

# **Project Based Learning-II**

(Guidelines and Work Book)

## **Course Code: 210258**

(2019 Course)

## **Second Year Engineering**

Year 2020 - 2021

Group ID:

--	--	--	--

Team Members: 1. Mousam Singh (SE64)

2. Nishant Shinde (SE69)

3. Vedant Garode (SE66)

Project Title : Gesture Controlled Media Player

Name of Mentor: Neeta Gawande



## **DEPARTMENT OF COMPUTER ENGINEERING**

**DR. D. Y. PATIL INSTITUTE OF TECHNOLOGY, PIMPRI, PUNE**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**2019 - 2020**

For better learning experience, along with traditional classroom teaching and laboratory learning; project based learning has been introduced with an objective to motivate students to learn by working in group cooperatively to solve a problem, Project-based Learning (PBL) is a student centric pedagogy that involves a dynamic classroom approach in which it is believed that students acquire a deeper knowledge through active exploration of real world challenges and problems. Students learn about a subject by working for an extended period of time to investigate and respond to a complex question, challenge or a problem. It is a style of active learning and inquiry-based learning.(Reference: Wikipedia). Problem based learning will also redefine the role of teacher as mentor in learning process. Along with communicating knowledge to students, often in a lecture setting, the teacher will also to act as an initiator and facilitator in the collaborative process of knowledge transfer and development.

This is a recommended workbook for PBL that will serve the purpose and facilitate the job of students, mentor and coordinator. This workbook will reflect accountability, punctuality, technical writing ability and work flow of the work undertaken.

**(For circulation at BoS Computer Engineering, Savitribai Phule Pune University only)**

---

## CERTIFICATE

This is to certify that Mr./ Ms. \_\_\_\_\_

Group No. \_\_\_\_\_ Division \_\_\_\_\_ Branch \_\_\_\_\_ has successfully completed the  
work associated with **Project Based Learning II (210258)** titled as

\_\_\_\_\_ and has submitted the work book associated under my supervision, in the partial fulfillment of Second Year Bachelor of Engineering(Choice Based Credit System) (2019 course) of Savitribai Phule Pune University.

Date:

Place:

Guide  
(Name & Sign)

Head  
(Name & Sign)

Principal  
(Name & Sign)

**1. Evaluation and Assessment Sheet** (To be filled in my mentor)

<b>Sr. No.</b>	<b>Details</b>	<b>Maximum Marks</b>	<b>Marks Obtained</b>
1.	Problem Identification (Idea Inception)	<b>10</b>	
2.	Problem Analysis (Requirement Gathering)	<b>15</b>	
3.	Proposed Solution Model/Design/ Process / prototype	<b>20</b>	
4.	Technology Solution Model	<b>15</b>	
5.	Expected Outcomes	<b>05</b>	
6.	Implementation and Testing	<b>10</b>	
7.	Regularity (Attendance + Weekly Progress Reporting)	<b>10</b>	
8.	Awareness /Consideration of - Environment/ Social /Ethics/ Safety measures/Legal aspects	<b>05</b>	
9.	Contest Participation/ publication	<b>05</b>	
10.	Report	<b>05</b>	
<b>Total Marks</b>		<b>100</b>	
<b>Date:</b>			
<b>Name &amp; Sign of Mentor</b>			

## **2. Project Information Sheet**

Project ID					
Title	Gesture Controlled Media Player				
Problem Statement	To Control Media[ Player Without using keyboard and mouse.				
Name of Mentor	Neeta Gawande				
Group Members	Division	Roll No.	Name	Mobile Number	Email ID
	B	SEB64	Mousam Singh	8446970633	singhmausam76@gmail.com
	B	SEB69	Nishant Shinde	9022799236	nishantshinde756@gmail.com
	B	SEB66	Vedant Garode	8149336088	Vedantgarode11@gmail.com

### **3. Continuous Assessment and Remarks Sheet**

Problem Identification (Idea Inception) –

To operate various computer operation without touching mouse or keyboard with the help of hand gestures.

Problem Analysis (Requirement Gathering) –

We could use sensors for detecting hand gestures but we wanted only software to do the work.

We thought of doing deep learning to do this work but it was very complex and that complex structure was not worth to just automate Media Player .So we Used we used Image processing for this job.

