

함수와 파이게임 입력

순천향대학교 컴퓨터공학과 이 상 정

순천향대학교 컴퓨터공학과

1

GUI 설계기법

학습 내용

- 함수를 사용한 그리기
- 키보드 입력
 - 키보드 입력 이벤트
- 마우스 입력
 - 마우스 입력 이벤트
- 입력과 상호작용하여 물체의 이동

순천향대학교 컴퓨터공학과

2

함수와 파이게임 입력

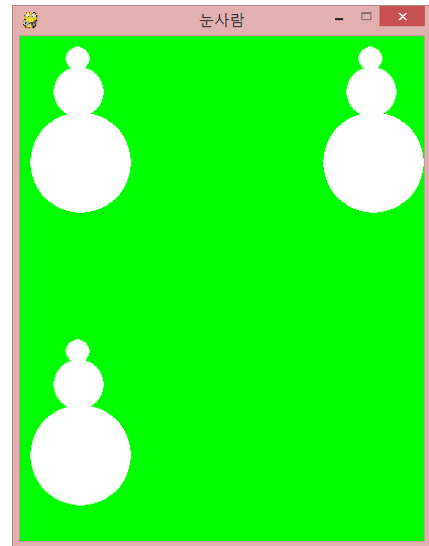
함수를 사용한 그리기: 눈사람 예

- 같은 물체를 스크린에 다른 위치에 표시
 - x, y 좌표를 인수로 받는 함수 작성

□ 눈사람 그리기 예

```
# 특정 위치에 눈사람을 그리는 함수
def draw_snowman(screen,x,y):
    pygame.draw.ellipse(screen,white,[35+x,0+y,25,25])
    pygame.draw.ellipse(screen,white,[23+x,20+y,50,50])
    pygame.draw.ellipse(screen,white,[0+x,65+y,100,100])

# 왼쪽 위 눈사람
draw_snowman(screen,10,10)
# 오른쪽 위 눈사람
draw_snowman(screen,300,10)
# 왼쪽 아래 눈사람
draw_snowman(screen,10,300)
```



GUI 설계기법

```
import pygame

# 특정 위치에 눈사람을 그리는 함수
def draw_snowman(screen,x,y):
    pygame.draw.ellipse(screen,white,[35+x,0+y,25,25])
    pygame.draw.ellipse(screen,white,[23+x,20+y,50,50])
    pygame.draw.ellipse(screen,white,[0+x,65+y,100,100])

pygame.init()

black = [ 0, 0, 0]
white = [255,255,255]
blue = [ 0, 0,255]
green = [ 0,255, 0]
red = [255, 0, 0]

# 윈도우 설정
size=[400,500]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("눈사람")

# 사용자 종료 클릭 하기 전까지 루프 수행
done=False
clock = pygame.time.Clock()
```

눈사람 코드

```
while done==False:
    # 초당 10 프레임 화면 표시
    clock.tick(10)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

    # 스크린 클리어 및 배경색 설정
    screen.fill(green)

    # 왼쪽 위 눈사람
    draw_snowman(screen,10,10)
    # 오른쪽 위 눈사람
    draw_snowman(screen,300,10)
    # 왼쪽 아래 눈사람
    draw_snowman(screen,10,300)

    # 디스플레이 갱신 및 화면 표시
    pygame.display.flip()

pygame.quit ()
```

