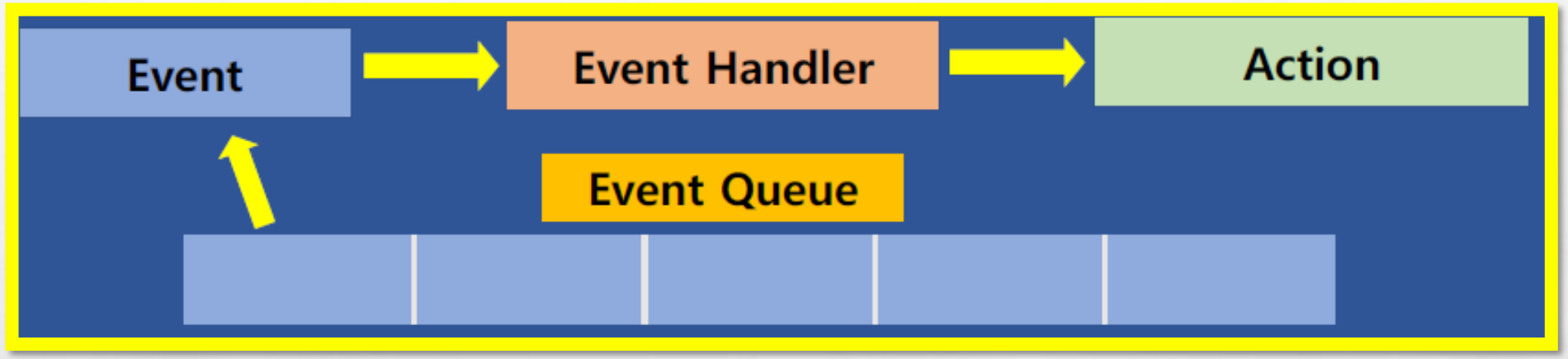
13. event

* using VSCode
* made by 세현 쌤

event?

- 단순한, 일종의 **action**
- ex) 마우스 클릭, 키보드 입력
- 한 event는 **다른 event**로 이어진다!

Python event handler



- class의 일종 (event 중 하나)
- event handler를 통해 일련의 event들을 관리하고 차례대로 수행

Python event 목록

event 종류	event 설명
pygame.QUIT	게임종료버튼 클릭 시 발생
pygame.KEYDOWN	키보드 누른 후 뗄 때 발생
pygame.MOUSEMOTION	마우스가 움직일 때 발생
pygame.MOUSEBUTTONDOWN	마우스 버튼을 눌렀을 때 발생
pygame.MOUSEBUTTONUP	마우스 버튼 누른 후 뗄 때 발생

```python
import sys
import pygame
from pygame.locals import QUIT, MOUSEBUTTONDOWN

pygame.init()
SURFACE = pygame.display.set_mode((400,300))
FPSCLOCK = pygame.time.Clock()

def main():
    mousepos = []

    while True:
        SURFACE.fill((255, 255, 255))

        for event in pygame.event.get():
            if event.type == QUIT:
                pygame.quit()
                sys.exit()
            elif event.type == MOUSEBUTTONDOWN:
                mousepos.append(event.pos)

        for pos in mousepos:
            pygame.draw.circle(SURFACE, (0, 255, 0), pos, 5)

        pygame.display.update()
        FPSCLOCK.tick(10)

if __name__ == '__main__':
    main()

```

mouse click event

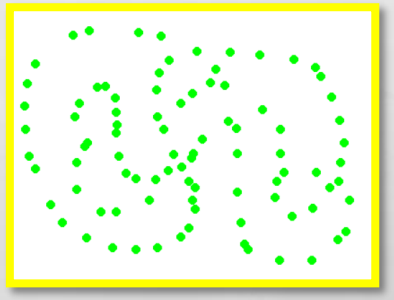
마우스 클릭한 곳에 점을 그리는 code!

* pygame의 총 2개의 event 학습!

1) pygame.QUIT

2) pygame.MOUSEBUTTONDOWN

** 실행결과



code 이해 (1)

```
def main():
    mousepos = []
```

```
for event in pygame.event.get():
    if event.type == QUIT:
        pygame.quit()
        sys.exit()
    elif event.type == MOUSEBUTTONDOWN:
        mousepos.append(event.pos)
```

Q1. QUIT이 발생하면 어떻게 될까?

Q2. MOUSEBUTTONDOWN이 발생하면?

