

8. 파이게임 그래픽스

순천향대학교 컴퓨터공학과
이 상 정

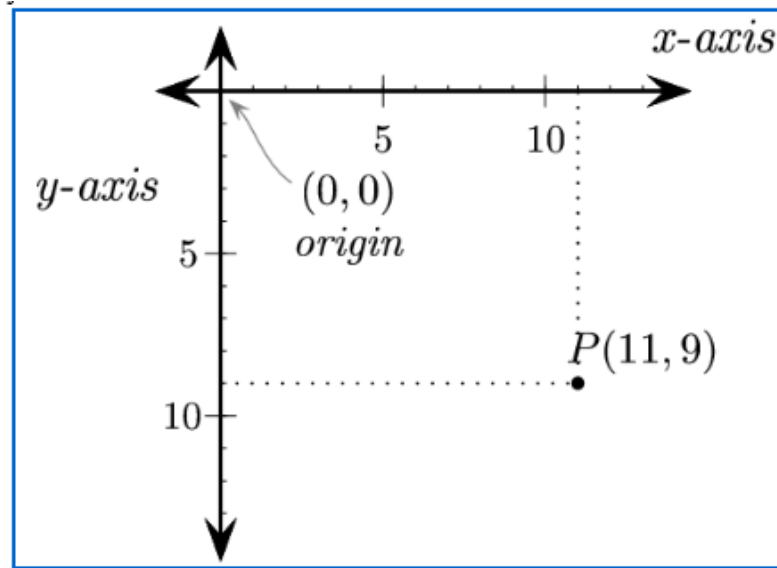
GUI 설계기법

학습 내용

- 컴퓨터 좌표 시스템
- 파이게임의 그래픽스 소개
 - 파이게임 라이브러리
- 색 표현
- 윈도우 오픈 및 설정
- 사용자 인터페이스
- 파이게임 그리기
 - 선, 도형, 폰트

컴퓨터 좌표 시스템

픽셀(pixel)의 위치 표시



파이게임 라이브러리

파이게임 라이브러리

- 그래픽 도형 그리기
- 비트맵 이미지 그리기
- 애니메이션
- 키보드, 마우스 등과의 상호 작용
- 사운드
- 물체 충돌 감지

파이게임 라이브러리 적재 및 초기화

```
import pygame          #파이게임 라이브러리 적재
pygame.init()         # 파이게임 초기화
```

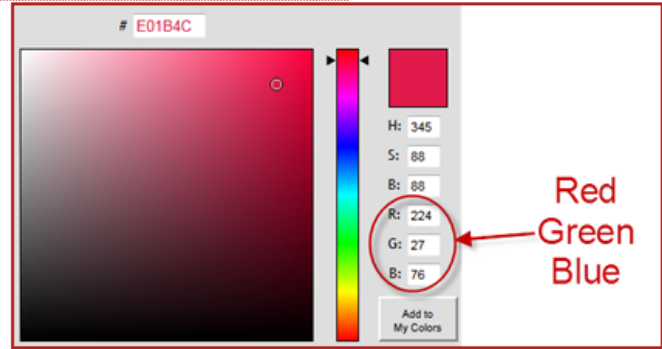
□ 색 (color)

- RGB(Red, Green, Blue) 의 조합을 리스트('[]')(또는 튜플('()'))로 표현
- 각 색은 0-255 값을 가짐

색 정의

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

<http://www.colorpicker.com/>



□ 윈도우 오픈 및 크기 설정

- `pygame.display.set_mode()` 함수
`pygame.display.set_mode(resolution=(0,0), flags=0, depth=0):`
return Surface
- 이미지를 표시하는 `pygame.Surface` 객체를 생성
윈도우 오픈과 스크린의 너비와 높이를 설정
size = (700, 500)
screen = `pygame.display.set_mode(size)`

□ 윈도우 제목 설정

- `pygame.display.set_caption()` 함수
`pygame.display.set_caption(title, icontitle=None):` return None
윈도우 제목을 설정
`pygame.display.set_caption("GUI design game")`

사용자 인터페이스

□ 사용자와 인터페이스를 위한 이벤트 루프 구성하고 다음을 처리

- 윈도우 닫기, 사용자 키보드 입력, 마우스 클릭 등과 같은 이벤트를 처리
- 물체의 이동 지정, 물체의 충돌 등을 결정하는 계산을 수행
- 스크린의 클리어, 물체 그리기
- 처리 예
 - 초당 f 프레임 속도로 윈도우를 표시
`pygame.time.Clock.tick(f)`
 => 초 당 f 프레임을 넘지 않도록 프로그램을 수행
 - 이벤트 읽기
`pygame.event.get()`
 => QUIT, KEYDOWN, KEYUP, MOUSEBUTTONDOWN 등
 입력 이벤트 타입 리스트 반환

