**Body Action Game Implementer**

**Mini-project Report**

#### Submitted in partial fulfillment of the requirements For the of

**B.Tech.**

**Computer Science & Engineering (AIML)**

#### by

**Babita Sheoran (22AM1013) Mythri Patel (22AM1125)**

**Aman Singh (22AM1023) Nirav Kadam (22AM1075)**

#### Supervisor

**Dr. Shubhangi Ghate**

#### Department of Computer & Science Engineering (AIML) Ramrao Adik Institute of Technology,



#### Sector 7, Nerul , Navi Mumbai

#### (Under the ambit of D. Y. Patil Deemed to be University) November 11, 2024

**Ramrao Adik Institute of Technology**



(Under the ambit of D. Y. Patil Deemed to be University)

Dr. D. Y. Patil Vidyanagar,Sector 7, Nerul, Navi Mumbai 400 706.

**Certificate**

#### This is to certify that, the Mini-project titled

**“Body Action Game Implementer ”**

#### is a bonafide work done by

**Babita Sheoran (22AM1013) Mythri Patel (22AM1125) Aman Singh (22AM1023) Nirav Kadam (22AM1075)**

#### and is submitted in the partial fulfillment of the requirement for the degree of

**Bachelor of Technology Computer Science & Engineering (AIML)**

#### to the

**D. Y. Patil Deemed to be University**.

#### Dr. Shubhangi Ghate Dr. Sandeep Sangle Supervisor Project Coordinator

#### Dr. Sangita Chaudhari Dr. Mukesh Patil Head of Department Principal

**Dissertation Approval for B.Tech**

This is certify the dissertation entitled **“Body Action Game Implementer”** is a bonafide work done by **Babita Sheoran (22AM1013), Mythri Patel (22AM1125), Aman Singh (22AM1023), Nirav Kadam, (22AM1075)** under the supervision of **Dr. Shubhangi Ghate**. This dissertation has been approved for the award of Mini Project in Computer Science and Engineering, D. Y. Patil Deemed to be University.

##### Examiners:

1:

2:

##### Supervisor:

1:

Date :

Place :

**Declaration**

I declare that this written submission represents my ideas in my own word and where others ideas or words have been included, I have adequately cited and referenced the orig- inal source. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of above will be cause for disciplinary action by institute and can also evoke penal action from the source which have thus not been properly cited or from whom proper permission has not been taken when needed.

##### (Name of student and Roll No.)

Date :

**Acknowledgement**

We take this opportunity to express my profound gratitude and deep regards to my guide Dr. Shubhangi Ghate for her exemplary guidance, monitoring and constant encouragement throughout the completion of this report. We are truly grateful to her efforts to improve my understanding towards various concepts and technical skills required in our project. The blessing, help and guidance given by her time to time shall carry us a long way in the journey of life on which we are about to embark.

We take this privilege to express my sincere thanks to Dr. Mukesh D. Patil, Principal, RAIT, D. Y. Patil deemed to be University for providing the much necessary facilities. We are also thankful to Dr. Sangita Chaudhari, Head of Department of Computer Science and Engineering, Dr. Sandeep Sangle, Mini Project Coordinator, for their generous support. Last but not the least we would also like to thank all those who have directly or indirectly helped us in completion of this project report.

**Abstract**

This project presents a real-time computer game control system using graphical processing techniques with OpenCV. The goal is to allow players to interact with the game environment through gestures or movements captured by the camera, the system uses the capabilities of OpenCV for image acquisition, preprocessing, feature recognition and gesture recognition. It interfaces seamlessly with game logic and translates learned gestures into corresponding game commands.

The program focuses on achieving realtime feedback and accuracy, ensuring an immersive and intuitive gaming experience. System performance is evaluated through rigorous testing and validation, demonstrating its potential for improving communication interests and exploring new communication. It introduces a comprehensive implementer framework for developing body action games, aimed at enabling seamless player-environment interaction through full-body tracking, real-time response mechanics, and intuitive game- play integration models. Through practical case studies and performance evaluations, this framework demonstrates its potential to enhance engagement and interactivity in body action games, providing a foundation for future innovations in physical and virtual game experiences.