스틱 맨 (Stick Man) 그리기: 고정 위치 그리기

```

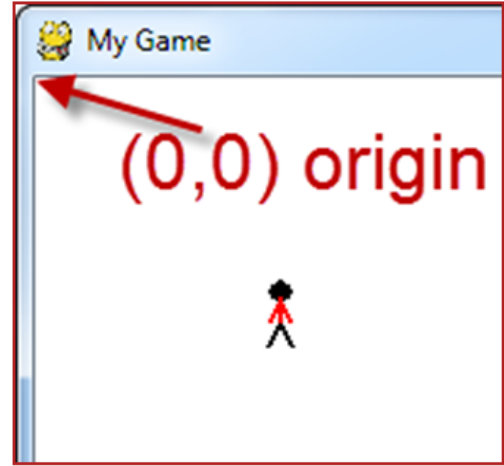
# Head
pygame.draw.ellipse(screen,black,[96,83,10,10],0)

# Legs
pygame.draw.line(screen,black,[100,100],[105,110],2)
pygame.draw.line(screen,black,[100,100],[95,110],2)

# Body
pygame.draw.line(screen,red,[100,100],[100,90],2)

# Arms
pygame.draw.line(screen,red,[100,90],[104,100],2)
pygame.draw.line(screen,red,[100,90],[96,100],2)

```



```

10 green = ( 0, 255, 0)
11 red = ( 255, 0, 0)
12
13
14 def draw_stick_figure(screen,x,y):
15     # Head
16     pygame.draw.ellipse(screen,black,[96+x,83+y,10,10],0)
17
18     # Legs
19     pygame.draw.line(screen,black,[100+x,100+y],[105+x,110+y],2)
20     pygame.draw.line(screen,black,[100+x,100+y],[95+x,110+y],2)
21
22     # Body
23     pygame.draw.line(screen,red,[100+x,100+y],[100+x,90+y],2)
24
25     # Arms
26     pygame.draw.line(screen,red,[100+x,90+y],[104+x,100+y],2)
27     pygame.draw.line(screen,red,[100+x,90+y],[96+x,100+y],2)
28
29 pygame.init()
30
31 # Set the width and height of the screen [width,height]
32 size=[700,500]
33 screen=pygame.display.set_mode(size)
34
35 pygame.display.set_caption("My Game")
36
37 #Loop until the user clicks the close button.
38 done=False
39
40 # Used to manage how fast the screen updates
41 clock=pygame.time.Clock()
42
43 # ----- Main Program Loop -----
44 while done==False:
45     # ALL EVENT PROCESSING SHOULD GO BELOW THIS COMMENT
46     for event in pygame.event.get(): # User did something
47         if event.type == pygame.QUIT: # If user clicked close
48             done=True # Flag that we are done so we exit this loop
49         # ALL EVENT PROCESSING SHOULD GO ABOVE THIS COMMENT
50
51         # ALL CODE TO DRAW SHOULD GO BELOW THIS COMMENT
52
53         # First, clear the screen to white. Don't put other drawing commands
54         # above this, or they will be erased with this command.
55         screen.fill(white)
56
57         # Head
58         pygame.draw.ellipse(screen,black,[96+x,83+y,10,10],0)
59
60         # Legs
61         pygame.draw.line(screen,black,[100+x,100+y],[105+x,110+y],2)
62         pygame.draw.line(screen,black,[100+x,100+y],[95+x,110+y],2)
63
64         # Body
65         pygame.draw.line(screen,red,[100+x,100+y],[100+x,90+y],2)
66
67         # Arms
68         pygame.draw.line(screen,red,[100+x,90+y],[104+x,100+y],2)
69         pygame.draw.line(screen,red,[100+x,90+y],[96+x,100+y],2)
70
71         # Call draw_stick_figure function
72         draw_stick_figure(screen, 50, 50)

```

스틱맨 그리기: 함수화

```

def draw_stick_figure(screen,x,y):
    # Head
    pygame.draw.ellipse(screen,black,[96,83,10,10],0)

    # Legs
    pygame.draw.line(screen,black,[100,100],[105,110],2)
    pygame.draw.line(screen,black,[100,100],[95,110],2)

    # Body
    pygame.draw.line(screen,red,[100,100],[100,90],2)

    # Arms
    pygame.draw.line(screen,red,[100,90],[104,100],2)
    pygame.draw.line(screen,red,[100,90],[96,100],2)

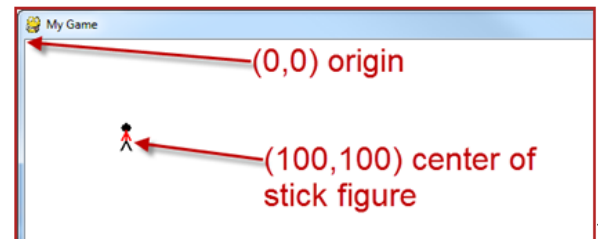
```