Proposed Solution Model/Design/ Process / prototype –

We took input from webcam as image and made a colour adjustment slider to adjust the required(Skin colour) to track it . As soon as we define the required color it automatically creates mask in black and white image .Then by that masked image we draw the contours around the hand to highlight as well as do the calculation over the certain part of hand.To find some proper gesture we just added defects option where initially defects was 0 and as soon as we find finger in image it will count no.of finger as shown. We added math calculation such it was counting defects based on angle between 2 contours. If angle between 2 finger in less than 90 then we calculate it as 2 finger and as many defects get added no of finger increases

Technology Solution Model-

OpenCV Lib is good for small projects like this . It uses image processing like tracking objects and colours ,adding text , images on webcam or Picture . OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

Expected Outcomes-

We designed our code such that it will show number of finger as well as do the task assigned to it.

Such as if code sees all 5 finger than it pauses the media player .

for 2 finger it increses sound of media player. For 3 finger it decreases its sound. For 1 finger it mutes the sound and so on .

Implementation and Testing-

We used VLC media player to test out code and it worked perfectly as it should be.we tested it on few other laptops webcams and it glitch on some other low end laptops .we couldn't find solution for that but it worked on most of the new generation laptops

Regularity (Attendance + Weekly Progress Reporting) –(To be filled by Teacher)

Awareness /Consideration of -Environment/ Social /Ethics/ Safety measures/Legal aspects-(To be filled by teacher)

Contest Participation/ publication-(To be filled by teacher)

Report –(To be filled by Teacher)

**4. Project Monitoring/ Progress Information Sheets** (one sheet per week, 3-4 weeks sheet should be added)

<b>Week 1</b>
<b>Date</b>
<b>Current Work phase of project-</b> Thinking the idea for project
<b>Discussions Held</b> We had a group call to discuss about the idea for project . we came up with a lot of idea but most of them required high programming knowledge so we thought to use already build library to enhance the accessibility of code as well as it will be easy to work on such library .As there are already lot of problem solved on the discussion forums .we didnt finalized our project yet but we made a list of projects on which we can work.
<b>Progress till Date</b> Not that much .Just sorting different ideas by their difficulty and accessibility over the internet.
<b>Remark</b>
<b>Sign of Mentor</b>

<b>Week 2</b>
<b>Date</b>
<b>Current Work phase of project-</b> Finalized idea of our project
<b>Discussions Held</b> We discussed about the functions which can be done using our program and we thought of making gesture controlled rock paper scissors game but its gestures where very hard to track so we moved to media player idea .
<b>Progress till Date</b> Finalized project idea started gathering resources regarding the project
<b>Remark</b>



<b>Sign of Mentor</b>
-----------------------

<b>Week 3</b>
---------------

<b>Date</b>
-------------

<b>Current Work phase of project-</b>
---------------------------------------

Writing code for our project
------------------------------

<b>Discussions Held</b>
-------------------------

By the help of documentation and reference videos we figured out the algorithm of our code .later we started writing the code and as expected it didn't run for 2 days .So with the help of more documentation we figured out the errors and we was able to create colour adjustment window.
--

<b>Progress till Date</b>
---------------------------

Took input from Webcam and done colour adjustment on it .
---

<b>Remark</b>
---------------

<b>Sign of Mentor</b>
-----------------------

<b>Week 4</b>
---------------

<b>Date</b>
-------------

<b>Current Work phase of project-</b>
---------------------------------------

Testing phase
---------------

<b>Discussions Held</b>
-------------------------

Done with our source code still there are some few errors left we are still figuring it out .code is running perfectly and we are testing it on different webcams to improve its accessibility.
---