프로그램 이벤트 루프 구성

```

# 사용자가 종료 클릭하기까지 루프 반복
done = False
# 스크린 갱신하는 속도 지정을 위한 클럭 객체 생성
clock = pygame.time.Clock()
# 프로그램 이벤트 루프
while done==False:
    # 이벤트 읽기
    for event in pygame.event.get(): # 이벤트 읽기 (EventType 인스턴스)
        if event.type == pygame.QUIT: # QUIT 이벤트이면
            done=True # 루프 종료

    # 게임을 위한 물체 그리기, 이동, 충돌 등 동작 수행
    .....

    # 디스플레이 갱신 및 화면 표시
    pygame.display.flip()
    # 초 당 20 프레임으로 제한
    clock.tick(20)

# 프로그램 종료
pygame.quit ()

```

□ 스크린 클리어 및 배경색 설정

- `Surface.fill(color)` 메서드
 - `pygame.Surface` 객체는 이미지를 표시하는 파이게임 객체
- `screen = pygame.display.set_mode(size)`
`screen.fill(white)`

□ 화면 디스플레이 갱신

- `pygame.display.flip()` 함수
 - 스크린의 전체 디스플레이 `Surface`를 갱신하여 화면에 표시
- `pygame.display.update(rectangle_list)` 함수
 - 스크린의 지정된 영역의 디스플레이 `Surface`를 갱신하여 화면에 표시
 - 인수 생략 시 전체 화면 갱신하고 표시

□ 프로그램 종료

- `pygame.quit()`

```
import pygame

# 색 정의
black = ( 0, 0, 0)
white = ( 255, 255, 255)
green = ( 0, 255, 0)
red = ( 255, 0, 0)

pygame.init()

# 윈도우 오픈 및 크기, 제목 설정
size=[700,500]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("My Game")

done=False

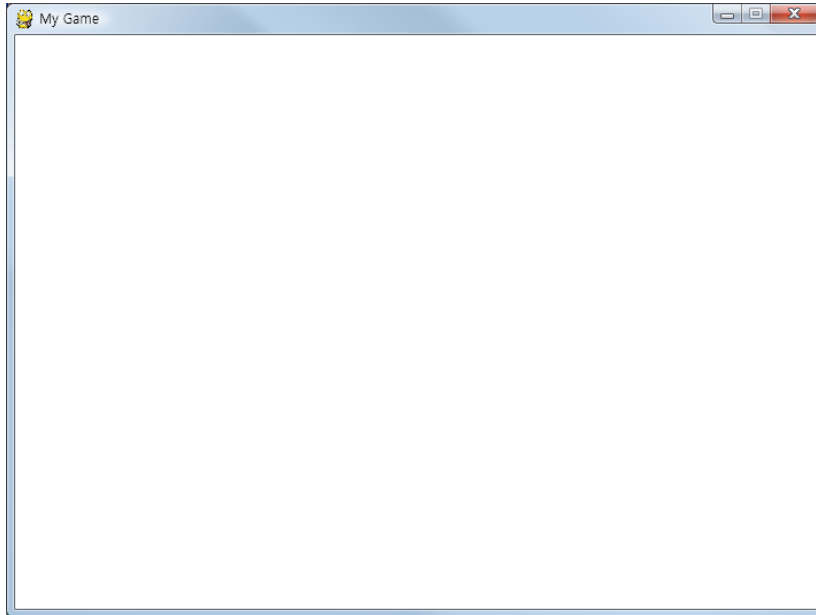
# Clock 객체 생성
clock=pygame.time.Clock()
```

```
# 이벤트 프로그램 루프
while done==False:
    # 이벤트 읽기
    for event in pygame.event.get(): # 이벤트 읽기
        if event.type == pygame.QUIT: # QUIT 이벤트이면
            done=True # 루프 종료
    # 스크린 배경색 설정
    screen.fill(white)

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

    # 초 당 20 프레임으로 제한
    clock.tick(20)

# 프로그램 종료
pygame.quit ()
```



