

CH05. OpenCV: 마우스로 도형 그리기

5. 마우스로 도형 그리기

- 5.1 테스트 환경

순 번	제 목	설치 버전
1	운영 체 제	Windows 10 64bit
2	프로그래밍언어	Python3.6.5

5. 마우스로 도형 그리기

- 5.2 [05-01 example.py] – 마우스로 원 그리기
 - 다음은 05-01 example.py 예제의 code이다.

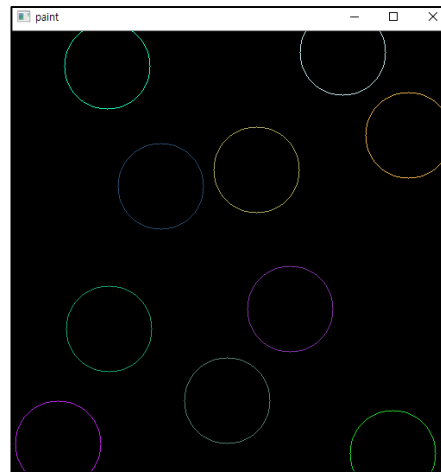
[그림] 05-01 example.py의 code (1/3)

[그림] 05-01 example.py의 code (2/3)

5. 마우스로 도형 그리기

▪ 5.2 [05-01 example.py] – 마우스로 원 그리기

- 다음은 05-01 example.py 예제의 code와 실행 결과로 나오는 화면이다. 뒷장 부터 05-01 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



[그림] 05-01 example.py의 실행 결과

5. 마우스로 도형 그리기

▪ 5.2 [05-01 example.py] – 마우스로 원 그리기

>> LINE 3) from random import shuffle

- 무작위 색상 값을 추출하기 위해 shuffle 모듈을 impor한다.

>> LINE 5) b = [i for i in range(256)]

>> LINE 6) g = [i for i in range(256)]

>> LINE 7) r = [i for i in range(256)]

- BGR 색상값을 위해 사용될 0~255를 멤버로 하는 리스트를 생성한다.

5. 마우스로 도형 그리기

▪ 5.2 [05-01 example.py] – 마우스로 원 그리기

>> LINE 5 ~ 12)

```
def onMouse(event, x, y, flags, param):
```

```
    if event == cv2.EVENT_LBUTTONDOWNBLCLK:
```

```
        shuffle(b), shuffle(g), shuffle(r)
```

```
        cv2.circle(param, (x,y), 50, (b[0], g[0], r[0], -1))
```

- 마우스 이벤트를 처리할 콜백 함수 이다. 이 함수는 cv2.setMouseCallBack() 함수의 인자로 지정되어 호출된다.

(1) event : 마우스 이벤트

(2) x,y : 마우스 이벤트가 일어난 위치

(3) params: cv2.setMouseCallback() 함수에서 전달받은 사용자 데이터이다. 우리의 코드에 서는 img가 전달된다.

5. 마우스로 도형 그리기

▪ 5.2 [05-01 example.py] – 마우스로 원 그리기

```
>> LINE 16) cv2.namedWindow('paint')
```

```
>> LINE 17) cv2.setMouseCallback('paint', onMouse, param=img)
```

- 'paint'로 이름 붙여진 윈도우상에서 발생하는 마우스 이벤트를 처리하기 위한 콜백 함수를 설정한다. 콜백 함수는 onMouse()이고, 콜백함수로 전달할 사용자 데이터는 img이다.

```
>> LINE 5) b = [i for i in range(256)]
```

```
>> LINE 6) g = [i for i in range(256)]
```

```
>> LINE 7) r = [i for i in range(256)]
```

- BGR 색상값을 위해 사용될 0~255를 멤버로 하는 리스트를 생성한다.

5. 마우스로 도형 그리기

- 5.3 [05-02 example.py] – 마우스로 도형 그리기
 - 다음은 05-02 example.py 예제의 code이다.

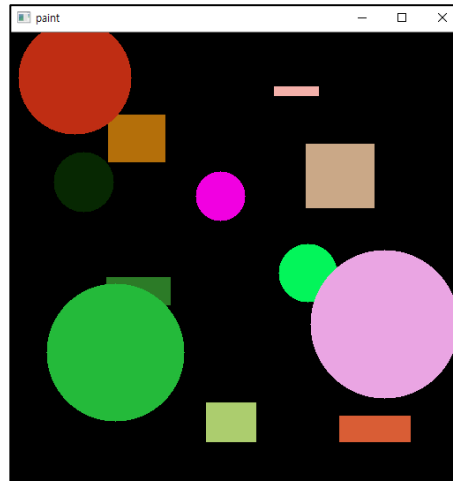
[그림] 05-01 example.py의 code (1/3)

[그림] 05-01 example.py의 code (2/3)

