A

Project Report

On

**Press The Button**

Submitted in partial fulfillment of the requirement for the Vth semester

**Bachelor of Computer Science**

By

**Neeraj Koshyari**

**Neeraj Gaira**

Under the Guidance of

**Mr. Devesh Pandey**

**Assistant Professor**

**Department of CSE**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS**

**SATTAL ROAD, P.O. BHOWALI,**

**DISTRICT- NAINITAL-263132**

**2023- 2024**

**STUDENT’S DECLARATION**

We, **Neeraj koshyari and Neeraj Gaira** here-by declare the work, which is being presented in the project, entitled “**Press The Button Game**” in partial fulfillment of the requirement for the award of the degree **B.Tech** in the session **2023-2024**, is an authentic record of our own work carried out under the supervision of “**Mr. Devesh Pandey”, Assistant Professor, Department of CSE, Graphic Era Hill University, Bhimtal.**

The matter embodied in this project has not been submitted by us for the award of any other degree.

Date: ………… ……………….

(Full signature of student)

**CERTIFICATE**

**The project report entitled “Press the Button game” being submitted by Neeraj Koshyari and Neeraj Gaira to Graphic Era Hill University Bhimtal Campus for the award of bonafide work carried out by them. They have worked under my guidance and supervision and fulfilled the requirement for the submission of report.**

**(Mr. Devesh Pandey) (Dr. Ankur Bisht)**

**Project Guide (HOD, CSE Dept.)**

**ACKNOWLEDGEMENT**

We take immense pleasure in thanking Honorable **“Mr. Devesh Pandey”** (**AssistantProfessor,CSE, GEHU Bhimtal Campus**) to permit me and carry out this project work with his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance and useful suggestions that helped me to develop as a creative researcher and complete the research work, in time.

Words are inadequate in offering my thanks to GOD for providing me everything that we need. We again want to extend thanks to our President **“Prof. (Dr.) Kamal Ghanshala”** for providing us all infrastructure and facilities to work in need without which this work could not be possible.

Many thanks to Professor **“Dr. Manoj Chandra Lohani”** (Director Gehu Bhimtal),

other faculties for their insightful comments, constructive suggestions, valuable advice, and time in reviewing this thesis.

Finally, yet importantly, we would like to express my heartiest thanks to our beloved parents,

for their moral support, affection and blessings. We would also like to pay our sincere thanks to all our friends and well-wishers for their help and wishes for the successful completion of this research.

**Neeraj Koshyari**

**Neeraj Gaira**

**ABSTRACT**

The project aims to create a virtual adaptation of the popular interactive game "Press the Button." The virtual implementation will maintain the strategic and suspenseful elements of the physical game while leveraging digital capabilities to enhance the gaming experience. Players will be able to enjoy the game remotely, fostering social interaction and collaboration in a virtual setting.

This report is an introduction to the Press the button game in Python language. Anybody, who doesn’t know even the basics of Press the button game in python will certainly be able to understand and gain great knowledge from this report. Our project aim is to detect the Hand of a person by which the person will be able to play a game. In the game a person have to touch the buttons virtually within the distance of less then 60 cm in front of the camera and the score will be calculated accordingly.

