1. 콜백함수란 무엇인가?

|  |
| --- |
| 특정한 함수를 다른 함수에서 호출하는 것 |

2. 윈도우를 지정하는 cv2.namedWindow() 함수의 두 번째 인수(flags)에 대한 옵션은 여러가지가 있다. 그 중에서 cv2.WINDOW\_NORMAL와 cv2.WINDOW\_AUTOSIZE 간의 차이를 설명하시오.

|  |
| --- |
| cv2.WINDOW\_NORMAL : 윈도우 크기 재조정 가능. 이미지가 윈도우 크기에 맞춰진다.  cv2.WINDOW\_AUTOSIZE: 표시된 행렬의 크기에 맞춰 자동 조정. 이미지의 크기는 일정하다. |

3. 타원을 그리는 cv2.ellipse() 함수의 인수를 자세히 설명하시오.

|  |
| --- |
| 이미지, 중심좌표, (x축반지름, y축반지름), 타원의 각도, 호의 시작각도, 호의 종료각도, 색, 선의 두께, 선의 현태, 비트 시프트 |

4. OpenCV이 제공하는 마우스 이벤트와 트랙바 이벤트를 제어할 콜백 함수를 시스템에 등록하는 함수는 각각 무엇이며, 인수가 어떻게 구성되었는지 자세히 설명하시오.

|  |
| --- |
| cv2.setMouseCallback(windowNamne, onMouse, param=None)  -windowName : 윈도우 이름  -onMouse : 콜백함수이름  -param : 추가적인 사용자 정의 인수  cv2.createTrackbar(trackbarname, winname, value, count, onChange)  -trackbarname : 트랙바 이름  -winname : 윈도우 이름  -count : 트랙바의 최댓값  -onChange : 슬라이더의 값이 변경될 때 호출되는 콜백함수 |

5. 다음 예시 코드의 실행 결과를 설명하시오.

|  |
| --- |
| 4행 : 300x400의 행렬 생성  5행 : 모든 화소 값을 100으로 초기화  8행 : 사용자가 크기를 조절이 가능한 윈도우창 생성  9행 : 윈도우창을 (100, 200)으로 이동  10행 : 윈도우창을 띠운다  11행 : 키가 입력되기 전까지 대기  12행 : 모든 윈도우창 제거 |

|  |
| --- |
| 3행 : 화소값이 0인 400x600의 3채널 행렬 생성  4행 : 모든 화소 값을 흰색으로 바꿈  7행 : image에 pt1에서 pt2까지 초록색, 선두께가 5인 선을 생성  8행 : 시작좌표(pt1), 종료좌표(300,400), 빨간색, 속이 찬, 4방향 연결선, 오른쪽으로 1만큼 비트시프트연산을 한 사각형 생성  10행 : 제목이 Line & Rectangle인 윈도우 생성  11행 : 키 입력전까지 대기  12행 : 모든 윈도우창 제거 |

6. 300행, 400열의 행렬을 회색 바탕색(100)으로 생성해서 500행, 600열의 윈도우에 표시하시오.

|  |
| --- |
| import numpy as np, cv2  img=np.zeros((300,400), np.uint8) img.fill(100)  cv2.imshow('title', img) cv2.resizeWindow('title', 500, 600) cv2.waitKey(0) cv2.destroyAllWindows() |

7. 다음 예시 코드는 컴파일 혹은 런타임 에러가 발생한다. 에러가 발생하는 부분을 수정하고 실행 결과를 적으시오.