Q3. event.pos는 무엇을 말하는 걸까?

Q4. for 문을 돌리는 이유!

code 이해 (2)

직접 circle을 그리는 code!

```
for pos in mousepos:
    pygame.draw.circle(SURFACE, (0, 255, 0), pos, 5)
```

* pygame.draw.circle()을 이해해보자

docu 확인

draw a circle

`circle(surface, color, center, radius) -> Rect`

`circle(surface, color, center, radius, width=0, draw_top_right=None, draw_top_left=None, draw_bottom_left=None, draw_bottom_right=None) -> Rect`

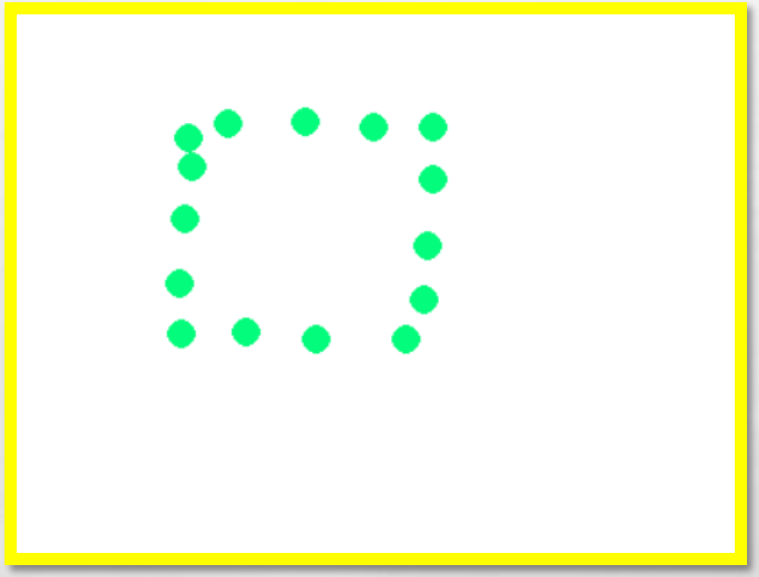
Draws a circle on the given surface.

Parameters:

- **surface** (***Surface***) -- surface to draw on
- **color** (***Color*** *or int or tuple(int, int, int, [int])*) -- color to draw with, the alpha value is optional if using a tuple `(RGB[A])`
- **center** (*tuple(int or float, int or float) or list(int or float, int or float) or* ***Vector2****(int or float, int or float)*) -- center point of the circle as a sequence of 2 ints/floats, e.g. `(x, y)`
- **radius** (*int or float*) -- radius of the circle, measured from the `center` parameter, nothing will be drawn if the `radius` is less than 1

* 문제 1.

Q. 반지름이 8이고, rgb(3, 252, 123)값을 가지는 circle 점을 여러 개 찍어 하나의 정사각형을 마우스로 그려보자



```python
import sys
import pygame
from pygame.locals import QUIT, MOUSEBUTTONDOWN, MOUSEMOTION, MOUSEBUTTONUP

pygame.init()
SURFACE = pygame.display.set_mode((400, 300))
FPSCLOCK = pygame.time.Clock()

def main():
    mousepos = []
    mousedown = False

    while True:
        for event in pygame.event.get():
            if event.type == QUIT:
                pygame.quit()
                sys.exit()

            elif event.type == MOUSEBUTTONDOWN:
                mousedown = True

            elif event.type == MOUSEMOTION:
                if mousedown:
                    mousepos.append(event.pos)

            elif event.type == MOUSEBUTTONUP:
                mousedown = False
                mousepos.clear()

        SURFACE.fill((255, 255, 255))

        if len(mousepos) > 1:
            pygame.draw.lines(SURFACE, (255, 0, 0), False, mousepos)
        pygame.display.update()
        FPSCLOCK.tick(10)

if __name__ == '__main__':
    main()
```

mouse move event

마우스 궤적을 선으로 그리는 code!

