Painting Transperacy

Using mouse as paint brush to paint transperacy

We will be creating PNG images with transperacy. Wherever we draw with brush we will see it being painted with black color as opencv window doenot support transperacy. But when we save the image and open it we will se that all the painted regions have become see through.

In [1]:

```
import cv2
import numpy as np

img = cv2.imread("images\\butter.png",-1)   # Reading the image with the fourth alpha ch
                                           # This just makes the task easier

drawing = False
cv2.namedWindow('image')

def nothing(x):
    pass

def mouse_call(event,x,y,flag,s):
    global drawing,rad

    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True

    if event == cv2.EVENT_MOUSEMOVE:
        if drawing:
            cv2.circle(img,(x,y),rad,(0,0,0,0),-1)

    if event == cv2.EVENT_LBUTTONUP:
        drawing = False

cv2.setMouseCallback('image',mouse_call)
cv2.createTrackbar('Radius','image',5,50,nothing)

while(1):
    cv2.imshow('image',img)
    k = cv2.waitKey(1) & 0xFF
    if k == 27:
        break

    if k == ord('s'):
        cv2.imwrite("images\\butter_copy.png",img)
    rad = cv2.getTrackbarPos('Radius','image')

cv2.destroyAllWindows()
```

If we do not have a imgae with fourth channel we can convert it to a four channel image using the below code

In [11]:

```
img = cv2.imread("images\\lena.jpg")
img2 = cv2.cvtColor(img,cv2.COLOR_BGR2BGRA) # convert 3 channels to 4 channels images
print(img2.shape)
```

```
(512, 512, 4)
```

Adding Images

cv2.add() method

Since the image is a numpy array we can uss the '+' operator to add the images. But this does not work porperly as the numpy addition is an overflow operation. So we can use the cv2.add() method which is a saturated addition operation.

In [1]:

```
img1 = cv2.imread("images\\scene.jpg")
img2 = cv2.imread("images\\opencv.png")

img1 = cv2.resize(img1,(180,222))

img_np = img1+img2
img_cv = cv2.add(img1,img2)

cv2.imshow("image1",img_np)
cv2.imshow("image2",img_cv)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-2145367d36e3> in <module>
----> 1 img1 = cv2.imread("images\\scene.jpg")
      2 img2 = cv2.imread("images\\opencv.png")
      3 
      4 img1 = cv2.resize(img1,(180,222))
      5 

NameError: name 'cv2' is not defined
```

Merging images using `cv2.addWeighted` method

`cv2.addWeighted(image_1, alpha, image_2, beta, Gamma)`

- image_1 : first input array.
- alpha : weight of the first array elements.
- image_2 : second input array of the same size and channel number as src1.
- beta : weight of the second array elements.
- gamma : scalar added to each sum.

In [1]:

```
import cv2
img1 = cv2.imread("images\\scene.jpg")
```

```
img2 = cv2.imread("images\\opencv.png")

img1 = cv2.resize(img1,(180,222))

dst = cv2.addWeighted(img1,0.9,img2,0.1,0)

cv2.imshow("image1",dst)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Reading directory

In [14]:

```
# This code will come in handy to read all the images in a folder.

import os
x = os.scandir('images')
for i in x:
    print(i.name)
```

```
butter.png
butter_copy.png
chess.png
lena.jpg
opencv.png
scene.jpg
```

HOMEWORK

Slideshow of images

Create a slide show off all images in a folder using the `addWeighted() method`

1. Using the above concept of reading directory files create a slideshow of images.

In [ ]:

```
```