- x,y 위치에 무관하게 고정된 위치에 표시

스틱맨 그리기: 지정 위치 그리기 (1)

- 지정된 x,y 좌표에서 100픽셀 떨어진 위치에 그림

- `draw_stick_figure(screen,0,0)`



```
def draw_stick_figure(screen,x,y):
    # Head
    pygame.draw.ellipse(screen,black,[96+x,83+y,10,10],0)

    # Legs
    pygame.draw.line(screen,black,[100+x,100+y],[105+x,110+y],2)
    pygame.draw.line(screen,black,[100+x,100+y],[95+x,110+y],2)

    # Body
    pygame.draw.line(screen,red,[100+x,100+y],[100+x,90+y],2)

    # Arms
    pygame.draw.line(screen,red,[100+x,90+y],[104+x,100+y],2)
    pygame.draw.line(screen,red,[100+x,90+y],[96+x,100+y],2)
```

스틱맨 그리기: 지정 위치 그리기 (2)

- 코드의 최소 x, y 값을 배중

- 최소 x 값: 95
- 최소 y 값: 83

```
def draw_stick_figure(screen,x,y):
    # Head
    pygame.draw.ellipse(screen,black,[96-95+x,83-83+y,10,10],0)

    # Legs
    pygame.draw.line(screen,black,[100-95+x,100-83+y],[105-95+x,110-83+y],2)
    pygame.draw.line(screen,black,[100-95+x,100-83+y],[95-95+x,110-83+y],2)

    # Body
    pygame.draw.line(screen,red,[100-95+x,100-83+y],[100-95+x,90-83+y],2)

    # Arms
    pygame.draw.line(screen,red,[100-95+x,90-83+y],[104-95+x,100-83+y],2)
    pygame.draw.line(screen,red,[100-95+x,90-83+y],[96-95+x,100-83+y],2)

def draw_stick_figure(screen,x,y):
    # Head
    pygame.draw.ellipse(screen,black,[1+x,y,10,10],0)

    # Legs
    pygame.draw.line(screen,black,[5+x,17+y],[10+x,27+y],2)
    pygame.draw.line(screen,black,[5+x,17+y],[x,27+y],2)

    # Body
    pygame.draw.line(screen,red,[5+x,17+y],[5+x,7+y],2)

    # Arms
    pygame.draw.line(screen,red,[5+x,7+y],[9+x,17+y],2)
    pygame.draw.line(screen,red,[5+x,7+y],[1+x,17+y],2)
```

스틱맨 그리기 코드 (1)

```
import pygame

black = ( 0, 0, 0)
white = ( 255, 255, 255)
green = ( 0, 255, 0)
red = ( 255, 0, 0)

def draw_stick_figure(screen,x,y):
    # 머리
    pygame.draw.ellipse(screen,black,[1+x,y,10,10],0)
    # 다리
    pygame.draw.line(screen,black,[5+x,17+y],[10+x,27+y],2)
    pygame.draw.line(screen,black,[5+x,17+y],[x,27+y],2)
    # 몸
    pygame.draw.line(screen,red,[5+x,17+y],[5+x,7+y],2)
    # 팔
    pygame.draw.line(screen,red,[5+x,7+y],[9+x,17+y],2)
    pygame.draw.line(screen,red,[5+x,7+y],[1+x,17+y],2)

pygame.init()
```