**Table of CONTENTS**

**DECLARATION……………………………..………………………………………..II**

**CERTIFICATE ………………………………………………………………………III**

**ACKNOWLEDGEMENT ..……………………...…………………………………..IV**

**ABSTRACT …………………………………………………………………………...V**

**Table Of Contents……………………………………………………………………VI**

**CHAPTER 1 INTRODUCTION…………………………………….………………….8**

* 1. INTRODUCTION………………………………………………………….8
  2. OBJECTIVE…………………………..……….…………………………..9

**CHAPTER 2 TOOLS.PLATFORN, H/W AND S/W REQUIREMENT…………10**

2.1 TOOL……………………………………………………………………10

2.2 PLATFORM……………………………………………………………….10

2.3 HARDWARE REQUIREMENT SPECIFICATION………………………10

2.4 SOFTWARE REQUIREMENT SPECIFICATION……………………….10

**CHAPTER 3 DATA FLOW DIAGRAM (DFD)…………………….…………….......11**

3.1 FLOWCHART REPRESENTATION OF THE PROGRAM……………..11

**CHAPTER 4 HAND LAND MARKING………………………..……………………...13**

4.1 MARKING DIAGRAM…………….……………………………………13

**CHAPTER 5 PROBLEM STATEMENT………….……………………………….……..13**

5.1 HAND TRACKING………………...……………………………………….13

5.2 VIRTUAL BUTTON INTRACTION……………..………………………...13

5.3 DISTANCE CALCULATION…………………………………………..…..13

**CHAPTER 6 CODE……………………………………………………………………15**

**CHAPTER 7 TESTTING………………………………………………………………19**

**CHAPTER 8 ENHANCEMENTS……………………………………………………..19**

**CHAPTER 9 CONCLUSION………………………………………………………......20**

**CHAPTER 10 REFERENCES…………………………………………………………20**

**INTRODUCTION**

The provided code implements a simple interactive game called "Press the Button" using the OpenCV and cvzone libraries in Python. The game involves tracking hand movements in the camera frame and pressing a virtual button when the hand is in the specified range.

**Code Structure:**

* **Imports:** The necessary libraries such as cv2, cvzone, numpy, math, and time are imported.
* **Webcam Setup:** The code initializes the webcam, sets its resolution, and flips the video feed for a more natural viewing experience.
* **Hand Tracking:** Utilizes the Hand Detector class from the cvzone library to track the hand in the camera frame.
* **Find Function:** Defines a function to calculate the distance of the hand from the camera and convert it into centimeters using a polynomial function.
* **Game Variables:** Initializes various variables, including the button position, colour, score, and game duration.
* **Main Loop:** The core of the game, where hand tracking, button pressing, and game logic are implemented.

**Game Logic:**

* Hand Tracking: The code tracks the user's hand, calculates the distance from the camera, and displays the distance on the screen.
* Button Pressing: When the hand is within a specific range and positioned over the button, the button changes color, and the score increases.
* Game Over: After a specified time (totalTime), the game ends, and the final score is displayed. Users can restart the game by pressing the 'r' key.

**OBJECTIVE**

The primary objective of the provided code is to implement an interactive game called "Press the Button" using computer vision techniques. The code leverages the OpenCV and cvzone libraries to enable hand tracking through a webcam, creating a virtual button that users can press within a specified range. The main objectives include:

Hand Tracking: Utilize the HandDetector class from the cvzone library to accurately track the movements of the user's hand in the webcam feed.

Virtual Button Interaction: Implement a virtual button that changes color when the user's hand is within a designated range, simulating the act of pressing a physical button.

Distance Calculation: Utilize a polynomial function to convert the calculated distance of the user's hand from the camera into centimeters, providing a realistic measurement.

Game Logic: Develop the core game logic to determine when the virtual button is pressed, updating the score accordingly. Implement a time-based game mechanism where the game concludes after a specified duration.

User Interface: Display a virtual button on the screen, along with a Head-Up Display (HUD) that shows the remaining time and the player's score, enhancing the user experience.

Game Over Handling: Implement a mechanism to conclude the game, displaying the final score and providing an option to restart the game upon user input.

Usability and Engagement: Create an engaging and interactive game that involves users in a virtual environment, fostering hand-eye coordination and providing an entertaining experience.

Code Readability and Structure: Ensure that the code is well-organized, readable, and commented, allowing for easy understanding and potential future modifications.

The overarching objective is to deliver a functional and enjoyable game that effectively combines computer vision capabilities with user interaction, providing a unique and entertaining experience for players.

