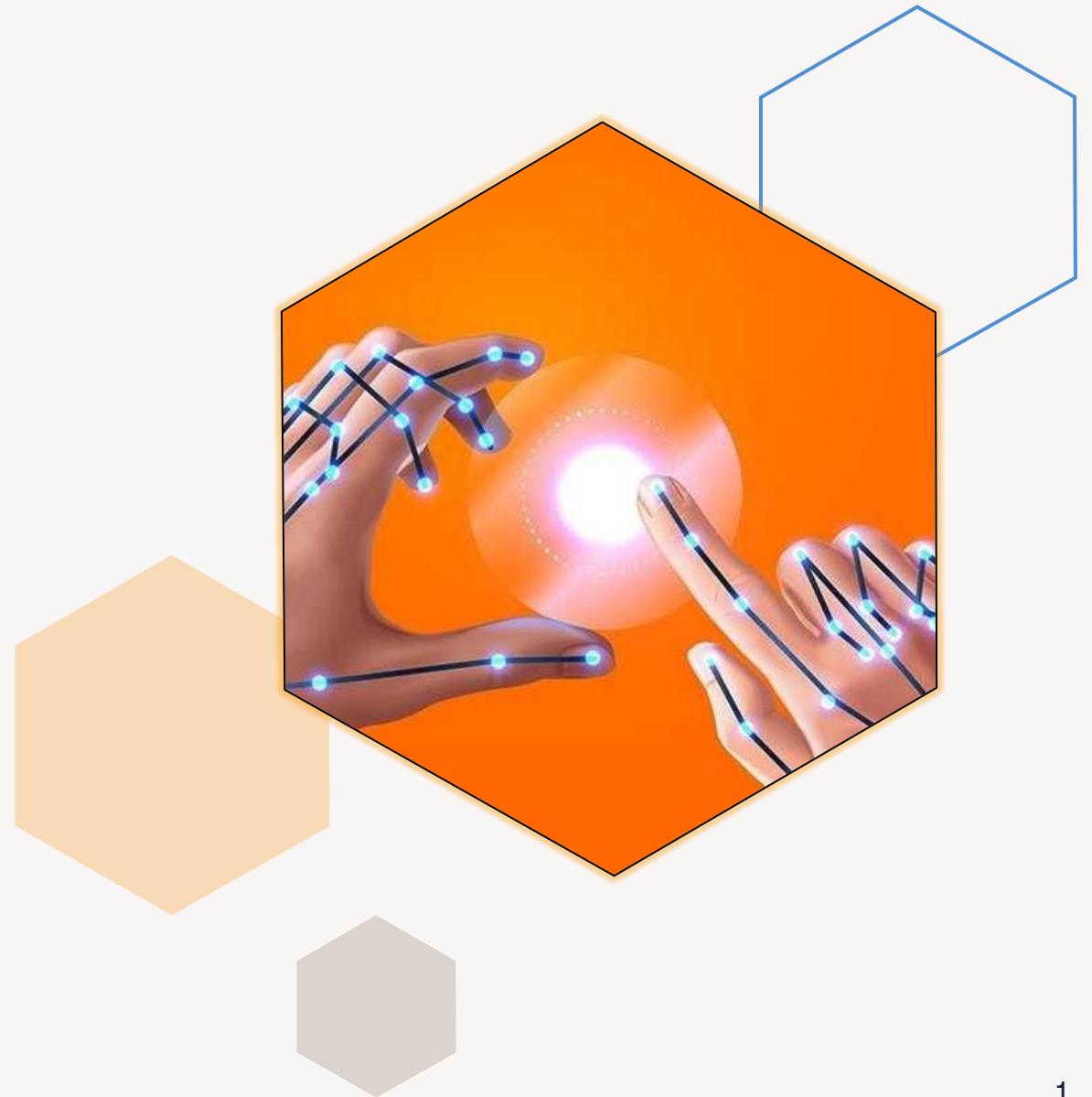
Controlled by Eye and Hand Gestures

Presented by: Group-12

Munna Biswas (221-15-5261)

Mehenaz Afsana (221-15-5693)

Afrin Jahan Moon (221-15-5499)



Objective

- To develop a virtual mouse that uses **eye** and **hand** gestures for operation.
- **Enhance accessibility** to supports users with **physical disabilities**
- Ensure the system operates smoothly and accurately in real-time.
- Contributing to the field of gesture-based computing.





Tobii Eye Tracker:

A high-precision eye tracking solution used in gaming and assistive technology.

Leap Motion:

A hand gesture control device commonly used for VR and other interactive applications.

EyeX:

Another eye tracking system used for accessibility and gaming.



Users and Stakeholders

- **Users:**
 - People with physical disabilities.
 - Gamers looking for an innovative control method.
 - General users seeking hands-free control.
- **Stakeholders:**
 - Developers.
 - Investors.
 - Accessibility advocates.

