Tanmay Mohanty
Undergraduate Student
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Computer Vision-Based Controller to Play

RedBall 4

GitHub Repository Link:
https://github.com/tanmay-mohanty-ignited/Computer-Vision-Based-Game-Controller-RedBall-4

ABSTRACT

The project aimed at building a computer vision-based controller to control a game, in our case, RedBall 4, which is an adventure-type game, where the subject is a Ball, and it passes through various obstacles. The model detects the upper body parts and processes their coordinates to initiate certain actions. The actions are then converted to game inputs to control the ball and eventually play the game, all through a camera device, without touching your keyboard.

INDEX

1. Introduction to the problem statement
2. Software requirements
3. Demonstration images
4. Theory
5. Conclusion

LIST OF FIGURES

1. Figure 1: Demonstration of the gameplay and the procedure
2. Figure 2: MediaPipe Pose Module software definitions of landmarks

INTRODUCTION TO PROBLEM STATEMENT

The problem statement can be divided into 4 parts :

Analysis of the controls and actions required

The game needs various inputs and according to that, a number of body actions need to be planned in order to initiate those actions and map them to the inputs.

Detection of Upper Body Coordinates

Use of computer vision to track the body parts and map them to their coordinates, relative to the capture screen, for further processing.

Processing coordinates to actions

The coordinates obtained are to be used to arrive at a mathematical relation, and upon its satisfaction, the flags to certain actions are activated.

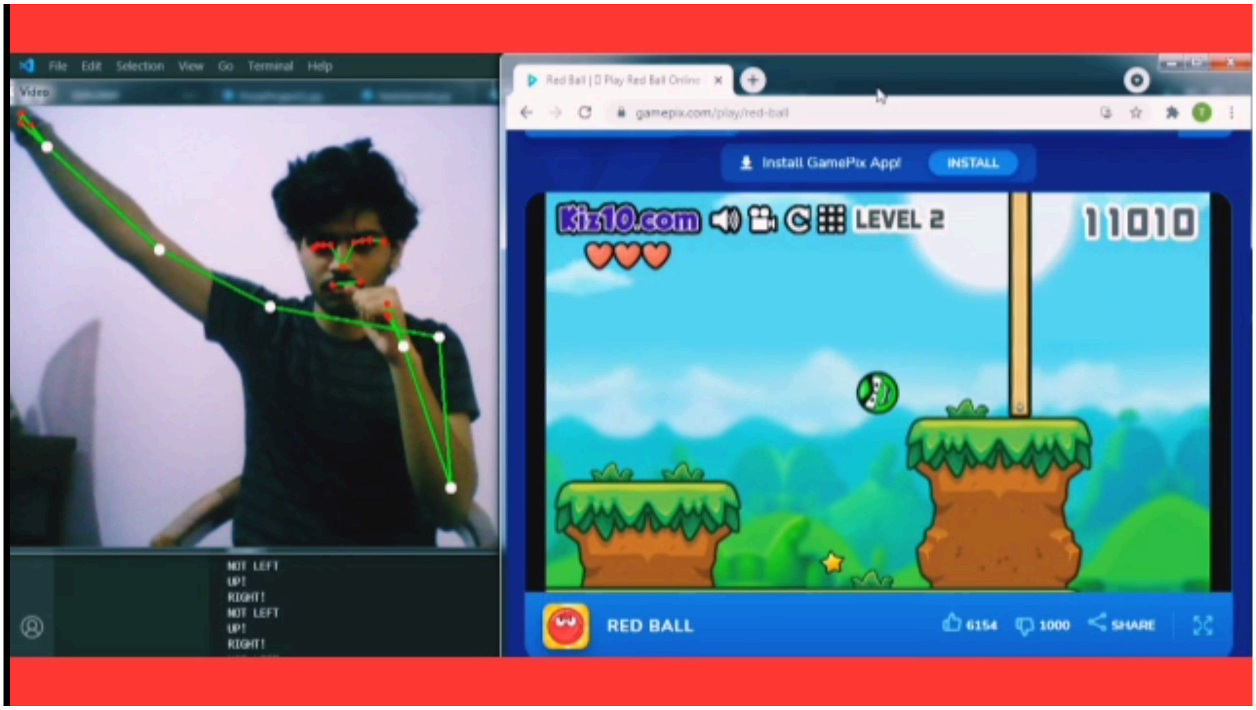
Conversion of flags to control the game

Flags after being activated are to be converted to specific inputs as per the game, in a form that it understands.

