User Interface

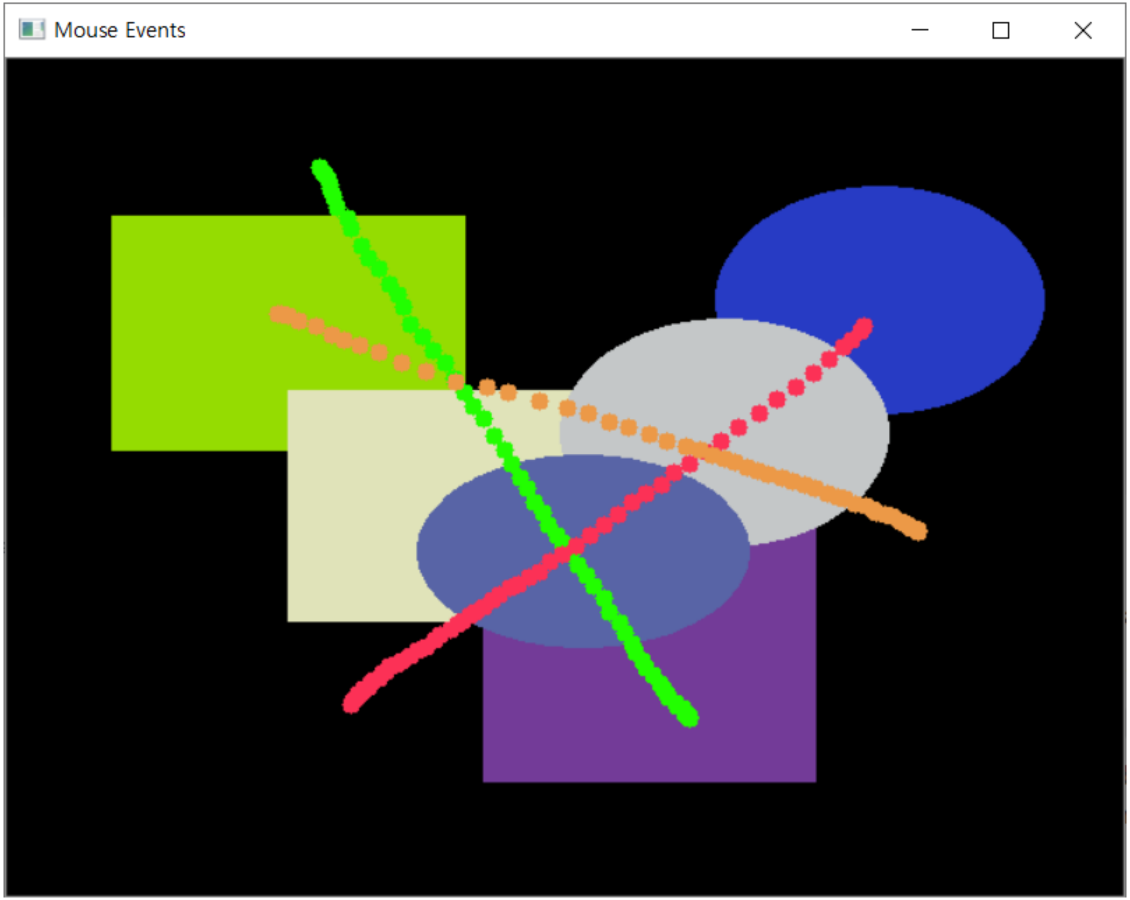
학습목표

학습목표

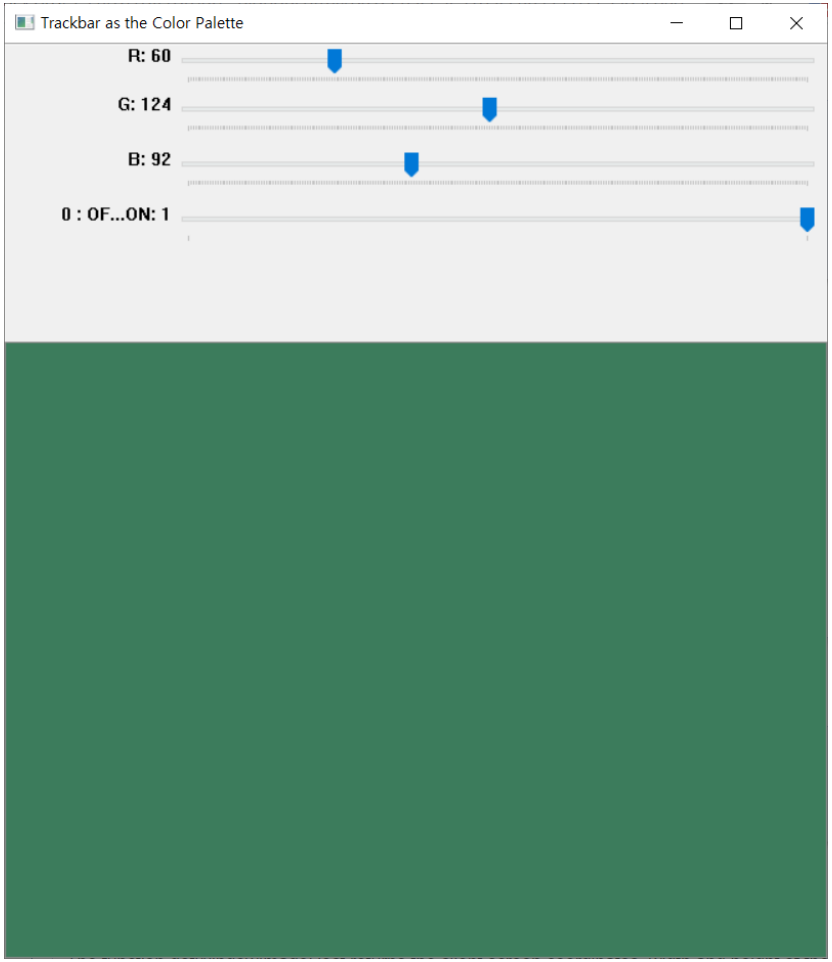
1. OpenCV에서 keyboard 입력 방법을 이해한다.
2. OpenCV에서 mouse 입력 방법을 이해한다.
3. OpenCV에서 trackbar 사용 방법을 이해한다.