**TOOLS/ PLATFORM, HARDWARE AND SOFTWARE REQUIREMENT SPECIFICATIONS**

Tools

* Android Studio(software)
* Android Phone

Platform

* Windows 7/8/10/11

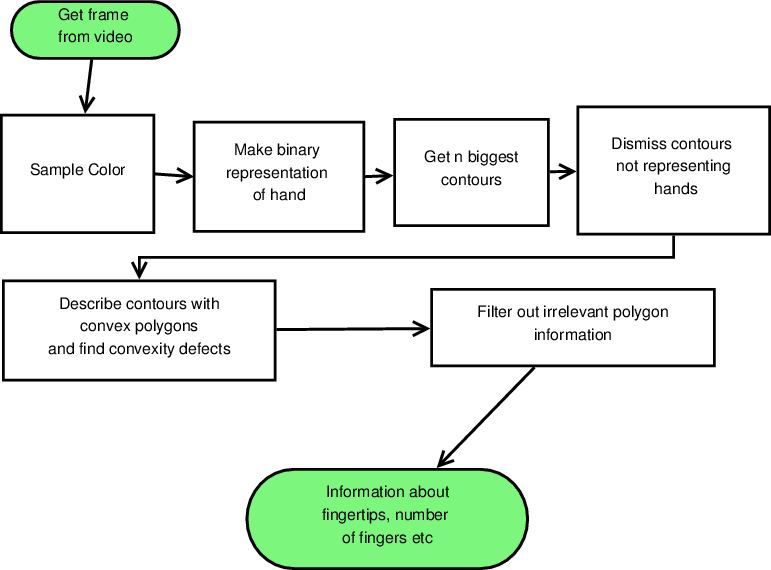
Hardware Requirement Specification

|  |  |
| --- | --- |
| **Criterion** | **Description** |
| **Disk Space** | 500 MB disk space for Android Studio, at least 1.5 GB for Android SDK, emulator system images, and caches |
| **RAM** | 3GB RAM minimum, 8 GB RAM recommended, plus 1GB for the Android Emulator |
| **Python Version** | Python 3.9.17 |

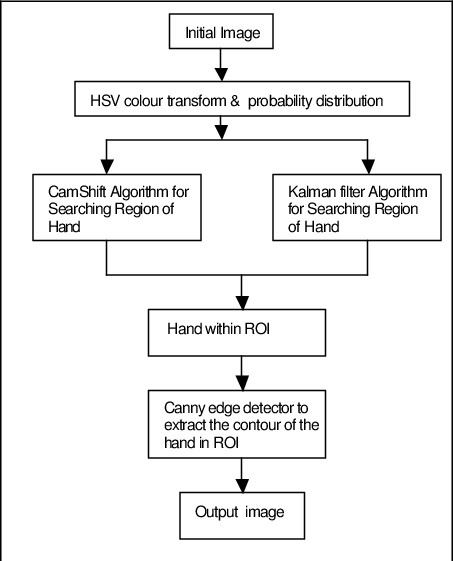
**Software Requirement Specification**

Latest version of Python.