**Contents**

##### Abstract v

[List of Figures vii](#_TOC_250014)

[List of Tables viii](#_TOC_250013)

1. Introduction 1
   1. [literature review 1](#_TOC_250012)
   2. [Motivation 2](#_TOC_250011)
   3. [Objective 2](#_TOC_250010)
   4. [Scope for work 2](#_TOC_250009)
   5. [Organization of Report 3](#_TOC_250008)
2. Literature Survey 4
3. Methodology 6
   1. [Working of the Program 6](#_TOC_250007)
      1. [Starting the Program 6](#_TOC_250006)
      2. [Navigation of Program 7](#_TOC_250005)
   2. [Flowchart 8](#_TOC_250004)
   3. [Algorithm 9](#_TOC_250003)
   4. [Hardware Requirements 9](#_TOC_250002)
   5. [Software Requirements 10](#_TOC_250001)
4. Result Analysis  
   4.1 [Application………………………………………………………………………………………………………… 17](#_TOC_250000)
5. Conclusion  
   5.1 Future Scope ………………………………………………………………………………………………………… 18  
   5.2 Conclusion ………………………………………………………………………………………………………..… 19

6 References 21

# List of Figures

* 1. Gesture to move in-game character ”left” 13
  2. Gesture to move in-game character ”right” 13
  3. Left move 14
  4. Right move 15
  5. jump 16
  6. roll 16

# List of Tables

**Abbreviations**

FOC Fractional order Controller

FOPID Fractional order Proportional Derivative Controller ABC Fractional order Controller

**Chapter 1 Introduction**

Over the years, real-time computer games have become increasingly sophisticated, offering Player’s immersive experiences. One of the most important elements of a game is the Control mechanism, which directly affects the player’s interaction with the virtual Environment. Traditional embedded devices such as keyboards, mice, and game controllers are the primary means of controlling games, but advances in computer vi- sion and imaging, New control methods like gesture recognition and camera-based object tracking have been made possible by technological advancements. An effective open source library for image processing and computer vision applications is called Open CV (Open Source Computer Vision Library). Open CV gives programmers the ability to design applications that evaluate and change visual data over time, creating new possibilities for computer game control.

In this work, we explore how Open CV can be used to enable real time control of computer games using graphical techniques. By attaching the camera to the environment, players can interact with the game using gestures, movements, or objects detected by the camera. This approach provides an intuitive and Immersive gaming experience, breaking down the barriers created by traditional mechanical devices.

## literature review

Body action games use physical movements like gestures or full body actions to control gameplay, making the experience more interactive and immersive. These games are especially popular in education, fitness, and therapy because they engage both the mind

and body, promoting learning, teamwork, and well being.

Technologies like motion sensors, VR, and AR enhance these experiences by merging physical actions with digital environments. Players stay motivated through active participation, making such games both fun and beneficial for health, learning, or rehabil- itation.

## Motivation

The project is carried out with a different vision to improve the gaming experience for everyone. The first aims to improve accessibility by creating other controls for people with disabilities. Image processing can be used to track movements or faces, allowing control without the need for traditional game controllers.

This is based on awareness of the importance of inclusive game design. Secondly, the project aims to increase player involvement in the game by tracking movements using webcams and translating them into action in the game world. This promotes a better and more intuitive way to interact with the game, with the ability to focus on the line between player and character.

## Objective

* + - Make exercise fun by turning physical movements into exciting gameplay.
    - Boost learning and engagement through interactive games that involve body actions
    - Encourage teamwork by creating activities that need collaboration to succeed.
    - Help people stay fit by offering enjoyable ways to stay active.

## Scope for work

This project focuses on creating a game that encourages physical movement through fun, interactive activities. It aims to use body gestures or actions to control the game, making it engaging for players of all ages. The project can be applied in various fields, including

education, fitness, and rehabilitation, helping people learn, stay active, or recover in enjoyable ways.

## Organization of Report

The report details the various components of the software, including the coding languages used. It aims to demonstrate the various environment variables used for the software and the way in which the software works. It also showcases some of the screen snips along with the code snippet.

This report explains the front-end and back-end codes, database software used, environment variables, authentication software and API declaration. It also aims to highlight the applications of the software and the future scope, along with the additional implementations that can be done on it, along with references.