순천향대학교 컴

함수와 파이게임 입력

스틱맨 그리기 코드 (2)

```
# 윈도우 설정
size=[700,500]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("스틱맨")

# 사용자 종료 클릭 하기 전까지 루프 수행
done=False
clock=pygame.time.Clock()

while done==False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

    # 스크린 클리어 및 배경색 설정
    screen.fill(white)

    draw_stick_figure(screen,0,0)
    draw_stick_figure(screen,150,150)
    draw_stick_figure(screen,300,200)
```

```
# 디스플레이 갱신 및 화면 표시
pygame.display.flip()

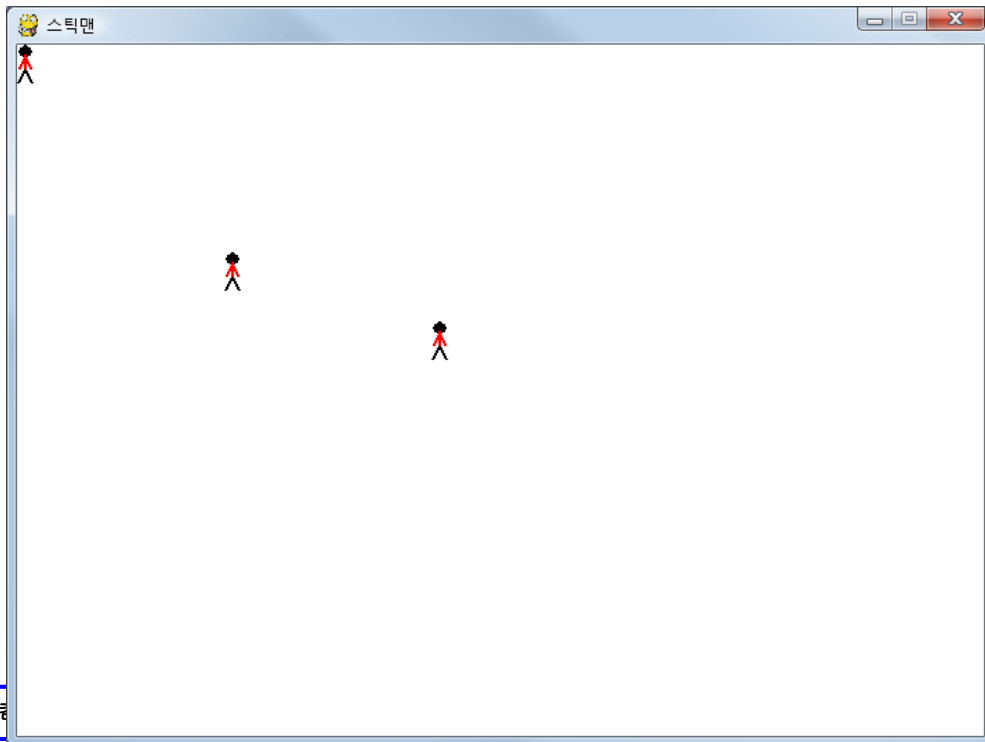
# 초당 20 프레임 화면 표시
clock.tick(20)

pygame.quit ()
```

함수와 파이게임 입력



시험주행

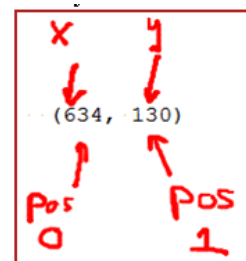


순천향대학교

이게임 입력

마우스 입력

- ❑ 마우스 커서 위치에 물체 그리기
- ❑ 마우스 커서 위치 읽기
 - `pygame.mouse.get_pos(): return (x, y)`
 - 예
 - `pos = pygame.mouse.get_pos()`



- ❑ 스틱맨을 마우스 위치에 그릴 때 마우스 커서가 그림을 가리는 것을 방지
- ❑ 마우스 커서 숨김
 - `pygame.mouse.set_visible(bool): return bool`
 - 인수가 참이면 커서 보이고, 거짓이면 숨김