(1)

|  |
| --- |
| import numpy as np import cv2  image = np.zeros((300, 400, 3), np.uint8) image[:] = (255, 255, 255)  pt1, pt2 = (50, 130), (200, 300)  cv2.line(image, pt1, (100, 200), (0,0,0)) cv2.line(image, pt2, (100, 100), (0,0,0)) cv2.rectangle(image, pt1, pt2, (255, 0, 255)) cv2.rectangle(image, pt1, pt2, (0, 0, 255))  title = "Line & Rectangle" cv2.namedWindow(title) cv2.imshow(title, image) cv2.waitKey(0) cv2.destroyAllWindows() |
| 7행 : line함수의 color인자가 없음  8행 : pt2가 이차원이므로 도착지점인 (100,100,100)또한 이차원이어야 한다. |

(2)

|  |
| --- |
| import numpy as np import cv2  def onMouse(event, x, y, flags, param):  global title  pt = (x, y)   if event == cv2.EVENT\_LBUTTONDOWN:  cv2.circle(image, pt, 5, 100, 1)  cv2.imshow(title, image)  elif event == cv2.EVENT\_RBUTTONDOWN:  cv2.rectangle(image, pt+(30,30), 100, 2)  cv2.imshow(title, image)  image = np.ones((300, 300), np.uint8) \* 255 title = "Draw Event"  cv2.namedWindow(title) cv2.imshow(title, image) cv2.setMouseCallback(title, onMouse) cv2.waitKey(0) cv2.destroyAllWindows() |
| 5행 : pt를 선언하지 않았음  11행 : pt+(30,30)와 pt, (30, 30)중 하나를 사용해야 함. |

8. 200행, 300열의 행렬 2개를 만들어서 다음과 같이 배치하시오.

|  |
| --- |
| import numpy as np,cv2  img=np.zeros((200,300), np.uint8) title1='win mode1' title2='win mode2'  cv2.namedWindow(title1, cv2.WINDOW\_AUTOSIZE) cv2.namedWindow(title2) cv2.moveWindow(title1, 0, 0) cv2.moveWindow(title2, 300, 200)  cv2.imshow(title1, img) cv2.imshow(title2, img) cv2.waitKey(0) cv2.destroyAllWindows() |

9. 600행, 400열의 윈도우를 만들고, 영상 안의 (100, 100) 좌표에 200x300 크기의 빨간 색 사각형을 그리시오.

|  |
| --- |
| import numpy as np, cv2  img = np.zeros((600, 400, 3), np.uint8)  title='rectangle' cv2.namedWindow(title, cv2.WINDOW\_AUTOSIZE) cv2.rectangle(img, (100, 100, 200, 300), (0, 0, 255))  cv2.imshow(title, img) cv2.waitKey(0) cv2.destroyAllWindows() |

10. 다음의 마우스 이벤트 제어 프로그램을 작성하시오.

(1) 마우스 오른쪽 버튼 클릭 시 원(클릭 좌표에서 반지름 20화소)을 그린다.  
(2) 마우스 왼쪽 버튼 클릭 시 사각형(크기 30x30)을 그린다.

|  |
| --- |
| import numpy as np, cv2  def onMouse(event, x, y, flags, param):  if event == cv2.EVENT\_LBUTTONDOWN:  cv2.circle(img, (x, y), 20, (255, 0, 0))  elif event == cv2.EVENT\_RBUTTONDOWN:  cv2.rectangle(img, (x, y, 30, 30), (0, 0, 255), 1)  cv2.imshow(title, img)  img = np.zeros((1000,1000, 3), np.uint8) img[:]=(255)  title='mouse event' cv2.namedWindow(title, cv2.WINDOW\_AUTOSIZE) cv2.setMouseCallback(title, onMouse) cv2.waitKey(0) cv2.destroyAllWindows() |

11. 10번 연습문제에서 다음을 추가하여 프로그램을 작성하시오.