<b>Progress till Date</b>
---------------------------

Done with source code .Testing phase running .
--

<b>Remark</b>
---------------

<b>Sign of Mentor</b>
-----------------------

# Code-

```
#GroupB4
#Title - Hand Gesture controlled Volume controller using OpenCV
#Team Memebers -
    #Mausam Singh
    #Nishant Shinde
    #Vedant Garode

import cv2
import numpy as np
import math
import pyautogui as p

cam = cv2.VideoCapture(0, cv2.CAP_DSHOW)

def empty(x):
    pass

cv2.namedWindow("Filter",cv2.WINDOW_NORMAL)
cv2.resizeWindow("Filter",300,100)
cv2.createTrackbar("Thresh", "Filter", 0, 255, empty)
cv2.createTrackbar("L_Hue", "Filter", 0, 255, empty)
cv2.createTrackbar("U_Hue", "Filter", 255, 255, empty)
cv2.createTrackbar("L_Sat", "Filter", 0, 255, empty)
cv2.createTrackbar("U_Sat", "Filter", 255, 255, empty)
cv2.createTrackbar("L_Value", "Filter", 0, 255, empty)
cv2.createTrackbar("U_Value", "Filter", 255, 255, empty)

while True:
    suc , frame = cam.read()
    frame = cv2.flip(frame, 2)
    frame = cv2.resize(frame, (600, 500))
    crop_image = frame[1:500, 0:300]

    hsv = cv2.cvtColor(crop_image, cv2.COLOR_BGR2HSV)

    l_h = cv2.getTrackbarPos("L_Hue", "Filter")
    u_h = cv2.getTrackbarPos("U_Hue", "Filter")
    l_s = cv2.getTrackbarPos("L_Sat", "Filter")
    u_s = cv2.getTrackbarPos("U_Sat", "Filter")
    l_v = cv2.getTrackbarPos("L_Value", "Filter")
    u_v = cv2.getTrackbarPos("U_Value", "Filter")

    lower_bound = np.array([l_h, l_s, l_v])
    upper_bound = np.array([u_h, u_s, u_v])

    mask = cv2.inRange(hsv, lower_bound, upper_bound)

    filtr = cv2.bitwise_and(crop_image, crop_image, mask=mask)

    mask1 = cv2.bitwise_not(mask)
    m_g = cv2.getTrackbarPos("Thresh", "Filter")
    ret, thresh = cv2.threshold(mask1, m_g, 255, cv2.THRESH_BINARY)
    dilata = cv2.dilate(thresh, (3, 3), iterations=6)

    cnts, hier = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    try:
        cm = max(cnts, key=lambda x: cv2.contourArea(x))
        epsilon = 0.0005 * cv2.arcLength(cm, True)
        data = cv2.approxPolyDP(cm, epsilon, True)
        hull = cv2.convexHull(cm)
        cv2.drawContours(crop_image, [cm], -1, (50, 50, 150), 2)
```

```

cv2.drawContours(crop_image, [hull], -1, (0, 255, 0), 2)

hull = cv2.convexHull(cm, returnPoints=False)
defects = cv2.convexityDefects(cm, hull)
count_defects = 0

for i in range(defects.shape[0]):
    s, e, f, d = defects[i, 0]

    start = tuple(cm[s][0])
    end = tuple(cm[e][0])
    far = tuple(cm[f][0])

    a = math.sqrt((end[0] - start[0]) ** 2 + (end[1] - start[1]) ** 2)
    b = math.sqrt((far[0] - start[0]) ** 2 + (far[1] - start[1]) ** 2)
    c = math.sqrt((end[0] - far[0]) ** 2 + (end[1] - far[1]) ** 2)
    angle = (math.acos((b ** 2 + c ** 2 - a ** 2) / (2 * b * c)) * 180) / 3.14

    if angle <= 50:
        count_defects += 1
        cv2.circle(crop_image, far, 5, [255, 255, 255], -1)

print("count==", count_defects)

if count_defects == 0:
    cv2.putText(frame, " ", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255),
2)
elif count_defects == 1:
    p.press("space")
    cv2.putText(frame, "Play/Pause", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, (0,
0, 255), 2)
elif count_defects == 2:
    p.press("up")
    cv2.putText(frame, "Volume UP", (5, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, (0,
0, 255), 2)
elif count_defects == 3:
    p.press("down")
    cv2.putText(frame, "Volume Down", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,
(0, 0, 255), 2)
elif count_defects == 4:
    p.press("right")
    cv2.putText(frame, "Forward", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0,
255), 2)
else:
    pass

except:
    pass

cv2.imshow("Thresh", thresh)
cv2.imshow("Result", frame)

key= cv2.waitKey(33) & 0xFF
if key == ord('q'):
    break
cam.release()
cv2.destroyAllWindows()

```

## Output -