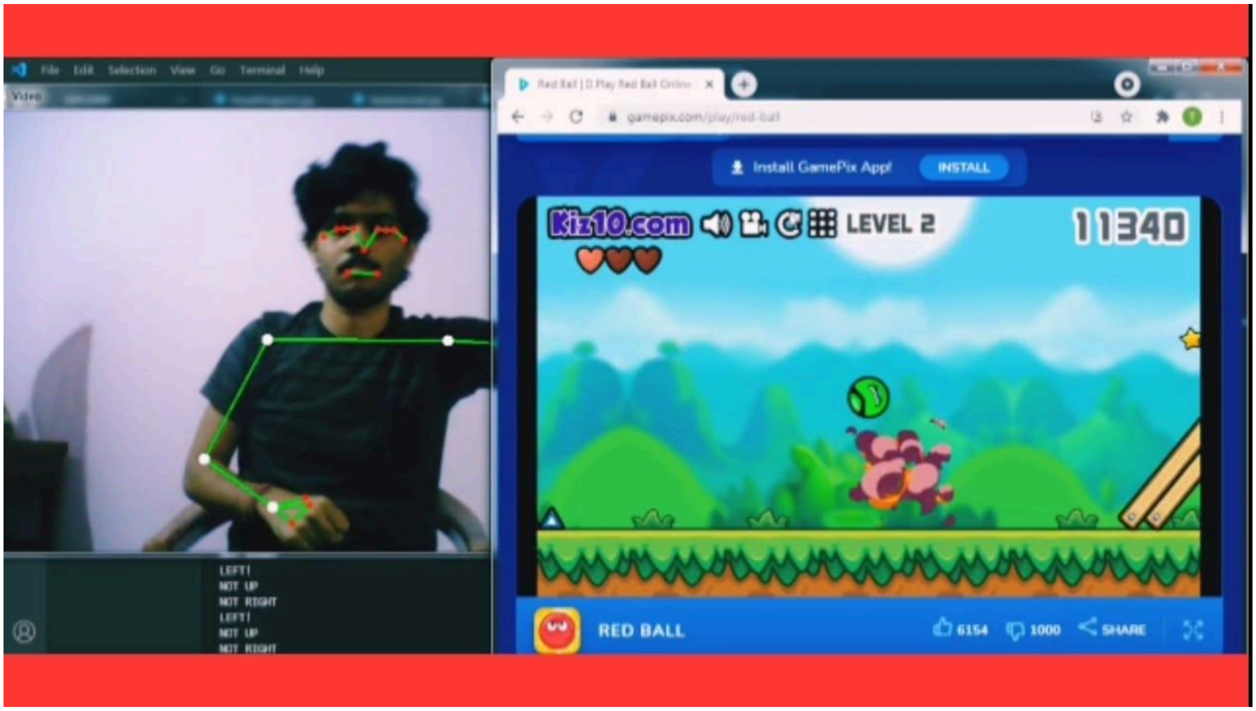
SOFTWARE REQUIREMENTS

1. Python 3
2. Any standard IDE for writing and running Python Code (Visual Studio Code, in our case)
3. Libraries: OpenCV, MediaPipe (Pose Module), Pynput.