- 왜? 사용자 입력이 필요한 경우 사용하기 위해서!

학습목표



학습목표



Trackbar as the Color Palette

R: 60

G: 124

B: 92

0 : OF...ON: 1

Keyboard Inputs

`cv.waitkey()`[1]

```
retval = cv.waitKey([, delay])
```

- Keyboard 입력을 기다린다.
  - `delay`: 대기 시간(ms), 0보다 작거나 같으면 무한히 대기
  - `retval`: 입력된 key 값, 대기 시간동안 입력이 없으면 -1

[1] https://docs.opencv.org/4.4.0/d7/dfc/group__highgui.html#ga5628525ad33f52eab17feebcfba38bd7

Example: Flip Images Around x- and y- Axes

```python
import cv2

# Load an image
img = cv2.imread('messi5.jpg')

# Infinite loop
while True:

    # Display the image in a window
    cv2.imshow('Lionel Messi', img)

    # Wait for a key to be pressed
    key = cv2.waitKey(1)

    # Flip the image
    if key == ord('x'):
        img = cv2.flip(img, 0)
    elif key == ord('y'):
        img = cv2.flip(img, 1)
    elif key == 27:
        break
```

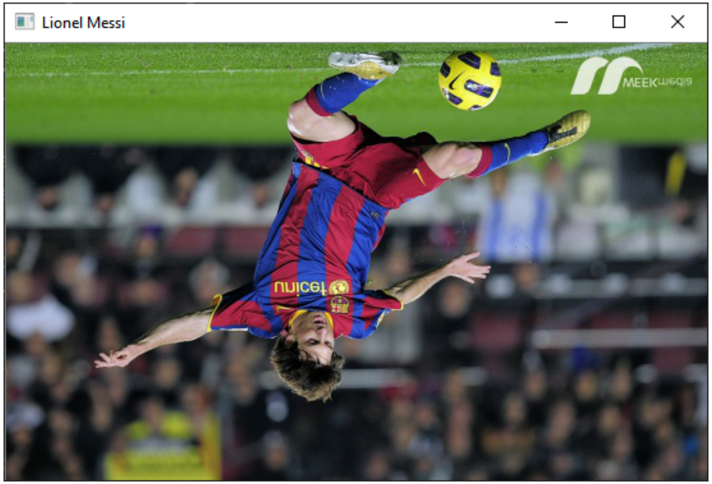
`cv.flip()`[1]

```
dst = cv.flip(src, flipCode[, dst])
```

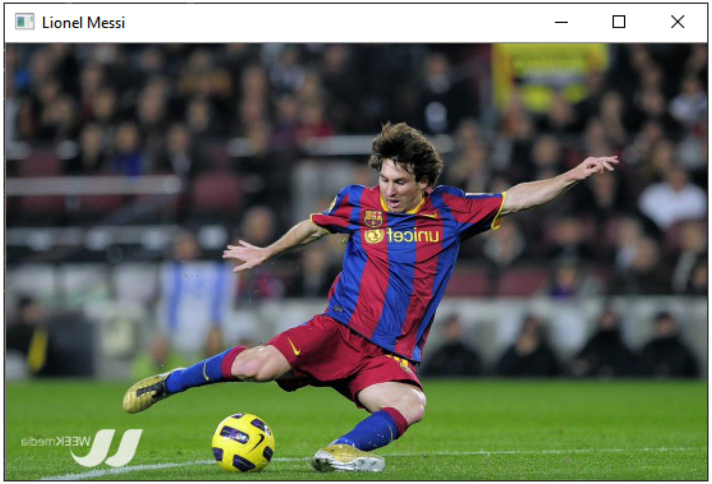
- Image를 수직 혹은 수평으로 flip 시킨다.
    - `src`: Input array
    - `dst`: Output array of the same size and type as src
    - `flipCode`: 0: x축 대칭, 양수(1): y축 대칭, 음수(-1): 원점대칭

[1] https://docs.opencv.org/4.4.0/d2/de8/group__core__array.html#gaca7be533e3dac7feb70fc60635adf441

