



Panini
School



SUBWAY SURFERS IN REAL LIFE

DOCUMENTATION

MID VALLEY SCIENCE & IT EXPO

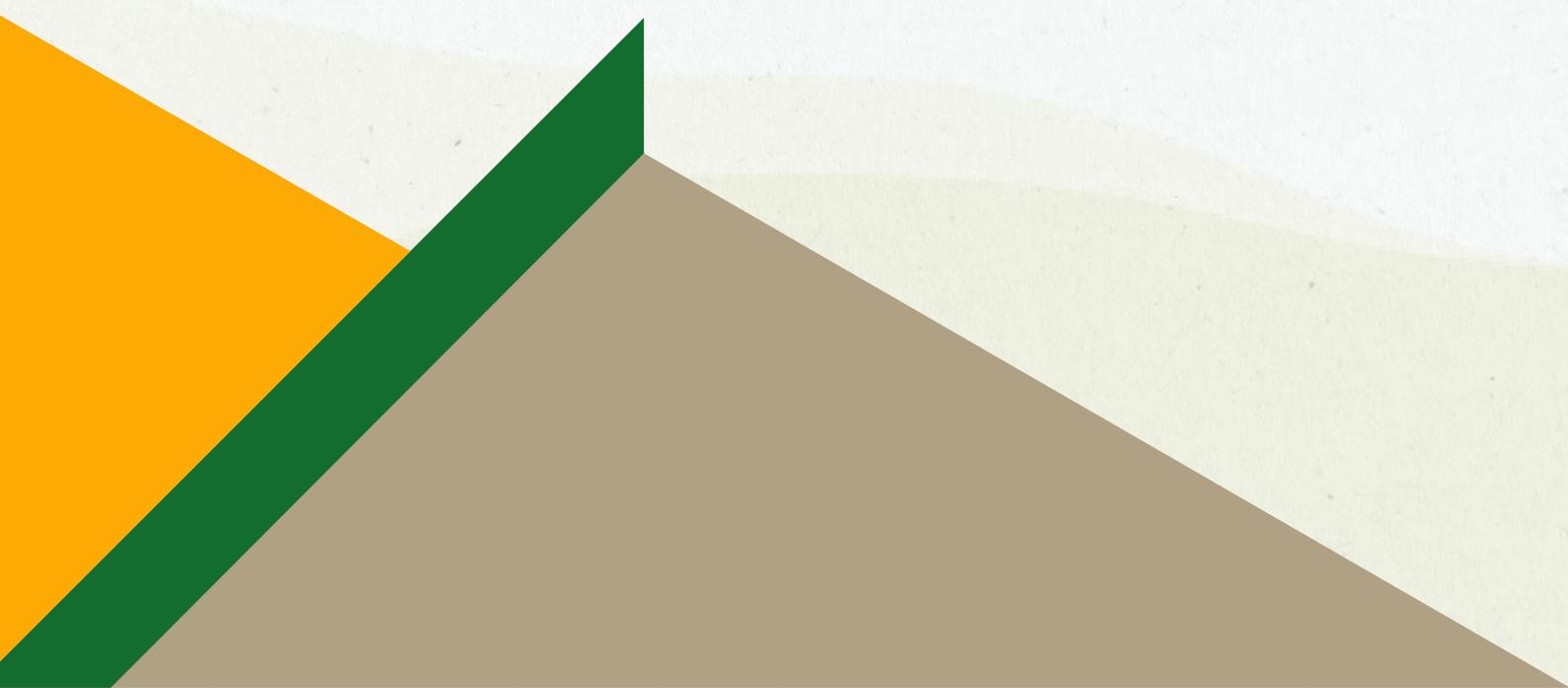
+01-4993290 

www.paninischool.edu.np 

Hi There!

We are a group of students from Panini, united by a shared passion for technology and innovation. What drives us is the thrill of building — not just for the sake of building, but to solve real problems and push what's possible. With every line of code and every experiment, we're learning, failing fast, and improving

We believe in the power of tech to make experiences more immersive and intuitive. We hope that by documenting our journey we can inspire others to think creatively about how software interacts with the human body.

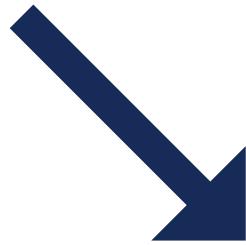


How It Works

Access the Camera

OpenCV to Access the Camera

- We use python library OpenCV to connect to the webcam.
- We create a VideoCapture object to grab live images from the camera.
- A loop runs that:
 - Reads each frame,
 - Shows it, and
 - Sends it for processing.



Get the Nose Position

MediaPipe to Track the Nose

- We use python framework MediaPipe which detects points (landmarks) on the face or body, like the nose and eyes.
- Each camera frame is processed to find landmarks.
- We select the nose tip coordinates.
- MediaPipe gives the nose position as normalized values, which we convert to the actual pixel positions on the screen.



Control the Game

PyAutoGUI to Press Keys

- We use python library PyAutoGUI to simulate keyboard presses based on the nose position.
- For example:
 - Nose moves left → press left arrow key
 - Nose moves right → press right arrow key

How we made it

Brainstorm Ideas

- We started by discussing different ways to make a fun and interactive project.
- Explored ideas involving games, computer vision, and real-time interaction.
- Decided on controlling Subway Surfers using nose movement for a unique experience.