스틱맨 코드: 마우스 입력 (1)

```
import pygame

black = ( 0, 0, 0)
white = ( 255, 255, 255)
green = ( 0, 255, 0)
red = ( 255, 0, 0)

def draw_stick_figure(screen,x,y):
    # 머리
    pygame.draw.ellipse(screen,black,[1+x,y,10,10],0)
    # 다리
    pygame.draw.line(screen,black,[5+x,17+y],[10+x,27+y],2)
    pygame.draw.line(screen,black,[5+x,17+y],[x,27+y],2)
    # 몸
    pygame.draw.line(screen,red,[5+x,17+y],[5+x,7+y],2)
    # 팔
    pygame.draw.line(screen,red,[5+x,7+y],[9+x,17+y],2)
    pygame.draw.line(screen,red,[5+x,7+y],[1+x,17+y],2)

pygame.init()
```

스틱맨 코드: 마우스 입력 (2)

```
# 윈도우 설정
size=[700,500]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("마우스 이동")

# 사용자 종료 클릭 하기 전까지 루프 수행
done=False
clock=pygame.time.Clock()

# 마우스 커서 숨김
pygame.mouse.set_visible(0)

while done==False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

    # 스크린 클리어 및 배경색 설정
    screen.fill(white)
```

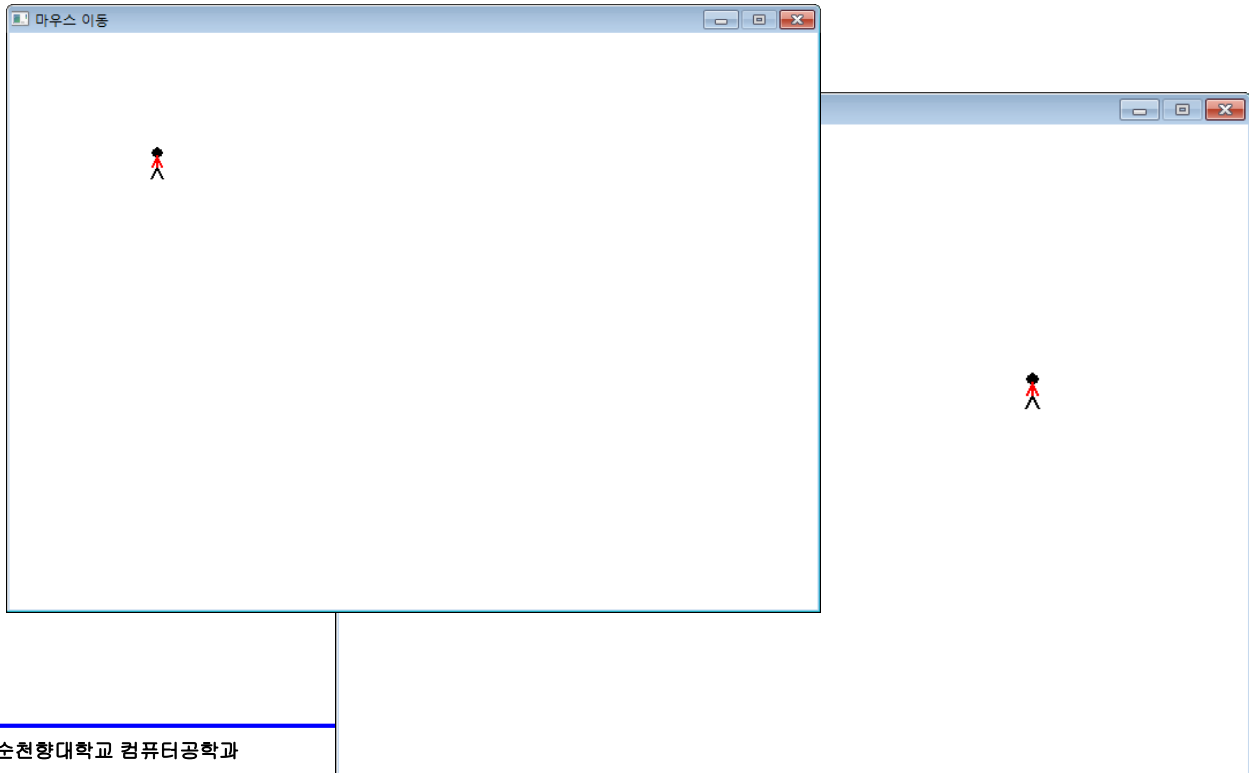
```
# 마우스 위치에 함수 호출
pos = pygame.mouse.get_pos()
x=pos[0]
y=pos[1]

draw_stick_figure(screen,x,y)

# 디스플레이 갱신 및 화면 표시
pygame.display.flip()

# 초당 20 프레임 화면 표시
clock.tick(20)

pygame.quit ()
```