Name of Features



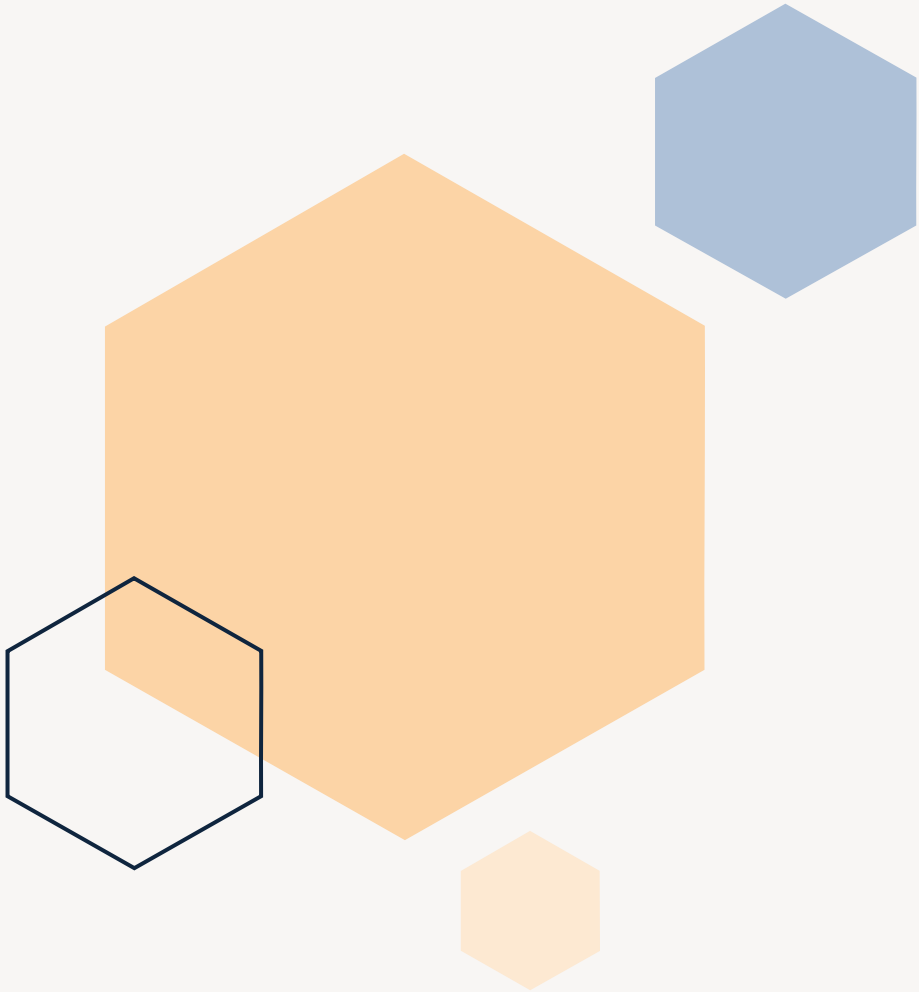
**Eye & hand gaze tracking
for cursor movement.**



**gesture recognition for clicks
and other actions.**

Customizable sensitivity





Implementation Requirements

Hardware:

- Webcam or eye tracker.

Software:

- Python
- Libraries for eye and hand tracking (OpenCV, MediaPipe, PyAutoGUI)

Application of OOP in the Project

Encapsulation

- Wrapping the tracking and recognition functionalities within classes.

```
6 1 usage  munnabiswas99 *
7 > class Eye(Mouse): ...
46
47
48 > class Hand(Mouse): ...
85
```

Application of OOP in the Project

Inheritance

- Creating base classes for common functionalities and deriving specialized classes for eye and hand tracking.

```
3 usages  munnabiswas99
4 class Mouse(ABC):
5     @abstractmethod
6     def run(self):
7         pass
```

```
6
7 > class Eye(Mouse):
46
47
1 usage  munnabiswas99 *
48 > class Hand(Mouse):
85
```


Application of OOP in the Project



Polymorphism

- Implementing method overriding for different types of gestures.

```
1 usage  munnabiswas99 *
7      class Eye(Mouse):
          1 usage  munnabiswas99 *
          def run(self):...
46
47
          1 usage  munnabiswas99
48      class Hand(Mouse):
          1 usage  munnabiswas99
          def run(self):...
85
```

Application of OOP in the Project

Abstraction

- Providing a simplified interface for gesture control without exposing the complex implementation details.

```
1 from abc import ABC, abstractmethod
2
3
4 class Mouse(ABC):
5     @abstractmethod
6     def run(self):
7         pass
```

```
1 from VirtualMouse import Mouse
2
3
4 class Eye(Mouse):
5     def run(self):
6         ...
7
8 class Hand(Mouse):
9     def run(self):
10        ...
```

Challenges and Solutions



Challenge: Accurate eye and hand tracking.

Solution: Implemented advanced algorithms and continuous calibration.

Challenge: Real-time performance.

Solution: Optimized code and utilized efficient libraries.

Challenge: User adaptability.

Solution: Provided customizable settings and thorough user testing.

Innovative Feature

Dual Control:

Simultaneous use of eye and hand gestures for enhanced precision and control.

Adaptive Learning:

System learns and adapts to user's behavior over time to improve accuracy.



Observations

- **User Feedback:**

Users find the system intuitive and useful, especially those with physical impairments.

- **Performance:**

Real-time tracking works efficiently on standard hardware.

- **Usability:**

Initial calibration is crucial for accurate performance.



A decorative graphic on the left side of the slide consists of a cluster of hexagons in various colors: light blue, orange, grey, and dark blue. Some hexagons contain images: a person in a meeting, two people working at a desk, a close-up of documents with charts, and a person looking at a whiteboard. Other hexagons are empty or have thin outlines.

Thank you