Result: Flip Images Around x- and y- Axes



`flipCode` = 0: flip the x-axis



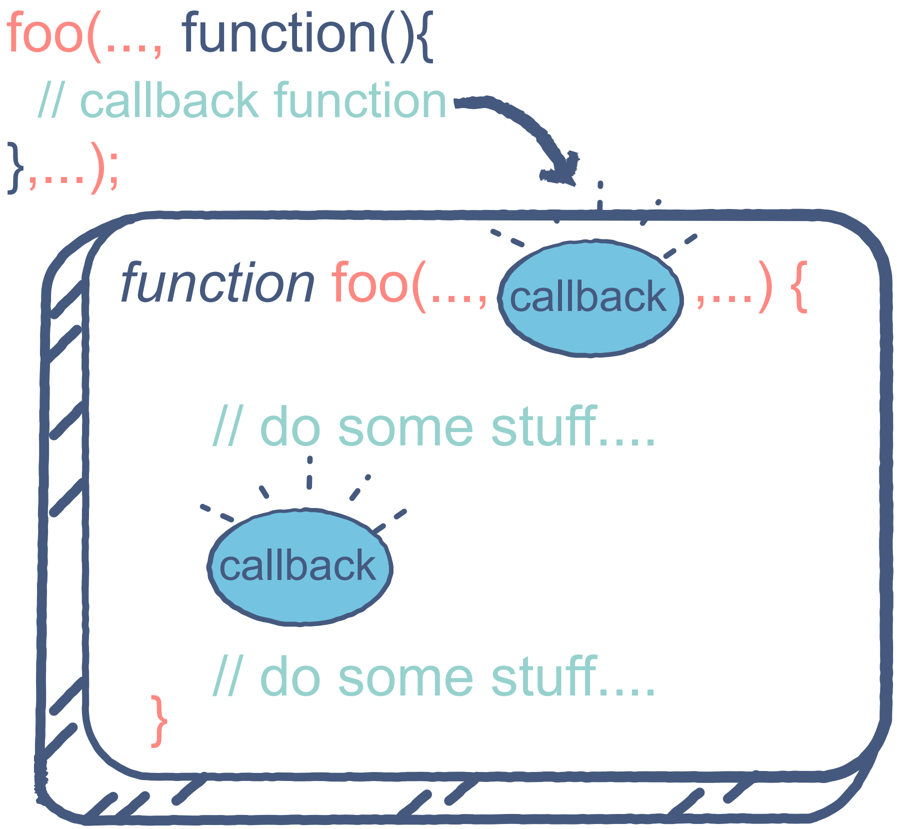
`flipCode` = 1: flip the y-axis

Mouse Events

Callback Functions

https://medium.com/@tgunix/dart-create-callback-via-typedef-2b913fbe3bac

Callback Functions



```
foo(..., function(){
  // callback function
},...);
```

```
function foo(..., callback ,...) {

    // do some stuff....

    callback

    // do some stuff....
}
```