- 키보드 입력으로 물체 이동
 - x, y 값을 초기화
 - 화살표 키를 누르면(keydown) 이동 (프레임 속도 조절)
 - 화살표 키를 풀면(keyup) 정지 (프레임 속도 값 리셋)
 - 속도 값에 따라 각 프레임의 x, y 값을 조정
- 이벤트 유형(event.type)
 - 키보드: `pygame.KEYDOWN`, `pygame.KEYUP`
 - 마우스: `pygame.MOUSEMOTION`, `pygame.MOUSEBUTTONUP`, `pygame.MOUSEBUTTONDOWN`
- 화살표 키보드 종류 (event.key)
 - `pygame.K_LEFT`, `pygame.K_RIGHT`
 - `pygame.K_DOWN`, `pygame.K_UP`

스틱맨 코드: 키보드 입력 (1)

```

import pygame

black = ( 0, 0, 0)
white = ( 255, 255, 255)
green = ( 0, 255, 0)
red = ( 255, 0, 0)

def draw_stick_figure(screen,x,y):
    # 머리
    pygame.draw.ellipse(screen,black,[1+x,y,10,10],0)
    # 다리
    pygame.draw.line(screen,black,[5+x,17+y],[10+x,27+y],2)
    pygame.draw.line(screen,black,[5+x,17+y],[x,27+y],2)
    # 몸
    pygame.draw.line(screen,red,[5+x,17+y],[5+x,7+y],2)
    # 팔
    pygame.draw.line(screen,red,[5+x,7+y],[9+x,17+y],2)
    pygame.draw.line(screen,red,[5+x,7+y],[1+x,17+y],2)

pygame.init()

# 윈도우 설정
size=[700,500]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("키보드 이동")

# 프레임 당 속도(픽셀/프레임)
x_speed=0
y_speed=0

# 현재 위치
x_coord=10
y_coord=10

# 사용자 종료 클릭 하기 전까지 루프 수행
done=False
clock=pygame.time.Clock()

while done==False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

```

스틱맨 코드: 키보드 입력 (2)

```

# 사용자가 키보드 누르면
if event.type == pygame.KEYDOWN:
    # 화살표 키이면 프레임 속도 조절
    if event.key == pygame.K_LEFT:
        x_speed=-3
    if event.key == pygame.K_RIGHT:
        x_speed+=3
    if event.key == pygame.K_UP:
        y_speed=-3
    if event.key == pygame.K_DOWN:
        y_speed+=3

# 사용자가 키보드를 풀면
if event.type == pygame.KEYUP:
    # 화살표 키이면 속도 리셋
    if event.key == pygame.K_LEFT:
        x_speed=0
    if event.key == pygame.K_RIGHT:
        x_speed=0

