

Source code

```
#pen and paint
# features:
# multiple colours _done_
# seperate pain screen _done_
# erase _done_
# clean all _done_
# image or white background ? _done_
# save the final image _done_
# 3 different brush sizes _done_
# wait feature i.e. holding the screen to take brake from drawing operation _done_

#libraries
import cv2
import numpy as np
from printy import printy
#*****
#opening program
printy('[cBD]*****WELCOME TO PEN AND PAINT*****
*****@')
printy('[mI]This program allows you to draw on the screen by moving the coloured cap in air.
@')
answer=input('Do you want to see help module :[y/n]')
if answer=='y':
    printy('[o]1.Press q for exit , s for save ,c for enabling pen and w for disabling pen.@ ')
    printy('[o]2.The default colour of cap that is detected is blue.@')
    printy('[o]3.A yellow circle gets drawn around the cap that acts as the pointer.@ ')
    printy('[o]4.Move the pointer circle to chhose from options given on top and also for drawi
ng on screen.@')
    printy('[o]5.To save paint window you need to specify the complete filename \ (eg:image.jp
g\ ) in terminal.@')
u=input('Do you want to open some image as background:[y/n] ')
if u=='y':
    address=input('Enter complete address of the image by opening the image and copying its
path: ')

#*****
#variable settings

# Define the upper and lower boundaries for a color to be considered "Blue"
blueLower = np.array([100, 60, 60])
blueUpper = np.array([140, 255, 255])
# Define a 5x5 kernel for erosion and dilation to remove the noises
kernel = np.ones((5, 5), np.uint8)
```

```

# bgr values of color
colors = [(255, 0, 0), (0, 255, 0), (0, 0, 255), (0, 255, 255), (255, 255, 255)]
# bgr values of color for mask
colors1 = [(255, 0, 0), (0, 255, 0), (0, 0, 255), (0, 255, 255), (0, 0, 0)]
colorIndex = 0
#default width of line
l=2
# wait
wait=0
x1=0
y1=0

*****
*****

#setting up the painting screen look

if u=='n' :
    paintWindow = np.zeros((471,636,3)) + 255 #creating an matrix #white colour
else :
    paintWindow = np.zeros((471,636,3))
    paintWindow=cv2.imread(address)

paintWindow = cv2.rectangle(paintWindow, (40,1), (140,65),(122,122,122), 2)
paintWindow = cv2.rectangle(paintWindow, (160,1), (200,65), colors[0], -1)
paintWindow = cv2.rectangle(paintWindow, (210,1), (250,65), colors[1], -1)
paintWindow = cv2.rectangle(paintWindow, (260,1), (300,65), colors[2], -1)
paintWindow = cv2.rectangle(paintWindow, (310,1), (350,65), colors[3], -1)
paintWindow = cv2.rectangle(paintWindow, (370,1), (410,65), (122,122,122), 2)
paintWindow = cv2.line(paintWindow,(500,32),(530,32),(0,0,0),2)
paintWindow = cv2.line(paintWindow,(550,32),(580,32),(0,0,0),4)
paintWindow = cv2.line(paintWindow,(600,32),(630,32),(0,0,0),6)
cv2.putText(paintWindow, "CLEAR ALL", (49, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.5
, (0, 0, 0), 2, cv2.LINE_AA)
cv2.putText(paintWindow, "RUB", (372, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0,
0), 2, cv2.LINE_AA)
cv2.namedWindow('Paint', cv2.WINDOW_AUTOSIZE)

*****
*****

#for webcam window

# Load the video
camera = cv2.VideoCapture(0)
camera.set(10,3000) # brightness
# This is the canvas on which we will draw upon
canvas = None
# Keep looping
while True:

```

```

# Grab the current paintWindow
(grabbed, frame) = camera.read()
frame = cv2.flip(frame, 1)
# Initilize the canvas as a black image

if canvas is None:
    canvas = np.zeros_like(frame)

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

# Add the coloring options to the frame

frame = cv2.rectangle(frame, (40,1), (140,65), (122,122,122), -1)
frame = cv2.rectangle(frame, (160,1), (200,65), colors[0], -1)
frame = cv2.rectangle(frame, (210,1), (250,65), colors[1], -1)
frame = cv2.rectangle(frame, (260,1), (300,65), colors[2], -1)
frame = cv2.rectangle(frame, (310,1), (350,65), colors[3], -1)
frame = cv2.rectangle(frame, (370,1), (410,65), (122,122,122), -1)
frame = cv2.line(frame,(500,32),(530,32),(0,0,0),2)
frame = cv2.line(frame,(550,32),(580,32),(0,0,0),4)
frame = cv2.line(frame,(600,32),(630,32),(0,0,0),6)
cv2.putText(frame, "CLEAR ALL", (49, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (25
5, 255, 255), 2, cv2.LINE_AA)
cv2.putText(frame, "RUB", (372, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255,
255), 2, cv2.LINE_AA)

# Check to see if we have reached the end of the video
if not grabbed:
    break

#*****
#*****
# detecting the blue cap

# Determine which pixels fall within the blue boundaries and then blur the binary image
blueMask = cv2.inRange(hsv, blueLower, blueUpper)
blueMask = cv2.erode(blueMask, kernel, iterations=2)
blueMask = cv2.morphologyEx(blueMask, cv2.MORPH_OPEN, kernel)
blueMask = cv2.dilate(blueMask, kernel, iterations=1)
#*****
#*****

if cv2.waitKey(10) & 0xFF == ord("w"):
    wait=0
elif cv2.waitKey(10) & 0xFF == ord("c"):
    wait=1
#*****
#*****