https://www.educative.io/edpresso/what-are-callbacks-in-javascript

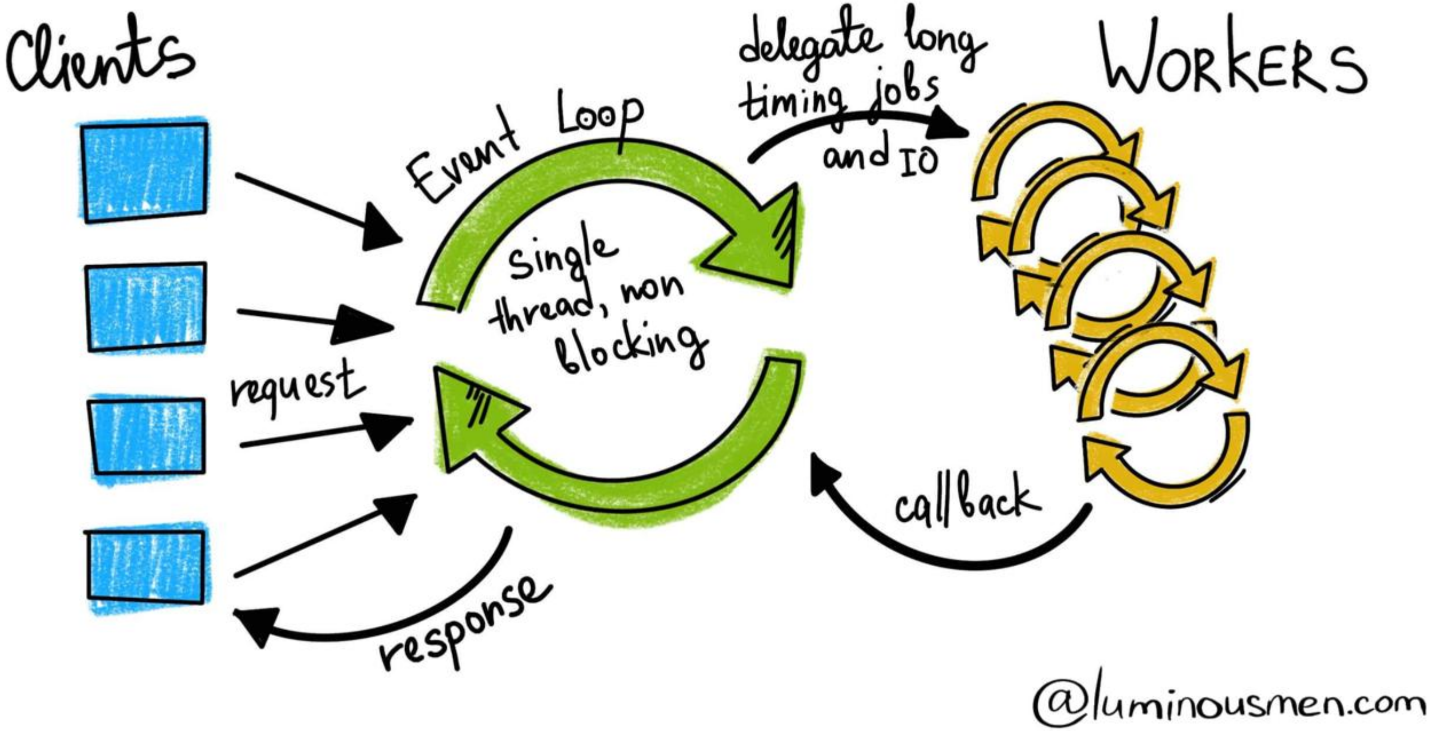
Asynchronous vs. Synchronous

1. 손님이 레스토랑에 들어와서 테이블에 앉는다.
2. 웨이터가 테이블로 와서 주문을 받는다.
3. 웨이터가 주문을 주방의 요리사에게 보낸다.
4. 웨이터는 다음 테이블로 가서 또다른 주문을 받는다.
5. 주방에서 음식이 완성되면 웨이터에게 알려준다.
6. 웨이터가 음식을 주문한 손님 테이블로 서빙한다.

1. 손님이 레스토랑에 들어와서 테이블에 앉는다.
2. 웨이터가 테이블로 와서 주문을 받는다.
3. 웨이터가 주문을 주방의 요리사에게 보낸다.
4. 워이터가 주방에서 음식이 완성될 때까지 기다린다.
5. 주방에서 음식이 완료되면 웨이터에게 알려준다.
6. 웨이터가 음식을 주문한 손님 테이블로 서빙한다.
7. 웨이터는 다음 테이블로 가서 또다른 주문을 받는다.

https://www.loginradius.com/engineering/blog/callback-vs-promises-vs-async-await/

Asynchronous Callback Functions



`cv.setMouseCallback()`[1]

```
None = cv.setMouseCallback(winname, mouse_callback[, userdata])
```

- Mouse callback 함수를 등록한다.
  - `winname`: Name of the window
  - `mouse_callback`: Callback funcion
  - `userdata`: Optional parameter passed to the callback

[1] https://docs.opencv.org/4.4.0/d7/dfc/group__highgui.html#ga89e7806b0a616f6f1d502bd8c183ad3e

`mouse_callback()`[1]

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

- Mouse callback 함수를 정의한다.
  - `event`: Mouse event type
  - `x`: X-coordinate of the mouse event
  - `y`: Y-coordinate of the mouse event
  - `flags`: Mouse event flags
  - `param`: Optional parameter passed to the callback

[1] https://docs.opencv.org/4.4.0/d7/dfc/group__highgui.html#gab7aed186e151d5222ef97192912127a4

`mouse_callback()`[1]

```python
def mouse_callback(event, x, y, flags, param):
```

event[2]	event[2]	flags[3]
`cv.EVENT_LBUTTONDOWN`	`cv.EVENT_LBUTTONUP`	`cv.EVENT_FLAG_LBUTTON`
`cv.EVENT_RBUTTONDOWN`	`cv.EVENT_RBUTTONUP`	`cv.EVENT_FLAG_RBUTTON`
`cv.EVENT_MBUTTONDOWN`	`cv.EVENT_MBUTTONUP`	`cv.EVENT_FLAG_MBUTTON`
`cv.EVENT_LBUTTONDBLCLK`	`cv.EVENT_MOUSEMOVE`	`cv.EVENT_FLAG_CTRLKEY`
`cv.EVENT_RBUTTONDBLCLK`	`cv.EVENT_MOUSEWHEEL`	`cv.EVENT_FLAG_SHIFTKEY`
`cv.EVENT_MBUTTONDBLCLK`	`cv.EVENT_MOUSEHWHEEL`	`cv.EVENT_FLAG_ALTKEY`

[1] https://docs.opencv.org/4.4.0/d7/dfc/group__highgui.html#gab7aed186e151d5222ef97192912127a4