*** pygame 총 4개 event 학습!**

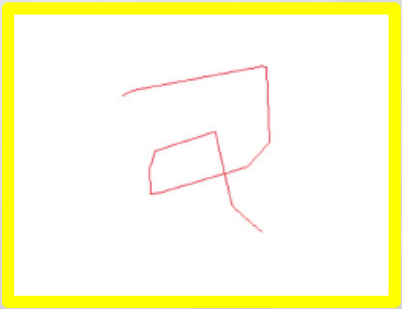
1) pygame.QUIT

2) pygame.MOUSEBUTTONDOWN

3) pygame.MOUSEMOTION

4) pygame.MOUSEBUTTONUP

** 실행결과



code 이해 (1)

```
elif event.type == MOUSEBUTTONDOWN:
    mousedown = True

elif event.type == MOUSEMOTION:
    if mousedown:
        mousepos.append(event.pos)

elif event.type == MOUSEBUTTONUP:
    mousedown = False
    mousepos.clear()
```

- 마우스 버튼을 눌렀을 때?
- 마우스 버튼을 누른 채로 움직일 때?
- 마우스 버튼을 누르고 뗐을 때?

code 이해 (2)

직접 line을 그리는 code!

```
if len(mousepos) > 1:
    pygame.draw.lines(SURFACE, (255, 0, 0), False, mousepos)
```

* 앞선 circle()과 마찬가지로 **pygame.draw.lines()**를 이해해보자

pygame.draw.lines()

draw multiple contiguous straight line segments

lines(surface, color, closed, points) -> Rect

lines(surface, color, closed, points, width=1) -> Rect

Draws a sequence of contiguous straight lines on the given surface. There are no endcaps or miter joints. For thick lines the ends are squared off. Drawing thick lines with sharp corners can have undesired looking results.

Parameters:

- **surface** (***Surface***) -- surface to draw on
- **color** (***Color*** *or int or tuple(int, int, int, [int])*) -- color to draw with, the alpha value is optional if using a tuple (RGB[A])
- **closed** (*bool*) -- if True an additional line segment is drawn between the first and last points in the points sequence
- **points** (*tuple(coordinate) or list(coordinate)*) -- a sequence of 2 or more (x, y) coordinates, where each *coordinate* in the sequence must be a tuple/list/pygame.math.Vector2 of 2 ints/floats and adjacent coordinates will be connected by a line segment, e.g. for the points [(x1, y1), (x2, y2), (x3, y3)] a line segment will be drawn from (x1, y1) to (x2, y2) and from (x2, y2) to (x3, y3), additionally if the closed parameter is True another line segment will be drawn from (x3, y3) to (x1, y1)
- **width** (*int*) --
  (optional) used for line thickness

  if width >= 1, used for line thickness (default is 1)
  if width < 1, nothing will be drawn

docu 확인

* closed가 True일 경우 그려보자!

* 문제 2.

Q. closed 속성값은 True이고, width 속성값은 3, 색깔은 rgb(3, 252, 123)값을 가지는 lines을 마우스로 그려보자

* 문제 3.

Q. 왜 마우스로 line을 그리고 난 뒤 마우스를 떼면 line이 사라지는 것일까? Code를 보면서 설명해보자

* 도전과제

Q. 공식 docu 찾아 해석하는 습관은 매우 중요! 오늘 두 개 docu로 소개한 circle()과 lines() docu를 각각 읽어보고 해석해보자

draw a circle

`circle(surface, color, center, radius) -> Rect`

`circle(surface, color, center, radius, width=0, draw_top_right=None, draw_top_left=None, draw_bottom_left=None, draw_bottom_right=None) -> Rect`

Draws a circle on the given surface.

Parameters:
- **surface** (***Surface***) -- surface to draw on
- **color** (***Color*** *or int or tuple(int, int, int, [int])*) -- color to draw with, the alpha value is optional if using a tuple (RGB[A])
- **center** (*tuple(int or float, int or float) or list(int or float, int or float) or* ***Vector2****(int or float, int or float)*) -- center point of the circle as a sequence of 2 ints/floats, e.g. (x, y)
- **radius** (*int or float*) -- radius of the circle, measured from the center parameter, nothing will be drawn if the radius is less than 1

`pygame.draw.lines()`

draw multiple contiguous straight line segments

`lines(surface, color, closed, points) -> Rect`

`lines(surface, color, closed, points, width=1) -> Rect`

Draws a sequence of contiguous straight lines on the given surface. There are no endcaps or miter joints. For thick lines the ends are squared off. Drawing thick lines with sharp corners can have undesired looking results.