5. 마우스로 도형 그리기

▪ 5.3 [05-02 example.py] – 마우스로 원과 사각형 그리기

- 다음은 05-02 example.py 예제의 code와 실행 결과로 나오는 화면이다. 뒷장 부터 05-02 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



[그림] 05-02 example.py의 실행 결과

5. 마우스로 도형 그리기

▪ 5.3 [05-02 example.py] – 마우스로 원과 사각형 그리기

- 512 x 512 크기의 검정색 그림판에 직사각형 원을 그릴 수 있다.
- 마우스 왼쪽 버튼을 누르고 움직이면 직사각형이 그려지고 누르고 있던 마우스 버튼을 떼면 직사각형이 완성됨.
- 마우스 왼쪽 버튼을 누르고 움직이면 원이 그려지고 누르고 있던 마우스 버튼을 떼면 원이 완성된다. 마우스 왼쪽 버튼을 누른 지점이 원의 중심이 된다.
- 직사각형, 원의 색상은 무작위로 선택된다.
- 키보드 'm' 버튼으로 직사각형 그리기, 원 그리기가 토글된다.
- ESC 키로 프로그램을 종료한다.

5. 마우스로 도형 그리기

▪ 5.3 [05-02 example.py] – 마우스로 원과 사각형 그리기

>> LINE 4) `import math`

- 마우스 왼쪽 버튼을 누른 지점과 마우스 왼쪽 버튼을 떼는 지점간 거리를 재는데 필요한 `sqrt()` 함수를 사용하기 위해 `math` 패키지를 `import` 한다.

>> LINE 6) `mode, drawing = True, False`

>> LINE 7) `ix, iy = -1, -1`

- `mode` 변수는 직사각형, 원 그리기를 토글하기 위한 플래그로 사용될 것이다. `drawing` 변수는 마우스 왼쪽 버튼을 누르고 움직이면 도형을 그리고, 마우스 왼쪽 버튼을 떼고 움직일 때 도형이 안 그려지도록 하는 플래그이다. `ix, iy`는 마우스 왼쪽 버튼을 누른 지점으로 활용될 변수이다.

5. 마우스로 도형 그리기

- 5.3 [05-02 example.py] – 마우스로 원과 사각형 그리기

>> LINE 13) global ix, iy, drawing, mode, B, G, R

- ix, iy, drawing, mode, B, G, R은 전역 변수로 선언된 것을 사용한다는 의미이다.

```

import numpy as np
import cv2
from random import shuffle
import math
mode, drawing = True, False
ix, iy = -1, -1
B = [i for i in range(256)]
G = [i for i in range(256)]
R = [i for i in range(256)]

def onMouse(event, x, y, flags, param):
    global ix, iy, drawing, mode, B, G, R
    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True
        ix, iy = x, y
        shuffle(B), shuffle(G), shuffle(R)
    elif event == cv2.EVENT_MOUSEMOVE:
        if drawing:
            if mode:
                cv2.rectangle(param, (ix, iy), (x, y), (B[0], G[0], R[0]), -1)
            else:
                r = (ix-x)**2 + (iy-y)**2
                r = int(math.sqrt(r))
                cv2.circle(param, (ix, iy), r, (B[0], G[0], R[0]), -1)
    elif event == cv2.EVENT_LBUTTONUP:
        drawing = False
        if mode:
            cv2.rectangle(param, (ix, iy), (x, y), (B[0], G[0], R[0]), -1)
        else:
            r = (ix-x)**2 + (iy-y)**2
            r = int(math.sqrt(r))
            cv2.circle(param, (ix, iy), r, (B[0], G[0], R[0]), -1)

```

```
def mouseBrush():
```

```
    global mode
```

```
    img = np.zeros((512, 512, 3), np.uint8)
```

```
    cv2.namedWindow('paint')
```

```
    cv2.setMouseCallback('paint', onMouse, param=img)
```

```
    while True:
```

```
        cv2.imshow('paint', img)
```

```
        k = cv2.waitKey(1) & 0xFF
```

```
        if k == 27:
```

```
            break
```

```
        elif k == ord('n'):
```

```
            mode = not mode
```

```
    cv2.destroyAllWindows()
```

```
mouseBrush()
```

```
import numpy as np
import cv2
from random import shuffle
b = [i for i in range(256)]
g = [i for i in range(256)]
r = [i for i in range(256)]
def onMouse(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        shuffle(b), shuffle(g), shuffle(r)
        cv2.circle(param, (x,y), 50, (b[0], g[0], r[0]), -1)
def mouseBrush():
    img = np.zeros((512,512,3), np.uint8)
    cv2.namedWindow('paint')
    cv2.setMouseCallback('paint', onMouse, param=img)

    while True:
        cv2.imshow('paint', img)
        k = cv2.waitKey(1) & 0xFF

        if k == 27:
            break

    cv2.destroyAllWindows()
mouseBrush()
```