□ 파이게임 그리기 모듈

- `pygame.draw` 모듈
- Surface에 선, 도형 등 그리기

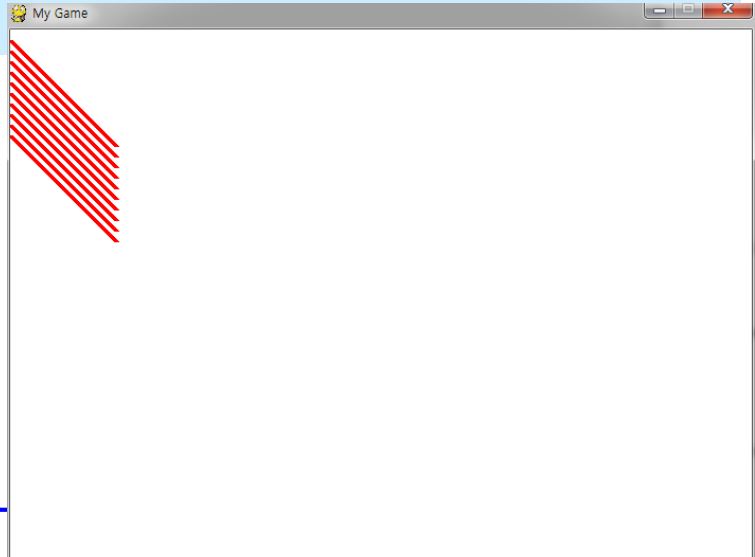
□ 선 그리기

- `pygame.draw.line(Surface, color, start_pos, end_pos, width=1):`
return Rect

좌표 (0,0)에서 (100,100)까지 5 픽셀 굵기의 빨강색 선 그리기
`pygame.draw.line(screen, red, [0, 0], [100, 100], 5)`

선 그리기 예

```
# (0,10)에서 (100,110) 부터 시작하여 여러 개의 선 그리기 예
y_offset = 0
while y_offset < 100
    pygame.draw.line(screen, red, [0, 10+y_offset], [100, 110+y_offset], 5)
    y_offset = y_offset + 10
```



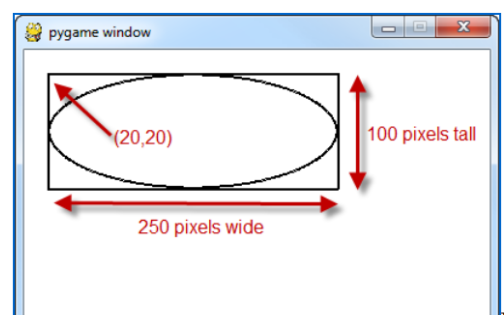
도형 그리기 - 사각형, 타원

□ 사각형

- `pygame.draw.rect(Surface, color, Rect, width=0): return Rect`
 - width = 0이면 color로 배경 채움
 - # 사각형 그리기
 - `pygame.draw.rect(screen, black, [20,20,250,100],2)`

□ 타원

- `pygame.draw.ellipse(Surface, color, Rect, width=0): return Rect`
 - width = 0이면 color로 배경 채움
 - # 타원 그리기
 - `pygame.draw.ellipse(screen, black, [20,20,250,100],2)`



호(Arc) 그리기

□ 호, 아크(arc)

- `pygame.draw.arc(Surface, color, Rect, start_angle, stop_angle, width=1): return Rect`
- 각도 표시를 위해 라디안(radian) 사용, 오른쪽이 0 (->)

호 그리기

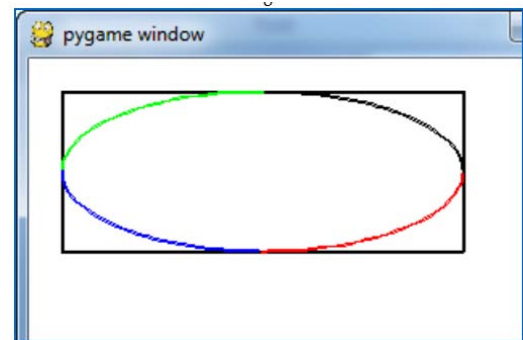
```
pygame.draw.arc(screen, green, [100,100,250,200], pi/2, pi, 2)
pygame.draw.arc(screen, black, [100,100,250,200], 0, pi/2, 2)
pygame.draw.arc(screen, red, [100,100,250,200], 3*pi/2, 2*pi, 2)
pygame.draw.arc(screen, blue, [100,100,250,200], pi, 3*pi/2, 2)
```