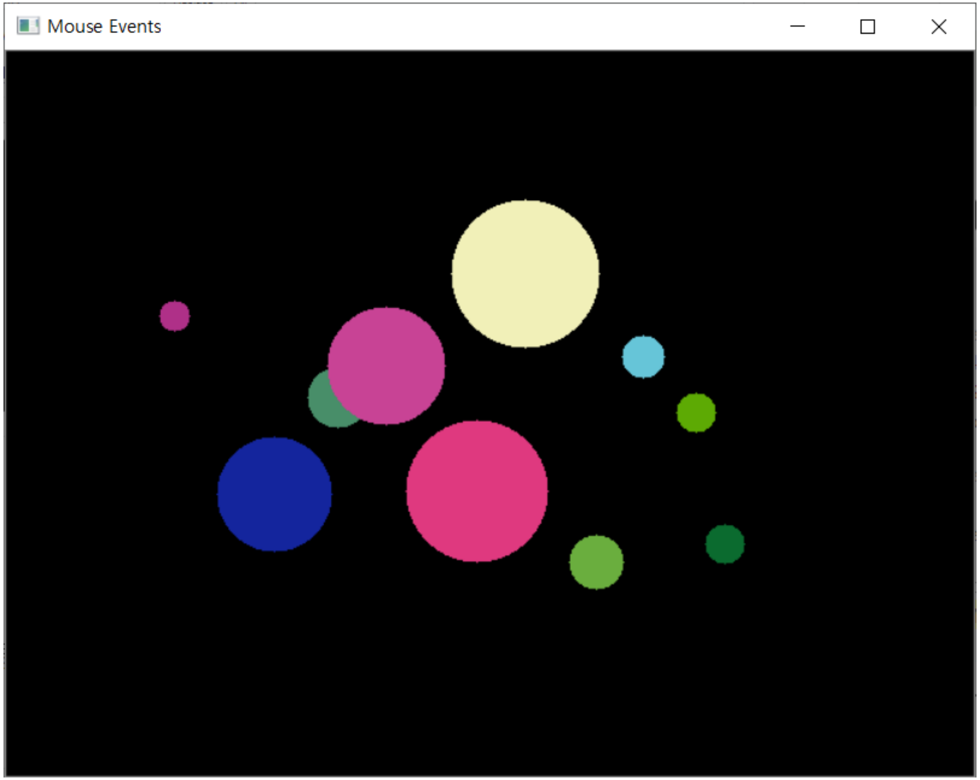
[2] https://docs.opencv.org/4.4.0/d7/dfc/group__highgui.html#ga927593befdddc7e7013602bca9b079b0

[3] https://docs.opencv.org/4.4.0/d7/dfc/group__highgui.html#gaab4dc057947f70058c80626c9f1c25ce

Example 1: Double-Click to Draw a Circle

```
import cv2
import numpy as np
import random

# mouse callback function
def mouse_callback(event, x, y, flags, param):
    # Left button double clicked
    if event == cv2.EVENT_LBUTTONDBLCLK:
        # Pick a random radius
        radius = random.randrange(10, 50)

        # Pick a random color
        color = (random.randrange(256), random.randrange(256), 
random.randrange(256))

        # Draw a circle
        cv2.circle(img_color, (x,y), radius, color,-1)

# Create a black image
rows = 480
cols = 640
```

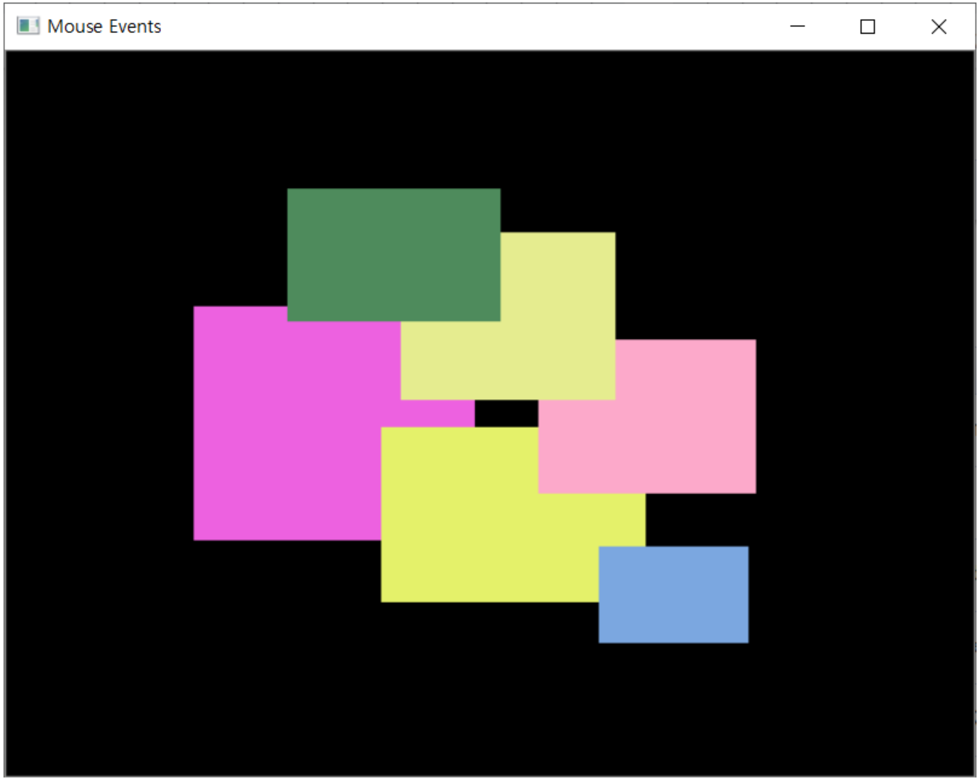
Example 1: Double-Click to Draw a Circle