(1) 트랙바를 추가해서 선의 굵기를 1~10픽셀로 조절한다.  
(2) 트랙바를 추가해서 원의 반지름을 1~50픽셀로 조절한다.

|  |
| --- |
| import numpy as np, cv2  def onChange1(value):  global thick  thick=value  def onChange2(value):  global Radius  Radius=value  def onMouse(event, x, y, flags, param):  if event == cv2.EVENT\_LBUTTONDOWN:  print(x, y)  cv2.circle(img, (x, y), Radius, (255, 0, 0), thick)  elif event == cv2.EVENT\_RBUTTONDOWN:  print("2")  cv2.rectangle(img, (x, y, 30, 30), (0, 0, 255), 1)  cv2.imshow(title, img)  img = np.zeros((1000,1000, 3), np.uint8) img[:]=(255) thick=1 Radius=20  title='mouse event' cv2.namedWindow(title, cv2.WINDOW\_AUTOSIZE) cv2.createTrackbar('thick', title, thick, 10, onChange1) cv2.createTrackbar('Radius', title, Radius, 50, onChange2) cv2.setMouseCallback(title, onMouse) cv2.waitKey(0) cv2.destroyAllWindows() |

12. 예제\_4.2.3인 05.event\_trackbar.py에서 화살표 키로 트랙바를 이동하는 코드를 추가하시오.

|  |
| --- |
| import numpy as np import cv2  switch\_case = {  2424832:"왼쪽 화살표",  2555904:"오른쪽 화살표" }  def onChange(value):  global image, title  image[:] = value  cv2.imshow(title, image)  brigthness=0 image = np.zeros((300, 500), np.uint8) title = 'Trackbar Event' cv2.imshow(title, image) cv2.createTrackbar('brightness', title, image[0][0], 255, onChange)  while True:  key = cv2.waitKeyEx(100)  if key == 2424832:  if(brigthness-1<0):  print('brigthness can not be minus')  else:  brigthness=brigthness-1  cv2.setTrackbarPos('brightness', title, brigthness)  elif key == 2555904:  if(brigthness+1>256):  print("brightness can not be over 255")  else:  brigthness=brigthness+1  cv2.setTrackbarPos('brightness', title, brigthness)   try:  result = switch\_case[key]  print(result)  except KeyError:  result = -1  cv2.destroyAllWindows() |

13. 컬러 영상파일을 적재한 후의 행렬을 윈도우에 명암도 영상으로 표시하고, “test.jpg”와 “test.png” 파일로 각각 저장하시오. 이때 영상파일을 가장 좋은 화질로 압축하시오.

|  |
| --- |
| import numpy as np, cv2  img=cv2.imread("C:/computervision/chap04/image/img.png", cv2.IMREAD\_GRAYSCALE)  cv2.imshow('title',img) img1=(cv2.IMWRITE\_JPEG\_QUALITY, 100) img2=(cv2.IMWRITE\_PNG\_COMPRESSION, 9)  cv2.imwrite("C:/computervision/chap04/image/img1.jpg", img1) cv2.imwrite("C:/computervision/chap04/image/img2.png", img2)  cv2.waitKey(0) cv2.destroyAllWindows() |

14. 심화예제\_4.3.5인 11.event\_draw.py를 수정해서 마우스 중간버튼을 클릭하여 타원을 그리세요.

|  |
| --- |
| import numpy as np, cv2  def onMouse(event, x, y, flags, param):  global title, pt   if event == cv2.EVENT\_LBUTTONDOWN:  if pt[0] < 0: pt = (x,y)  else:  cv2.rectangle(image, pt, (x, y), (255, 0, 0), 2)  cv2.imshow(title, image)  pt = (-1, -1)  elif event ==cv2.EVENT\_MBUTTONDOWN:  if pt[0] < 0: pt=(x,y)  else:  dx, dy = pt[0]-x, pt[1]-y  radius=int(np.sqrt(dx\*dx+dy\*dy))  cv2.circle(image, pt, radius, (0, 0, 255), 2)  cv2.imshow(title, image)  pt = (-1, -1)  image = np.full((300,500,3),(255,255,255), np.uint8)  pt=(-1,-1) title="DRAW EVENT" cv2.imshow(title, image) cv2.setMouseCallback(title, onMouse) cv2.waitKey(0) |

15. 심화예제\_4.5.2인 18.set\_camera\_attr.py를 수정해서 트랙바로 카메라 영상의 밝기와 대비 변경할 수 있도록 수정하시오.