Parameters:
- **surface** (***Surface***) -- surface to draw on
- **color** (***Color*** *or int or tuple(int, int, int, [int])*) -- color to draw with, the alpha value is optional if using a tuple (RGB[A])
- **closed** (*bool*) -- if True an additional line segment is drawn between the first and last points in the points sequence
- **points** (*tuple(coordinate) or list(coordinate)*) -- a sequence of 2 or more (x, y) coordinates, where each *coordinate* in the sequence must be a tuple/list/pygame.math.Vector2 of 2 ints/floats and adjacent coordinates will be connected by a line segment, e.g. for the points [(x1, y1), (x2, y2), (x3, y3)] a line segment will be drawn from (x1, y1) to (x2, y2) and from (x2, y2) to (x3, y3), additionally if the closed parameter is True another line segment will be drawn from (x3, y3) to (x1, y1)
- **width** (*int*) --
  (optional) used for line thickness

  if width >= 1, used for line thickness (default is 1)
  if width < 1, nothing will be drawn

```python
import sys
import pygame
from pygame.locals import QUIT, KEYDOWN, K_LEFT, K_RIGHT, K_UP, K_DOWN

pygame.init()
pygame.key.set_repeat(5,5)
SURFACE = pygame.display.set_mode((400, 300))
FPSCLOCK = pygame.time.Clock()

def main():
    logo = pygame.image.load("pythonlogo.jpg")
    pos = [200, 150]
    while True:
        for event in pygame.event.get():
            if event.type == QUIT:
                pygame.quit()
                sys.exit()
            elif event.type == KEYDOWN:
                if event.key == K_LEFT:
                    pos[0] -= 5
                elif event.key == K_RIGHT:
                    pos[0] += 5
                elif event.key == K_UP:
                    pos[1] -= 5
                elif event.key == K_DOWN:
                    pos[1] += 5
        pos[0] = pos[0] % 400
        pos[1] = pos[1] % 300

        SURFACE.fill((225, 225, 225))
        rect = logo.get_rect()
        rect.center = pos
        SURFACE.blit(logo, rect)
        pygame.display.update()
        FPSCLOCK.tick(30)

if __name__ == '__main__':
    main()
```

key press event

방향키로 그림을 움직이는 code!

* pygame.KEYDOWN
- K_LEFT
- K_RIGHT
- K_UP
- K_DOWN

** 실행결과

```
logo = pygame.image.load("pythonlogo.jpg")
pos = [200, 150]
```

```
elif event.type == KEYDOWN:
    if event.key == K_LEFT:
        pos[0] -= 5
    elif event.key == K_RIGHT:
        pos[0] += 5
    elif event.key == K_UP:
        pos[1] -= 5
    elif event.key == K_DOWN:
        pos[1] += 5
```

```
rect = logo.get_rect()
rect.center = pos
SURFACE.blit(logo, rect)
pygame.display.update()
```

code 이해 (1)

- 왼쪽 방향키 눌렀을 때
- 오른쪽 방향키 눌렀을 때
- 위쪽 방향키 눌렀을 때
- 아래쪽 방향키 눌렀을 때

code 이해 (2)

Q. 만약 pos가 범위를 벗어났다면?..

```
pos[0] = pos[0] % 400
pos[1] = pos[1] % 300
```

* 문제 4.

Q. 방향키를 눌렀을 때 로고가 움직이는 속도가 크게 나오는 code부터 순서대로 나열!

```
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == KEYDOWN:
            if event.key == K_LEFT:
                pos[0] -= 5
            elif event.key == K_RIGHT:
                pos[0] += 5
            elif event.key == K_UP:
                pos[1] -= 5
            elif event.key == K_DOWN:
                pos[1] += 5
    pos[0] = pos[0] % 400
    pos[1] = pos[1] % 300
```

```
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == KEYDOWN:
            if event.key == K_LEFT:
                pos[0] -= 10
            elif event.key == K_RIGHT:
                pos[0] += 10
            elif event.key == K_UP:
                pos[1] -= 10
            elif event.key == K_DOWN:
                pos[1] += 10
    pos[0] = pos[0] % 400
    pos[1] = pos[1] % 300
```

```
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == KEYDOWN:
            if event.key == K_LEFT:
                pos[0] -= 2
            elif event.key == K_RIGHT:
                pos[0] += 2
            elif event.key == K_UP:
                pos[1] -= 2
            elif event.key == K_DOWN:
                pos[1] += 2
    pos[0] = pos[0] % 400
    pos[1] = pos[1] % 300
```

* 오늘 배운 것 정리!

1> event의 개념

2> event 종류 다섯가지

3> docu 읽기 연습

4> circle()과 lines() 함수 익히기 & 응용

5> key event 움직이는 속도 빠르게 하는 법