Example 2: Draw Shapes

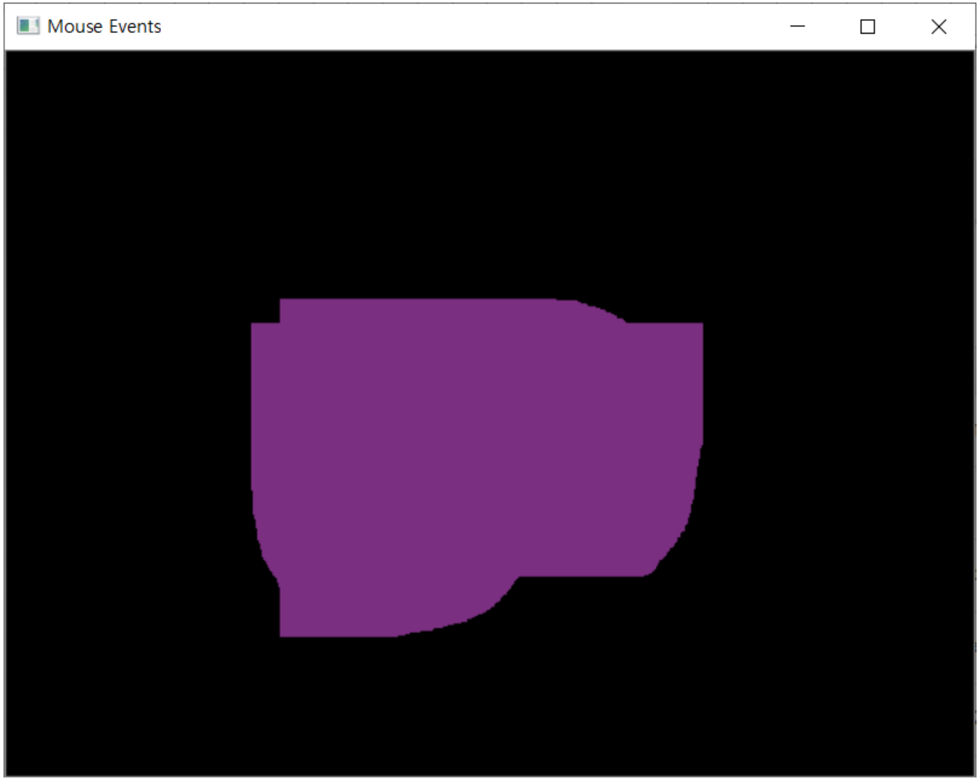
```
import cv2
import random

# Global variables
mouse_is_pressed = False
mouse_start_x = -1
mouse_start_y = -1
color = (255, 255, 255)

# Mouse event callback
def mouse_callback(event, x, y, flags, param):
    global mouse_is_pressed, mouse_start_x, mouse_start_y, color

    # Left button pressed
    if event == cv2.EVENT_LBUTTONDOWN:
        # Flag on
        mouse_is_pressed = True

        # Record the mouse position
        mouse_start_x = x
        mouse_start_y = y

        # Pick a random color
```

Result 2: Draw Shapes



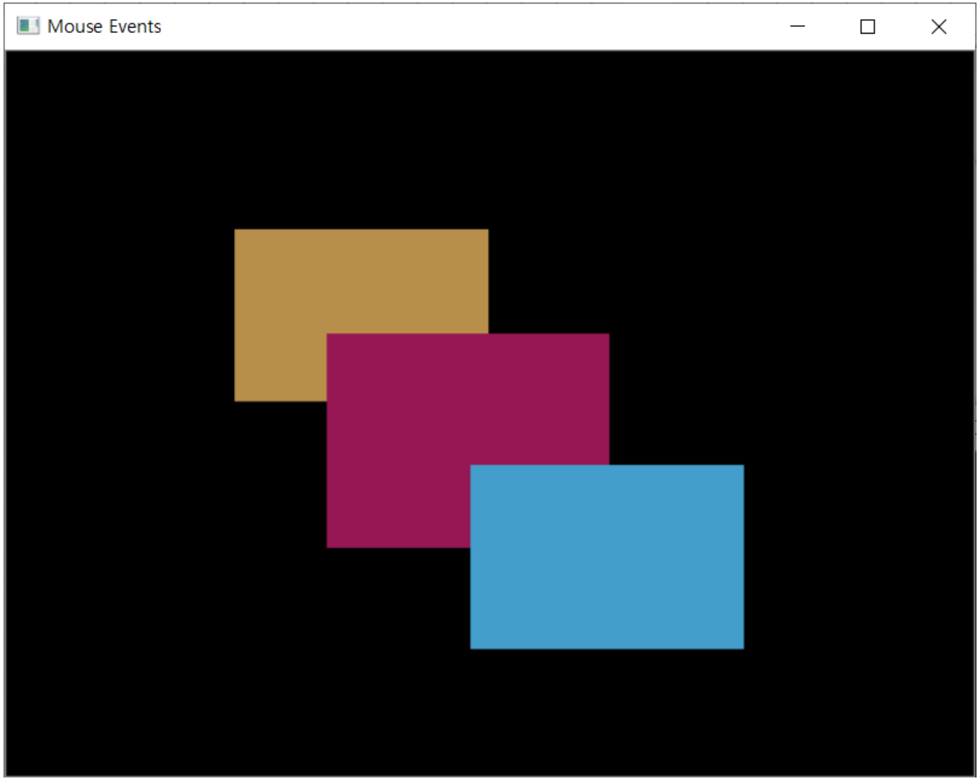
마우스를 드래그 하는 동안에도 사각형이 보이게 하려면?