**Chapter 2 Literature Survey**

**R**ecognition of Hand Gestures We have studied the following ideas using Open CV and Python, which were published by V. Harini, V. Prahelika, Sneka, and P. Adlene Ebenezer: Hand gesture recognition is among the most practical and widely used methods for en- hancing human computer communication. Its use in gaming consoles like the Xbox and PS4 as well as other gadgets like laptops and smartphones has made it extremely popular in recent years. Applications for hand gesture recognition include accessibility assistance and medical fields. In this work, we would want to provide a method for creating a hand gesture detection simulation with Python 2.7 and OpenCV. The hand and background image are separated using a histogram-based technique. Techniques for background can- cellation are employed.

**M**ouse Control Using Computer Vision Faiz Khan, Basit Halim, and Asifur Rah- man’s book Object Detection and Marker Motion Tracking helped us understand the fundamentals of OpenCV. To find the items in the particular color range, we will thresh- old the image once it has been converted from HSV. Using certain conditions, this method transforms a grayscale image into a binary image.

**B**ased on Ibrahim Furkan Ince, Manuel Socarras-Garzon, and Tae Cheon Yang’s Analysis of Finger Blobs, we developed Hand Mouse: Real Time Hand Motion Detection System. We realized that, with the current single camera implementation, features like window scaling are feasible and would be accomplished by utilizing the difference in detected object size. Some kind of 3D hand detection is required for more intricate hand operations, like pointing to a wall to project a desktop, and we plan to use it in our upcoming 3D projects.

**S**wati Singhvi’s publication, Virtual Gaming Using Gesture Recognition Model, As human-computer interaction continues to advance and gesture-based apps gain popular- ity, Naman Gupta and Shashank Mouli Satapathy recognized the necessity of gesture recognition in the gaming business. In the past, computer interfaces were restricted to keyboard and mouse operations, where button clicks were required for input. As technol- ogy has advanced, vision-based human–computer interaction offers a wider spectrum of input by processing data from many cameras using computer vision. The term ”computer vision” refers to a branch of research that focuses on developing tools that let computers ”see” and understand the information contained in digital images, including pictures and movies.

**A**ccording to Sowmya Kini Ma, Rekha Bhandarkarband, and K Praveen Shenoy’s paper ”Real Time Moving Vehicle Congestion Detection and Tracking using Open CV,” the number of people using cars is rising daily in this day and age. It is getting increasingly difficult to organize, monitor, and control these vehicles. Since the system must be im- plemented without changing the infrastructure, video-based vehicle capture and analysis that doesn’t interfere with traffic is necessary to identify traffic accidents and congestion. Using video surveillance and traffic camera video data, we have developed a solution to the aforementioned issue in this research. We have employed tracking techniques such blob sampling, adaptive thresholding, and Gaussian-based background subtraction.

5

**Chapter 3 Methodology**

To bring body action games to life, this framework combines motion-sensing technol- ogy with adaptable game mechanics. First, it captures player movements using devices like cameras or wearable sensors, which track gestures and translate them into in-game actions. Then, AI algorithms process these movements in real time, adjusting game diffi- culty based on the player’s skill level and style. This approach makes the gameplay feel intuitive and responsive. The framework is modular, so developers can easily tweak or add features—like specific gestures or unique interaction feedback—without overhauling the entire setup. This methodology aims to make body action games more engaging and accessible by letting players control the game naturally, just by moving.

## Working of the Program

### Starting the Program

After starting the program by importing libraries in the main loop function camera captures the center part of the body, in Camera captures center part of the objects and start recognizing if the person/object moves extend this, after capturing the center body part with the camera, the program can delve deeper. Movement detection kicks in, ana- lyzing differences between frames to identify where and how the person is moving. This can be further refined by dividing the image into regions to pinpoint specific limb move- ment.

By interpreting these movements, the program translates them into in game actions. Raising your entire body might trigger a jump. To ensure accuracy, sensitivity thresholds can be set to distinguish intentional movements from camera shake. The possibilities extend beyond simple body tracking. Pose estimation can identify key body joints for precise control, and object recognition allows the program to react to specific items in the camera’s view. Machine learning can even be implemented for the program to learn and adapt to different player movements over time.