```

```

# Find contours in the image
(cnts, _) = cv2.findContours(blueMask.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
center = None
# Check to see if any contours were found
if len(cnts) > 0:
    # find the largest countour
    cnt = max(cnts, key = cv2.contourArea)
    x2,y2,w,h = cv2.boundingRect(cnt)
    # Get the moments to calculate the center of the contour (in this case a circle)
    M = cv2.moments(cnt)
    center = (int(M['m10'] / M['m00']), int(M['m01'] / M['m00']))
    # Draw the circle around the contour
    cv2.circle(frame, center, int(w/2), (0, 255, 255), 2)

#*****
#*****

# Draw lines of all the colors (Blue, Green, Red, Yellow and eraser)
if x1 == 0 and y1 == 0:
    x1,y1= x2,y2
else :
    if wait ==1 :
        canvas=cv2.line(canvas,(x1,y1),(x2,y2),colors1[colorIndex],1)
        cv2.line(paintWindow,(x1,y1),(x2,y2), colors[colorIndex], 1)
        x1,y1 =x2,y2
    else:
        x1,y1 =x2,y2
        frame = cv2.add(frame,canvas)

#*****
#*****

#checking boxes
if center[1] <= 65:
    if 40 <= center[0] <= 140: # Clear All

    paintWindow[:, :, :] = 255
    if u=='y':
        paintWindow=cv2.imread(address)
        canvas=None
        paintWindow = cv2.rectangle(paintWindow, (40,1), (140,65), (122,122,122), 2)
        paintWindow = cv2.rectangle(paintWindow, (160,1), (200,65), colors[0], -1)
        paintWindow = cv2.rectangle(paintWindow, (210,1), (250,65), colors[1], -1)
        paintWindow = cv2.rectangle(paintWindow, (260,1), (300,65), colors[2], -1)
        paintWindow = cv2.rectangle(paintWindow, (310,1), (350,65), colors[3], -1)
        paintWindow = cv2.rectangle(paintWindow, (370,1), (410,65), (122,122,122),2)
        paintWindow = cv2.line(paintWindow,(500,32),(530,32),(0,0,0),2)
        paintWindow = cv2.line(paintWindow,(550,32),(580,32),(0,0,0),4)
        paintWindow = cv2.line(paintWindow,(600,32),(630,32),(0,0,0),6)

```

```

cv2.putText(paintWindow, "CLEAR ALL", (49, 33), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0), 2, cv2.LINE_AA)
cv2.putText(paintWindow, "RUB", (372,33), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0), 2, cv2.LINE_AA)

```

```

elif 160 <= center[0] <= 200:
    colorIndex = 0 # Blue
elif 210 <= center[0] <= 250:
    colorIndex = 1 # Green
elif 260 <= center[0] <= 300:
    colorIndex = 2 # Red
elif 310 <= center[0] <= 350:
    colorIndex = 3 # Yellow
elif 370 <= center[0] <= 410:
    colorIndex = 4 # white

```

```

#thickness of line
elif 500<=center[0]<=530:
    l=4
elif 550<=center[0]<=580:
    l=8
elif 600<=center[0]<=630:
    l=12

```

```

#*****
*****

```

```

# Show the frame and the paintWindow image
cv2.imshow("Webcam frame", frame)
cv2.imshow("Paint", paintWindow)
# to save
if cv2.waitKey(1) & 0xFF == ord("s"):
    filename=input('enter the file name:')
    cv2.imwrite(filename,paintWindow)

```

```

# If the 'q' key is pressed, stop the loop
if cv2.waitKey(20) & 0xFF == ord("q"):
    break

```

```

#*****
*****

```

```

# Release the camera & destroy the windows.
camera.release()
cv2.destroyAllWindows()

```

```

printy('[mI]Thank you for using this program.I hope you had a wonderful experience.@')
answer=input('For more details enter \'y\'/else press any key to end the program: ')
if answer=='y':

```

```
printy('[o]This program is made as a part of IPD Training by :@ ')
printy('[o]Kriti Aggarwal\nCSE, 1 st year\nChandigarh College of Engineering & Technol
ogy @')
printy('[o]Date of submission:02-08-2020@')
printy('[o]Contact details :@')
printy('[o]email:co19335\@ccet.ac.in @ ')
printy('[o]linkden:https://www.linkedin.com/in/kriti-aggarwal-b372721a2@')

input()
```