|  |
| --- |
| import cv2 from Common.utils2 import put\_string  def bright\_bar(value):  global capture  capture.set(cv2.CAP\_PROP\_BRIGHTNESS,value)  def contrast\_bar(value):  global capture  capture.set(cv2.CAP\_PROP\_CONTRAST, value)  capture = cv2.VideoCapture(0) if capture.isOpened() == False: raise Exception("카메라 연결 안됨")  capture.set(cv2.CAP\_PROP\_FRAME\_WIDTH,400) capture.set(cv2.CAP\_PROP\_FRAME\_HEIGHT,300) capture.set(cv2.CAP\_PROP\_BRIGHTNESS,0) capture.set(cv2.CAP\_PROP\_CONTRAST, 0)  title = "Change Camera Properties" cv2.namedWindow(title) cv2.createTrackbar('bright',title, 0, 255, bright\_bar) cv2.createTrackbar('contrast', title, 0, 100, contrast\_bar)  while True:  ret, frame=capture.read()  if not ret:break  if cv2.waitKey(30) >=0: break  bright = int(cv2.getTrackbarPos('bright', title))  contrast = int(cv2.getTrackbarPos('contrast', title))  put\_string(frame, 'bright : ', (10, 240), bright)  put\_string(frame, 'contrast : ', (10, 270), contrast)  cv2.imshow(title, frame)  capture.release() |

16. PC 카메라를 통해서 받아온 프레임에 대해 다음의 영상처리를 수행하고, 결과 영상을 윈도우에 표시하는 프로그램을 작성하시오.

|  |
| --- |
| 1) (200, 100)좌표,에서 100x200크기의 관심 영역 지정  2) 관심영역에서 녹생성분을 50만큼 증가  3) 관심영역의 테두리를 두께 3의 빨간색으로 표시 |

|  |
| --- |
| import numpy as np, cv2  capture = cv2.VideoCapture(0) if capture.isOpened()==False: raise Exception("카메라 연결 안됨")  capture.set(cv2.CAP\_PROP\_FRAME\_WIDTH, 1000) capture.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, 1000)  title = '1' cv2.namedWindow(title)  while True:  ret, frame=capture.read()  if not ret:break  if cv2.waitKey(30) >=0: break   cv2.rectangle(frame, (200, 100), (300,300),(0, 0, 255), 3, cv2.LINE\_4)  b, g, r = cv2.split(frame)  g[100:300,200:300]+=50  cv2.merge((b,g,r),frame)  cv2.imshow(title, frame)  capture.release() |

17. PC 카메라를 통해서 받아온 프레임을 좌우로 뒤집어서 “flip\_test.avi” 이름의 동영상 파일로 저장하는 프로그램을 작성하시오.

|  |
| --- |
| import numpy as np, cv2  capture = cv2.VideoCapture(0) if capture.isOpened()==False: raise Exception("카메라 연결 안됨")  title = '1' cv2.namedWindow(title)  while True:  ret, frame=capture.read()  if not ret:break  if cv2.waitKey(30) >=0: break   flip\_img=cv2.flip(frame, 1)  cv2.imshow(title, flip\_img)  capture.release() |

18. 다음과 같이 태극 문양을 그리는 프로그램을 작성하시오.

|  |
| --- |
| import numpy as np, cv2  red, green, blue=(0,0,255),(0,255,0),(255,0,0)  img=np.zeros((400,600,3), np.uint8) img[:]=(255,255,255)  title='1' cv2.namedWindow(title, cv2.WINDOW\_AUTOSIZE) cv2.ellipse(img, (300,200), (100, 100), 0, 180, 360, red, -1) cv2.ellipse(img, (300,200), (100, 100), 0, 0, 180, blue, -1) cv2.ellipse(img, (350,200), (50, 50), 0, 180, 360, blue, -1) cv2.ellipse(img, (250,200), (50, 50), 0, 0, 180, red, -1)  cv2.imshow(title, img) cv2.waitKey(0) cv2.destroyAllWindows() |