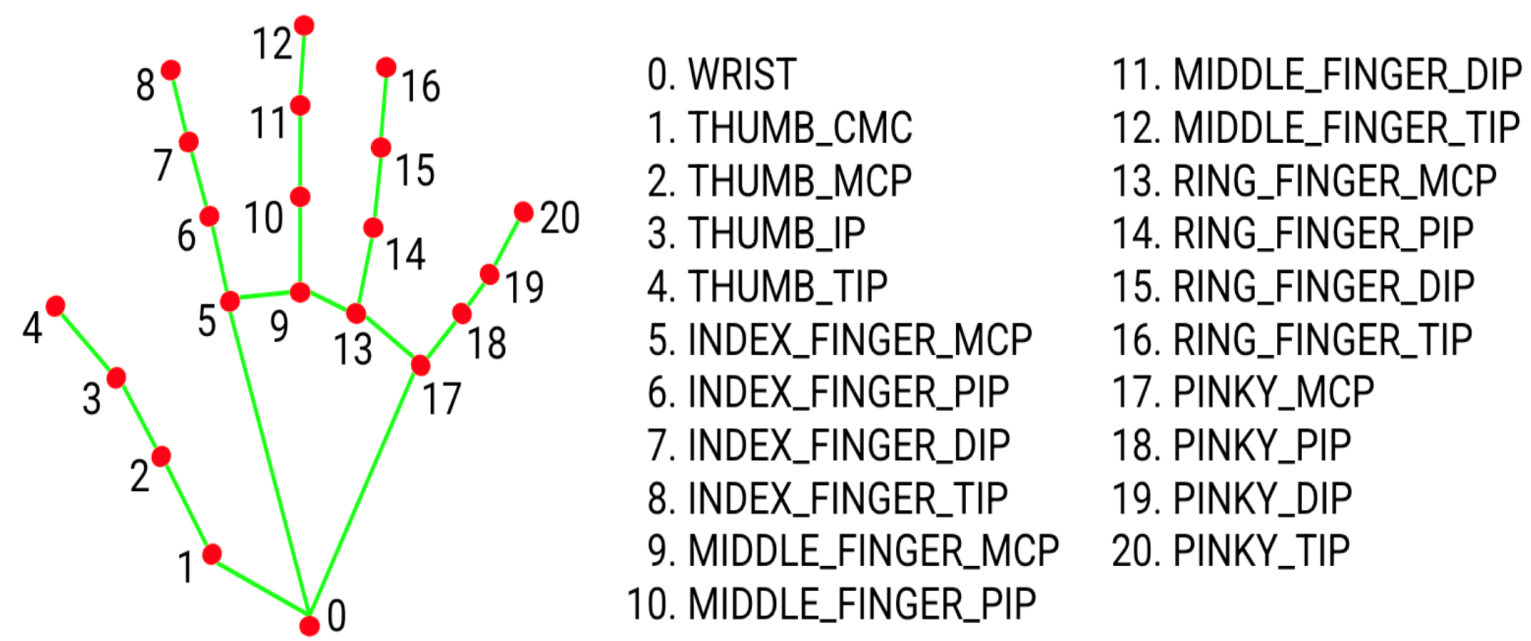
**Data Flow Diagram**

****

**Reference: -**[**https://sa-cybernetics.github.io/blog/2013/08/12/hand-tracking-and-recognition-with-opencv/**](https://sa-cybernetics.github.io/blog/2013/08/12/hand-tracking-and-recognition-with-opencv/)

****

**Reference:-** [**https://www.researchgate.net/figure/Flow-chart-of-hand-tracking-system\_fig1\_37616167**](https://www.researchgate.net/figure/Flow-chart-of-hand-tracking-system_fig1_37616167)

**Hand Land marking**

**PROBLEM STATEMENT**

The problem at hand is to develop a computer vision-based interactive game called "Press the Button." The primary goal is to create a virtual environment where users can engage with the game by tracking their hand movements through a webcam and simulating the act of pressing a virtual button. The specific challenges and requirements within this problem statement are as follows:

**Hand Tracking:** Implement an effective hand tracking mechanism using the HandDetector class from the cvzonelibrary.Ensure accurate and real-time tracking of the user's hand movements within the webcam feed.

**Virtual Button Interaction:** Create a virtual button on the screen that responds dynamically to the position of the user's hand.

Implement logic to detect when the user's hand is within a predefined range of the virtual button, simulating the act of pressing it.

**Distance Calculation:** Develop a mathematical model, such as a polynomial function, to accurately convert the distance of the user's hand from the camera into centimeters.

Ensure that the distance calculation provides a realistic representation of the user's interaction with the virtual environment.

**Game Logic:** Implement the core game logic that determines when the virtual button is pressed based on the user's hand position and distance.

Incorporate a time-based game mechanism, defining the duration of the game and managing the progression of the game over time.

**User Interface:** Design a user-friendly interface that displays the webcam feed, the virtual button, and a Head-Up Display (HUD) showing essential information such as the remaining time and the player's score.

**Game Over Handling:** Implement a clear and informative game over screen that displays the final score and provides an option for the user to restart the game.

The successful resolution of these challenges will result in a functional and entertaining computer vision-based game that effectively simulates the interaction of users with a virtual button in a dynamic environment.