Finally, user feedback through visual cues and customization options ensures a smooth and personalized experience for all players. The game starts if the person makes the victory symbol, this is the starting position for our character to move. In this unique game, the character’s journey commences not with a typical button press, but with a triumphant gesture the victory sign. As the player raises their index and middle finger in a V shape

### Navigation of Program

This game takes immersion to a new level by utilizing hand gestures for character control. If the person moves right, the character in game will also move right, and vice versa. If the person jumps the character in game also jumps.

The game starts if the person makes the victory hand sign, this is the starting position for our character to move. In this unique game, the character’s journey commences not with a typical button press, but with a triumphant gesture the victory sign.

## Flowchart

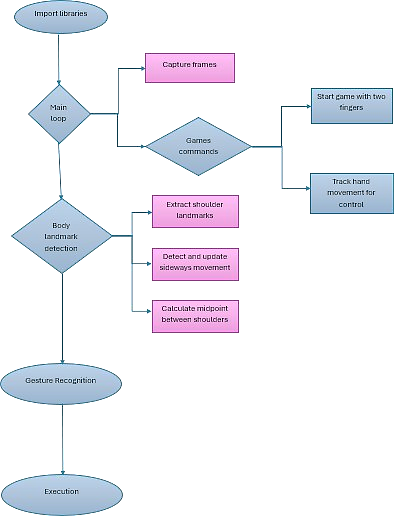


Figure. 3.2: Flowchart

## Algorithm

1. Imports and Setup.
   1. Import necessary libraries: cv2, mediapipe, pyautogui.
   2. Define tiplds for fingertip landmarks.
   3. Initialize variables for game state, character position, and hand tracking.
2. Main Loop
   1. Continuously capture frames from the camera.
   2. Process each frame for hand and body landmarks using Mediapipe.
   3. Convert the frame to RGB for processing.
3. Body Landmark Detection
   1. Extract shoulder landmarks to determine character movement.
   2. Calculate mid-point between shoulders for center reference.
   3. Detect sideways movement (left and right) and update character position accordingly.
4. Hand Landmark Detection
   1. Detect hand landmarks for both hands.
   2. Determine the type of hand (left or right).
   3. Check if fingers are open or closed for both hands. 5. Game Control Commands.
   4. Start the game if both hands have two fingers open (thumb and index finger).
   5. Track hand movements for character control (up, down, left, right).
5. Display
   1. Display the processed frame with annotations (landmarks, lines, circles).
   2. Wait for key press event and update the frame accordingly. This breakdown provides a structured view of the code, making it easier to understand each part and its functionality.

## Hardware Requirements

Camera, The investment requires a web browser or any other camera capable of capturing video images. Higher resolution cameras with good frame rates are preferred for accurate detection of the gesture. Processing Unit: A computer or embedded system with enough processing power to run image processing algorithms in real time. For parallel processing, the CPU must have more cores, and a dedicated GPU can speed up some calculations.

## Software Requirements

Operating System: Windows, Linux, or macOS are examples of suitable operating systems that must be used by the system. The OpenCV Library Install Open CV, an open source library for computer vision image processing that offers a variety of tools and methods for gathering objects and images for examination. Programming Environment: Select the programming environment or IDE (Integrated Development Environment) you use to write the project. Typical products are Jupyter Notebook,PyCharm, or Python with Visual Studio for C++development.Optional Dependencies: Depending on the specific requirements of the project, additional software libraries or tools may be required. For example, machine learning libraries such as TensorFlow or scikit-learn can be used for gesture recognition, and GUI frameworks such as Tkinter or PyQt can be used to create a user interface. Game Engine or API (Optional): When adding the gesture control system to an existing game, make sure it is compatible with the game engine or API used for game development. Common game engines include Unity, virtual engines, or custom en- gines. Development tools: Installation of necessary development tools such as compilers, debugging systems and version control systems for efficient development and debugging. Documentation and references: Make documentation available, including Open CV documentation, tutorials, and descriptions of the algorithms and methods used in the project.

**OpenCV**: OpenCV is a robust computer vision toolkit that may be used to create body-action games, in which players manipulate the game with their gestures and mo- tions. Capture video from a webcam or camera using OpenCV’s VideoCapture method, which allows you to capture live feed data. This is crucial for real-time feedback, as the player’s body movements are captured frame by frame.