Result 2: Draw Shapes



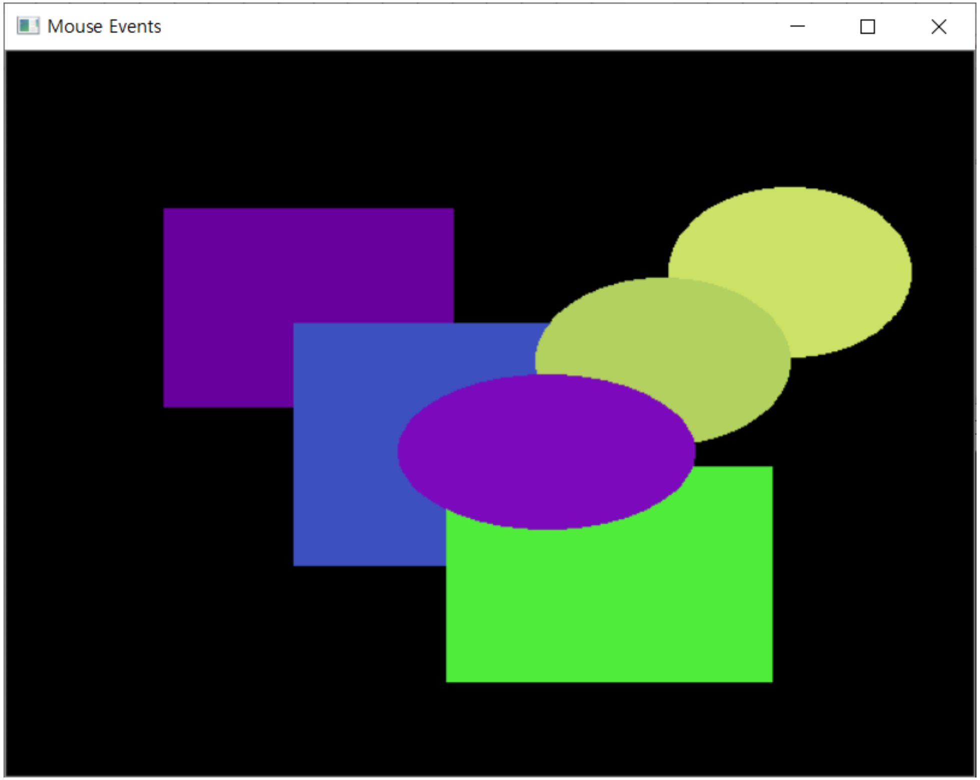
마우스를 드래그 하는 동안에도 사각형을 덮어칠하지 않게 하려면?

Result 2: Draw Shapes



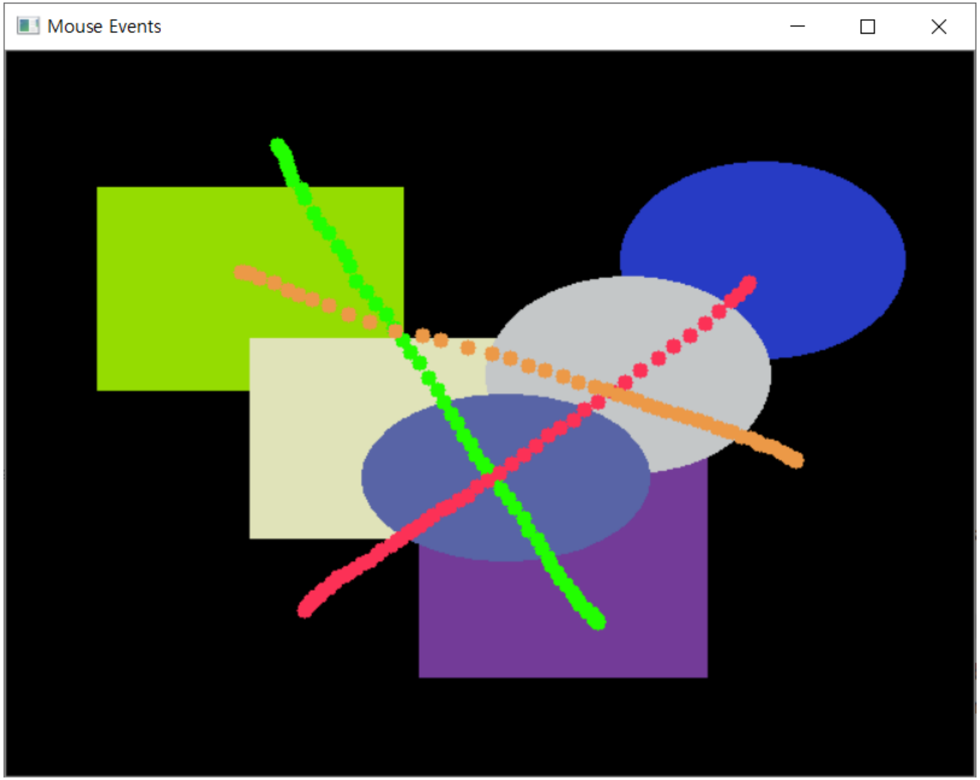
키보드로 m을 입력하면 사각형/타원을 그리도록 하려면?

Result 2: Draw Shapes



키보드로 m을 입력하면 사각형/타원/브러쉬를 그리도록 하려면?