**CODE**

import cv2

from cvzone.HandTrackingModule import HandDetector

import math

import numpy as np

import cvzone

import random

import time

# Webcam frame

cap = cv2.VideoCapture(0)

cap.set(3, 1280) #width

cap.set(4, 720) #height

#Hand Detector

detector = HandDetector(detectionCon=0.8, maxHands=1)

# Find Function

# x is the raw distance y is the value in cm

x = [300, 245, 200, 170, 145, 130, 112, 103, 93, 87, 80, 75, 70, 67, 62, 59, 57]

y = [20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100]

coff = np.polyfit(x,y,2) #second order polynomial function to find cofficient a,b,c

# y = ax^2 + bx+c

# Game Variables

cx, cy = 250, 250 #initial button position

color = (0,0,255) #initial button color

counter = 0 #tell wheter the button is pressed or just the hand is in the frame( < 60 distance)

score = 0 #initial score

timeStart = time.time()

totalTime = 22

# Loop

while True:

success, img=cap.read()

img = cv2.flip(img ,1)

if time.time()-timeStart < totalTime:

hands, img = detector.findHands(img,draw=False)

if hands:

lmList = hands[0]['lmList']

x, y, w, h = hands[0]['bbox'] #bounding box x-axes, y-axes, width and height

x1,\_,y1 = lmList[5] #index fingure start point

x2,\_,y2 = lmList[17] #pinky fingure start point

distance = math.sqrt((y2-y1)\*\*2 + (x2-x1)\*\*2) #to resolve hand rotation problem

A, B, C =coff

distanceCM = (A \* (distance\*\*2)) + (B \* distance) + C # equation, y = ax^2 + bx+c where x is distance in centimeter (cm)

#print(distanceCM, distance)

# Hand is in the range or not

if distanceCM < 60 and x < cx < x + w and y < cy < y + h:

counter = 1

#display cordinate in Hand

cv2.rectangle(img, (x, y), (x + w, y + h), (255,0,255), 3) #to show the rectangle in image of hand

cvzone.putTextRect(img, f'{int(distanceCM)} cm', (x+8, y-10)) #show distance in image of hand

# if hand in range change color and reset after 4 sec

if counter:

counter+=1

color = (0,255,0)

if counter == 4:

cx = random.randint(100, 1100) #random positon for button

cy = random.randint(100, 600)

color = (0,0,255)

score+=1 #increase score after one button press

counter = 0 #reset counter

# Draw Button

cv2.circle(img, (cx, cy), 32, (0,0,0), cv2.FILLED)

cv2.circle(img, (cx, cy), 28, (255,255,255), cv2.FILLED)

cv2.circle(img, (cx, cy), 20, color, cv2.FILLED)

# Game Head Up Display

cvzone.putTextRect(img, f'Time: {int(totalTime-(time.time()-timeStart))}', (60, 75), scale=3, offset=20, colorR=(214,237,23),border=2) #Time bar

cvzone.putTextRect(img, f'Score: {str(score).zfill(2)}', (1000, 75), scale=3, offset=20,colorR=(214,237,23),border=2) #Points bar

else:

cvzone.putTextRect(img, 'Game Over', (400, 400), scale=5, offset=30, thickness=7,colorR=(214,237,23),border=2)

cvzone.putTextRect(img, f'Your Score: {str(score).zfill(2)}', (440, 500), scale=3, offset=20,colorR=(214,237,23),border=2)

cvzone.putTextRect(img, 'Press \'r\' to restart', (460, 575), scale=2, offset=10,colorR=(214,237,23),border=2)

cv2.imshow("Image",img)

key = cv2.waitKey(1)

if key == ord('r'):

timeStart = time.time()

score = 0

**TESTING**

**TEAM INTERATION**

The following describes the level of team interaction necessary to have a successful product.

* The test team will work closely with the Development team to acviece a high quality design and user interface specifications based on customer requirements. The test team is responsible for visualizing test cases and raising quality issues and concerns during meetings to address issues early enough in the development cycle.
* The Test team will work closely with Development Team to determine whether or not the application meets standards for completeness. If an area is not acceptable for testing, the code complete date will be pushed out, giving the developers additional time to stabilize the area.
* Since the application with a back-end system component, the Test Team will need to include a plan for integration testing. Integration testing must be executed successfully prior to system testing.

**TESTING OBJECTIVE**

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program.

**ENHENCEMENTS**

* Instead of Button moving bugs can be added
* Background music can be while playing the game.
* Can be implement in GUI bases.
* Can be launched in VR mode.

**CONCLUSION**

We have successfully implemented the Press the Button game in Python language.

**REFERENCES**

1. <https://www.geeksforgeeks.org/opencv-python-tutorial/>
2. <https://www.computervision.zone/courses/hand-distance-measurement/>