**MediaPipe**: MediaPipe provides robust models for pose detection, face tracking, hand tracking, and more. MediaPipe works seamlessly with OpenCV for capturing camera input and processing frames. Use OpenCV to capture video and pass each frame to MediaPipe for analysis.

**PYAutoGUI**: PyAutoGUI is integrated into a body-action game to control the mouse, keyboard, or interact with elements on the screen based on player movements detected by MediaPipe. This is especially useful for games where body movements need to translate into mouse clicks, drags, keyboard presses, or other on-screen actions.

**Chapter 4 Result Analysis**

The provided code implements a system for controlling a game, likely Subway Surfers, using Body Movement and Shoulder position through a webcam. Here’s a breakdown of its functionalities achieved:

1. **Game Start Gesture:** By extending two fingers on each hand (excluding the thumbs), the user initiates the game with a victory symbol as shown in Fig 4.1.

Fig 4.0 Start Gesture

1. **Sideways Movement (Left/Right):** The user’s movement within a designated area on the screen is tracked based on shoulder position. Moving a shoulder past the center triggers a left or right as shown in Fig 4.2 and 4.3 to control the character’s movement.

Above images show the working of the game. The images tell us where the position of the person should be in order to move the in-game character left and right (Fig 4.1 and

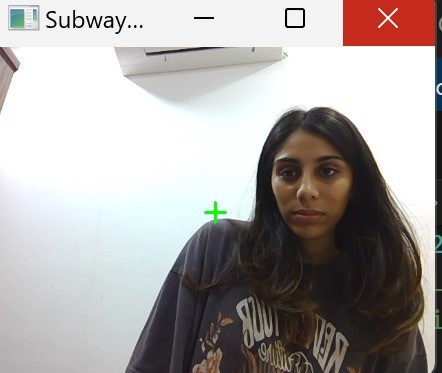


Figure 4.1: Gesture to move in-game character ”left”

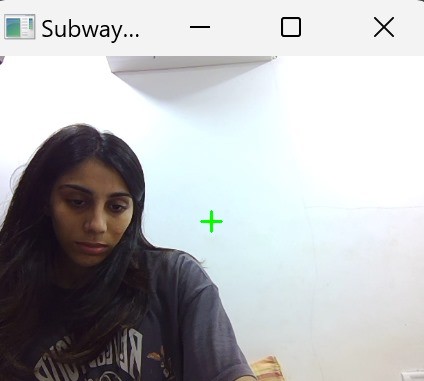


Figure 4.2: Gesture to move in-game character ”right”

Fig 4.2). For the character to jump, the person also has to jump in front of the webcam. All of it is captured using the webcam. Moving a shoulder past the center triggers a left or right arrow key press to control the character’s movement. The user’s movement within a designated area on the screen is tracked based on shoulder position.

1. **Up and Down Movement (Jump/Crouch):** The vertical distance between the user’s shoulders (compared to a center point established at game start) determines jump or crouch actions using up and down arrow key presses.

Building on the game’s unique control scheme. By recognizing the specific gesture, the code translates it into an in-game action, in this case, moving the character towards the left side of the screen as shown in Fig 4.3. This innovative approach creates a direct link



Figure 4.3: Left move

between the player’s physical actions and the character’s movements within the game world, fostering a more intuitive and engaging gamep lay experience.

Fig 4.3 showcases the character moving left, likely triggered by the gesture in Fig 4.1. There’s a possibility this is controlled by Algorithm, which might analyze camera detection of the player’s hand movement and translate it into leftward movement.

However, there’s also Fig 4.2, which suggests a different scenario. Here, the character moves right.

Building upon the innovative control scheme, Fig 4.6 depicts the character performing a jump. This action is likely triggered by a specific hand gesture, though the details of this gesture aren’t provided here in Fig 4.3, Fig 4.4, Fig 4.5 and Fig 4.6 (directional gestures). Coming into play here , which presumably works in conjunction with the camera’s detection system. The camera captures the player’s hand movement, and Fig

4.5 interprets it as a jump command, propelling the character upwards within the game world. This unique approach to character control seamlessly integrates physical actions with in-game movements, fostering a more interactive and engaging game play experience.