```

```

if event.key == pygame.K_UP:
    y_speed=0
if event.key == pygame.K_DOWN:
    y_speed=0

# 스크린 클리어 및 배경색 설정
screen.fill(white)

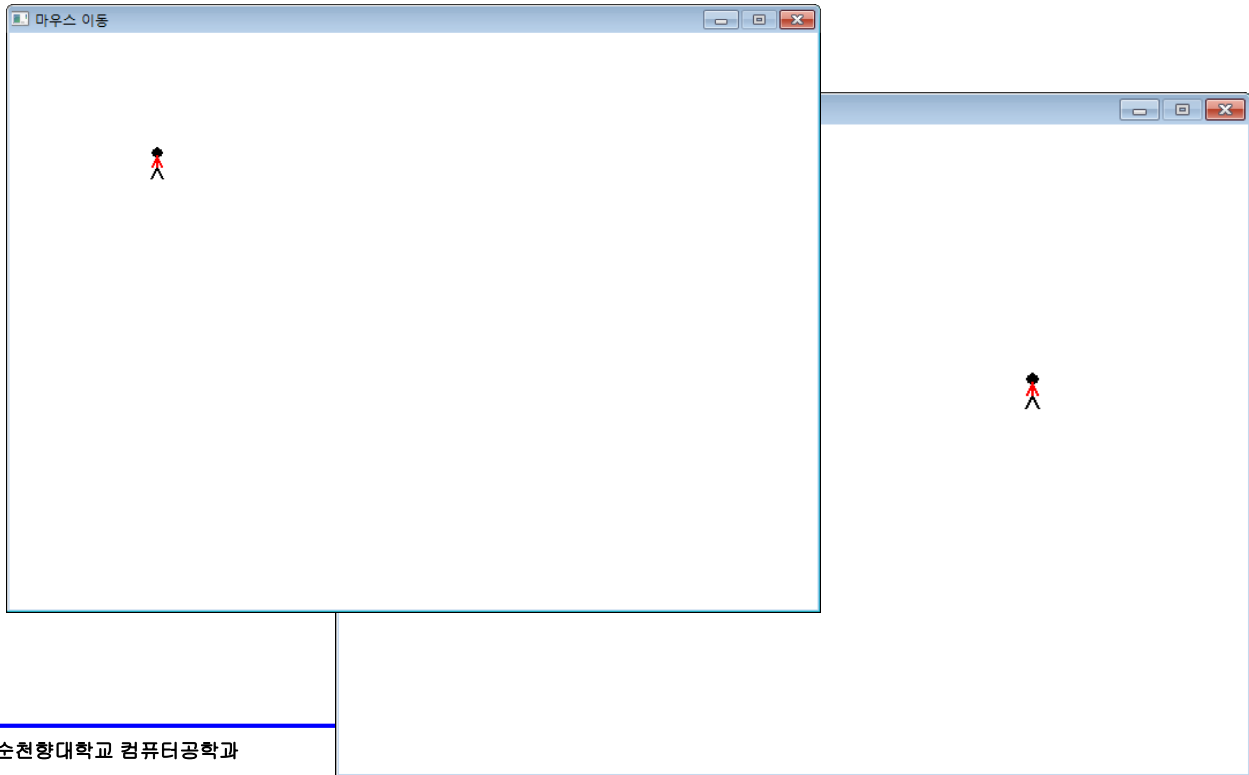
# 속도 값에 따라 물체 이동
x_coord=x_coord+x_speed
y_coord=y_coord+y_speed

draw_stick_figure(screen,x_coord,y_coord)
# 디스플레이 갱신 및 화면 표시
pygame.display.flip()

# 초당 20 프레임 화면 표시
clock.tick(20)

```

```
pygame.quit ()
```



1. 앞에서 소개된 마우스/키보드 입력의 스틱맨 프로그램을 작성하고 실행
2. 앞에서 배운 내용을 사용한 임의의 프로그램 작성
 - 프로그램 설명
 - 프로그램 소스
 - 실행 결과