DEMONSTRATION IMAGES


File Edit Selection View Go Terminal Help
Video
Red Ball | D Play Red Ball Online
gamepix.com/play/red-ball
Install GamePix App! INSTALL
Kiz10.com
LEVEL 2
11010
RED BALL
6154
1000
SHARE

```
NOT LEFT
UP!
RIGHT!
NOT LEFT
UP!
RIGHT!
```


File Edit Selection View Go Terminal Help
Video
Red Ball | D Play Red Ball Online
gamepix.com/play/red-ball
Install GamePix App! INSTALL
Kiz10.com
LEVEL 2
11340
RED BALL
6154
1000
SHARE

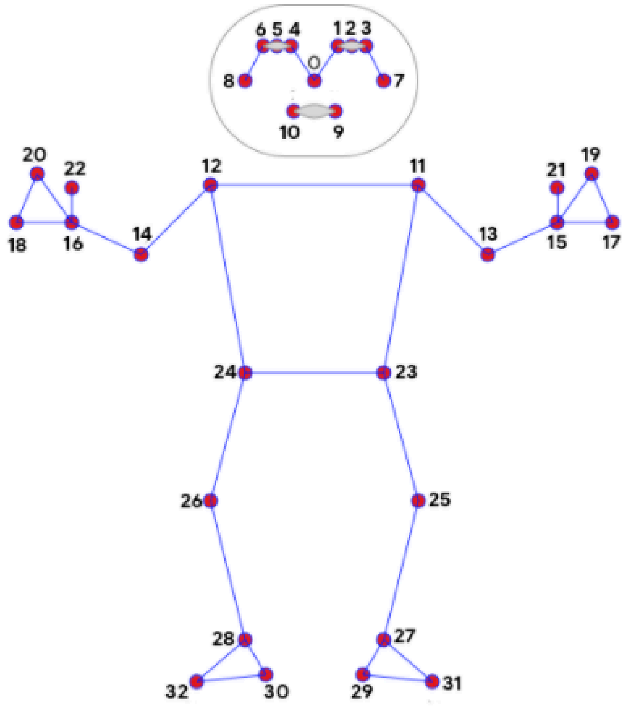
```
LEFT!
NOT UP
NOT RIGHT
LEFT!
NOT UP
NOT RIGHT
```

Figure 1: Demonstration of the Process to sit in front of the camera and control the game, the IMage also shows the landmark points of body parts detection

Link to Demo Video: https://youtu.be/qBx-1Z10TxA

THEORY

The detection is based on Google's MediaPipe Pose Module, which is backed by Machine Learning models, to detect the body pose and provide the coordinates of 33 landmarks.



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Figure 2: Landmarks definition and index mapping of different body landmarks based on MediaPipe Pose Module

The coordinates are obtained for all 33 landmarks. The upper body coordinates targeted in our program are: Right Shoulder, Left Shoulder, Right Elbow, Left Elbow, Right Wrist, and Left Wrist.

Using coordinate geometry principles, I derived three relations, one on each arm and one for above shoulders move. The controls and motions are as follows:

1. Right Movement: The Right wrist is to be detected when it goes to the right of the right shoulder (away from the body).
2. Left Movement: The Left wrist is to be detected when it goes to the left of the left shoulder (away from the body).
3. Up Movement: When any one of the elbows (either right or left) goes above the line connecting the right and left shoulder coordinates.

The Pynput library and its Keyboard Module convert the flags that are activated on these specific body movements into keyboard key presses, for the UP, RIGHT, LEFT movements to control the ball movements in the game.

The live camera feed is taken into the OpenCV program and the processes above are processed on each frame of the video.

CONCLUSION

Learning Outcomes

Through this project, I learned about OpenCV and worked on it using Python. I learned various special functions in OpenCV, to manipulate image properties and detect colors and boundaries. I learned about using Pose Module in MediaPipe, to detect body pose and identify the coordinates of the defined landmarks. I applied coordinate geometry principles to derive the equations for detecting particular upper body movements. The next challenge was to convert these signals to keypresses, which was solved through the Pynput Keyboard Module.

REFERENCES AND IMPORTANT LINKS

Youtube Video (Demonstration) : https://youtu.be/qBx-1Z10TxA

OpenCV : Home - OpenCV

MediaPipe Pose Module : Pose - mediapipe (google.github.io)

Pynput : pynput · PyPI

GitHub Repository Link:
https://github.com/tanmay-mohanty-ignited/Computer-Vision-Based-Game-Controller-RedBall-4