Building on the game’s unique control scheme, Fig 4.6 showcases the character performing a rolling action. While the specific body gesture isn’t revealed here (figures 4.0 for starting pose, fig 4.1 and fig 4.2 for directional movements), Algorithm again



Figure 4.4: Right move

plays a central role.

We can hypothesize that Algorithm is designed to be versatile, interpreting specific hand movements beyond just jumping. This flexibility in gesture recognition allows for a wider range of character actions, creating a more dynamic and engaging game play experience.



Figure 4.5: jump



Figure 4.6: roll

## Application

* + 1. **Education**: Helps students learn through movement-based interactive activities, improving engagement and memory retention.
    2. **Fitness and Well-being**: Makes exercise fun, motivating users to stay active through gamified workouts.
    3. **Workplace Wellness**: Encourages physical activity among employees through playful fitness challenges.
    4. **Entertainment**: Create immersive and enjoyable gaming experiences using body based controls.

**Chapter 5**

# Future Scope and Conclusion

## 5.1 Future Scope

1. **Advanced Gesture Recognition:** Future development will focus on improving the accuracy and robustness of gesture recognition algorithms. Machine learning techniques including deep learning can be used to train models capable of recognizing a wide range of gestures with high accuracy under different lighting conditions and backgrounds 3D Motion.
2. **Tracking:** Although current approaches mainly on 2D image formation, future research could investigate 3D motion capture and analysis techniques. This could include the use of depth- sensing cameras or stereo vision systems to track movement in three-dimensional. space, allowing for more sophisticated interactions with games. Augmented Reality (AR).
3. **Games:** Graphics techniques can be fused with AR technology to create games that overlay virtual objects in a real-world environment captured by a camera Players can interact with these virtual objects using gestures or movements, and for the boundaries between physical and digital world’s.
4. **Social and cooperative gaming experiences**: Visualization techniques can lead to multiplayer games where players interact with gestures and movements captured by their cameras This can create a new social gaming experience where players physically cooperate or compete activities regardless of physical location Interaction with Virtual.

## 5.2 Conclusion

In conclusion, body action games hold transformative potential in reshaping the way we experience and interact with video games. Unlike traditional games, these experiences bridge the digital and physical worlds, bringing players closer to the action in a truly immersive way. By simulating real-world physical interactions, body action games go beyond just playing—they allow players to actively live within their virtual environments, making the gaming experience both more engaging and deeply personal.

One of the most exciting aspects of body action games is their adaptability. As these games increasingly incorporate adaptive algorithms, they can mold challenges to fit each player’s unique skill level and physical capacity, making gameplay enjoyable yet appropriately demanding.

Beyond entertainment, body action games have real-world benefits. By encouraging players to move and engage physically, they contribute to healthier lifestyles and offer an enjoyable way to stay active and hands-on learning experience that traditional methods often struggle to achieve.

As technology advances, body action games will likely continue to evolve, opening up new possibilities for skill development, from fine-tuning motor skills to enhancing coordination and problem-solving abilities.

# Bibliography

1. V. Harini, V. Prahelika, I. Sneka, and P. A. Ebenezer, "Hand Gesture Recognition Using OpenCV and Python," in *Proc. of the International Conference*, 73, 55-60.
2. F. Khan, "Computer Vision Based Mouse Control Using Object Detection and Marker Motion Tracking," in *Proc. of the International Conference*, 37(12), 2436-2472 .
3. I. F. Ince, M. Socarras-Garzon, and T.-C. Yang, "Hand Mouse: Real Time Hand Motion Detection System Based on Analysis of Finger Blobs," in *Proc. of the International Conference*, 87, 771-779.
4. S. Singhvi, N. Gupta, and S. M. Satapathy, "Virtual Gaming Using Gesture Recognition Model," in *Proc. of the International Conference*, 6(5).
5. S. K. M., "Real Time Moving Vehicle Congestion Detection and Tracking using OpenCV," in *Proc. of the International Conference*, 19, pp. 1-17.
6. F. M. Inamdar, R. Jambhulkar, I. Patil, and A. Misal, "Snake Game with Motion Control Using Computer Vision," in *Proc. of the International Conference (2015)*.
7. P. Joshi and V. Godoy, *OpenCV by Example*. Vol. 1901.03107v1, pp. 24-40.