Result 2: Draw Shapes



Trackbar

`cv.createTrackbar()`[1]

```
retval = cv.createTrackbar(trackbarname, winname, value, count,
onChange[, userdata])
```

- Trackbar를 만들어서 해당 window에 붙인다.
  - `trackbarname`: Name of the created trackbar
  - `winname`: Name of the window that will be used as a parent of the created trackbar
  - `value`: Optional pointer to an integer variable whose value reflects the position of the slider.
  - `count`: Maximal position of the slider. The minimal position is always 0.
  - `onChange`: Pointer to the function to be called every time the slider changes position.
  - `userdata`: User data that is passed as is to the callback.

[1] https://docs.opencv.org/4.4.0/d7/dfc/group__highgui.html#gaf78d2155d30b728fc413803745b67a9b

`onChange()`[1]

```
onChange(trackbar_position[, userdata])
```

- Trackbar의 callback function을 정의한다.
  - `trackbar_position`: Trackbar position
  - `userdata`: User data

[1] https://docs.opencv.org/4.4.0/d7/dfc/group__highgui.html#gaf78d2155d30b728fc413803745b67a9b

`cv.getTrackbarPos()`[1]

```
retval = cv.getTrackbarPos(trackbarname, winname)
```

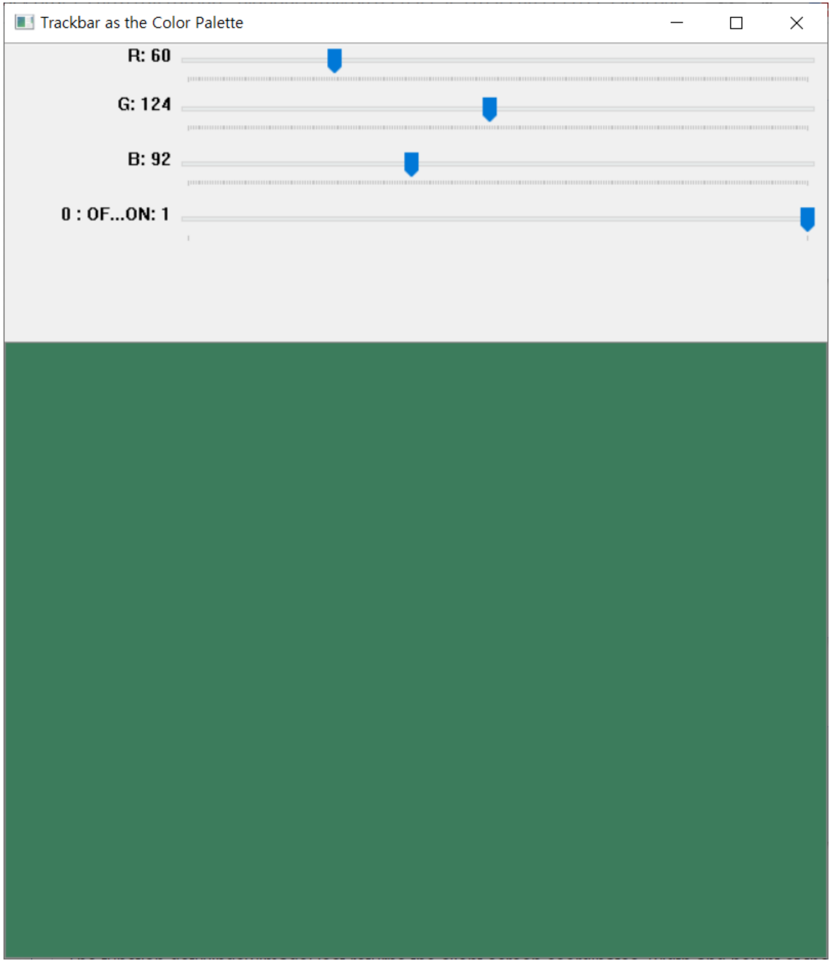
- Trackbar를 만들어서 해당 window에 붙인다.
  - `trackbarname`: Name of the created trackbar
  - `winname`: Name of the window that is the parent of the trackbar.
  - `retval`: Current position of the specified trackbar

[1] https://docs.opencv.org/4.4.0/d7/dfc/group__highgui.html#ga122632e9e91b9ec06943472c55d9cda8

Example: Trackbar as the Color Palette

```
import numpy as np
import random

# mouse callback function
def mouse_callback(event, x, y, flags, param):
    # Left button double clicked
    if event == cv2.EVENT_LBUTTONDBLCLK:
        # Pick a random radius
        radius = random.randrange(10, 50)

        # Pick a random color
        color = (random.randrange(256), random.randrange(256),
random.randrange(256))

        # Draw a circle
        cv2.circle(img_color, (x,y), radius, color,-1)

# Create a black image
rows = 480
cols = 640
img_color = np.zeros((rows, cols, 3), np.uint8)

# Create a window
```

Example: Trackbar as the Color Palette



Trackbar as the Color Palette

R: 60

G: 124

B: 92

0 : OF...ON: 1

Push Code to GitHub

References

References

References

- OpenCV Python Tutorials
  - GUI Features in OpenCV
    - Mouse as a Paint-Brush
    - Trackbar as the Color Palette