Prepare the Essentials

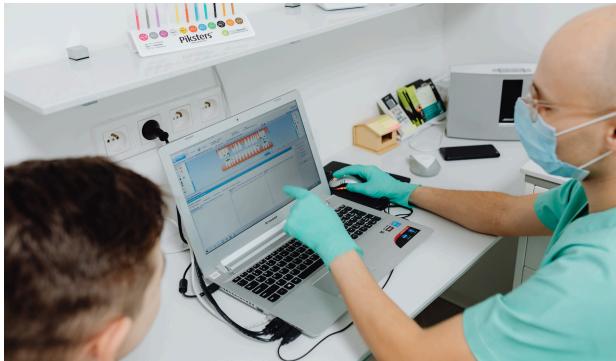
- Gathered the tools and libraries we needed: Python, OpenCV, MediaPipe, PyAutoGUI.
- Tested each library separately to understand how they work.
- Set up the webcam and created small experiments to track movement and simulate key presses.
- Finalize Project Design and Logic



Finalize Project Design and Logic

- Designed the overall workflow:
 - a. Capture camera frames with OpenCV.
 - b. Detect nose position using MediaPipe.
 - c. Convert nose movement into keyboard inputs with PyAutoGUI.
- Implemented, tested, and refined the logic to make the game respond smoothly and accurately.

Purpose

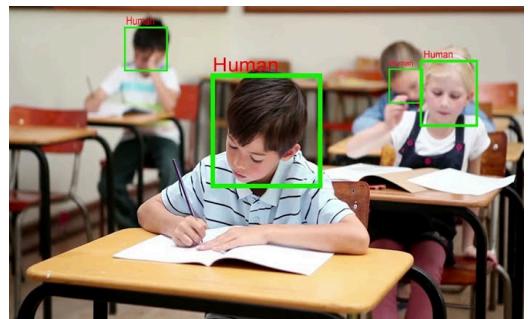


Medical and Accessibility

- Nose or head tracking can be used as an assistive technology for people who can't use hands, letting them control computers or wheelchairs.

Education and Learning

- Helps students understand computer vision and AI by building hands-on projects.

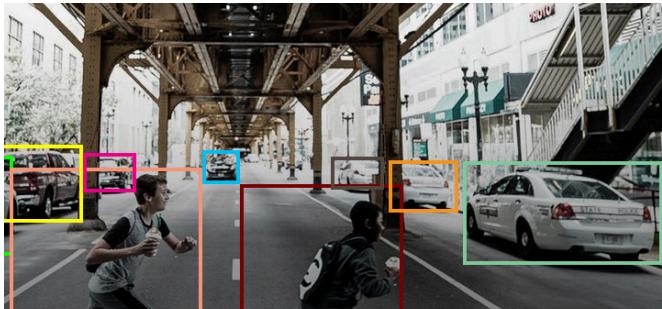


Military and Security

- Systems like this can be adapted for CCTV surveillance, tracking people's movements automatically.
- Can help monitor restricted areas or detect unusual behavior without manual observation.

Gaming and Entertainment

- Makes games more interactive and immersive, turning real-world movements into in-game actions.



Research and Robotics

- Can be used in human-robot interaction, teaching robots to respond to gestures or movement.

Meet Our Team



Aadim Gyawali

Team Leader

He's the strategic brain, the problem-solver and keeps the team aligned, efficient in a productive way and made the whole logic of the project

Designer & Artist

If something in this project looks clean, sharp, or actually pleasant to look at then its because of Riwaj. He brought the aesthetic clarity, visual direction and gave the project a colorful life.



Riwaj Karki



Smile Kalu

WEB DEVELOPER (HTML/CSS)

He brings structure, layout, and front-end polish to the table. He ensures everything looks professional and alignment & spacing are consistent.

Tools and Skills Used

Languages

- **Python** - Used to build the core game logic, track nose movement, and control the game in real time.
- **JavaScript** - Used to build the online scoreboard and handle the web-based features.

Frameworks

- **Next.js** - Used to build fast, scalable web applications for the scoreboard system.
- **React.js** - Used for creating the interactive web UI and rendering live scoreboard data.

Libraries

- **OpenCV** – Accesses the camera and reads each video frame.
- **MediaPipe** – Detects facial landmarks and gives the exact nose position.
- **PyAutoGUI** – Simulates keyboard presses (left, right, jump, etc.) to control the game.
- **Mongoose** – Used to structure, validate, and manage data in MongoDB for the scoreboard.

Database

- **MongoDB** – Stores player scores, game results, and scoreboard data reliably.

Backend Tools

- **Node.js** – Runs the backend for the scoreboard and handles requests.

Frontend Tools

- **HTML** – Structures the layout of the scoreboard website.
- **CSS** – Styles the interface to make it clean, readable, and visually appealing.

Tried the game?

Thank you!

Thank you for taking the time to look at our work. Your time, interest, and encouragement mean a lot to us. This project has taught us more than any classroom ever could, and we're proud to share the results with you.

I am Aadim guiding the vision and ensuring everything comes together smoothly. I have a strong interest in both web and app development, and I enjoy exploring innovative ways to combine technology with real-world interaction.

I have skills including computer vision, AI/ML, and full-stack development, Python, OpenCV, MediaPipe, PyAutoGUI, Next.js, React, and MongoDB.

I approach problem-solving in a hands-on and iterative way — constantly testing, optimizing, and refining to ensure the project is not only functional but also intuitive and user-friendly.

My projects:



Scan it!