도를 라디안 단위로 변환

$$\text{도} * \left(\frac{\pi}{180^\circ}\right) = \text{라디안}$$

라디안을 도 단위로 변환

$$\text{라디안} * \left(\frac{180^\circ}{\pi}\right) = \text{도}$$



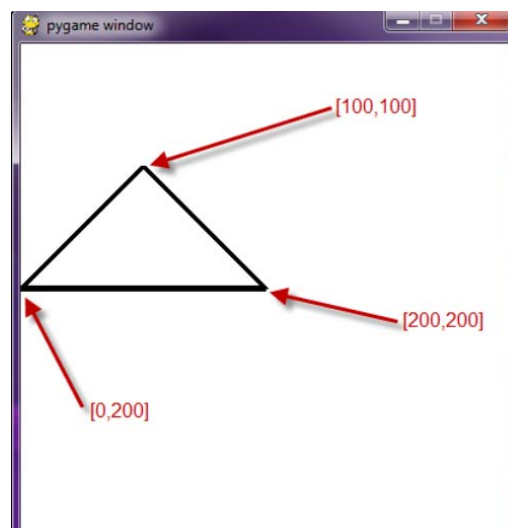
다각형(Polygon) 그리기

□ 다각형(polygon)

- `pygame.draw.polygon(Surface, color, pointlist, width=0): return Rect`

다각형 그리기

```
pygame.draw.polygon(screen, black, [[100,100], [0,200], [200,200]],5)
```



텍스트: 폰트(Font)

□ 파이게임 폰트 모듈

- pygame.font 모듈

□ 폰트 객체 생성

- pygame.font.Font(filename, size): return Font
- pygame.font.Font(object, size): return Font
 - 크기는 폰트의 높이를 픽셀로 표시
- 메서드
 - Font.render(text, antialias, color, background=None): return Surface
 - 새로운 Surface에 텍스트를 그리기
 - antialias가 true이면 글자가 부드러운 모서리(smooth edge)로 표시
 - 다른 Surface의 blit 메서드로 이를 표시

□ Surface.blit() 메서드

- 한 Surface에 다른 Surface를 그리기
- Surface.blit(source, dest, area=None, special_flags = 0):

return Rect
8. 파이게임 그래픽스

폰트 예

폰트 객체 생성, 나눔고딕 폰트 적용

```
font = pygame.font.Font("nanumgothic.ttf", 25)
```

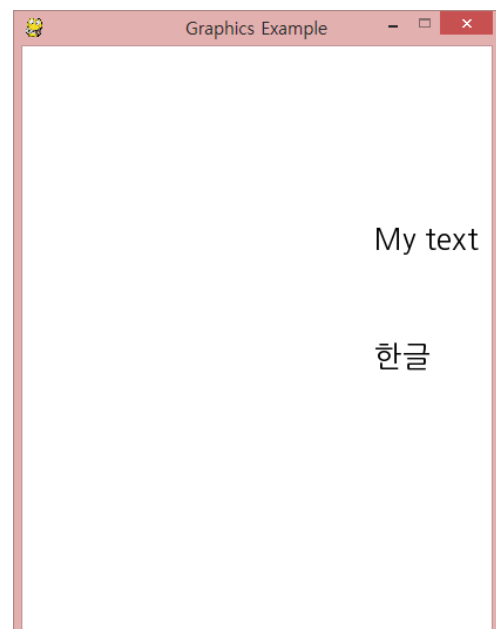
폰트 그리기

```
text1 = font.render("My text", True, black)
```

```
screen.blit(text1, [300, 150])
```

```
text2 = font.render("한글", True, black)
```

```
screen.blit(text2, [300, 250])
```



그래픽스 예: 통합 예 (1)

```

import pygame

# 파이게임 초기화
pygame.init()

# 색 정의
black = [ 0, 0, 0]
white = [255,255,255]
blue = [ 0, 0,255]
green = [ 0,255, 0]
red = [255, 0, 0]

pi=3.141592653

# 윈도우 오픈 및 크기, 제목 설정
size=[400,500]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("Graphics Example")

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

while done==False:
    # 초당 10 프레임을 넘지 않도록 프로그램을 수행
    clock.tick(10)

    # 이벤트 읽기
    for event in pygame.event.get():
        if event.type == pygame.QUIT: #QUIT 이벤트이면
            done=True # 루프 종료

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

    # 선 그리기
    pygame.draw.line(screen,green,[0,0],[100,100],5)

```

그래픽스 예: 통합 예 (2)

```

# 선 그리기
y_offset=0
while y_offset < 100:
    pygame.draw.line(screen,red,[0,10+y_offset],[100,110+y_offset],5)
    y_offset=y_offset+10

# 사각형 그리기
pygame.draw.rect(screen,black,[20,20,250,100],2)

# 타원 그리기
pygame.draw.ellipse(screen,black,[20,20,250,100],2)

# 타원 그리기, 각도 표시를 위해 라디안(radian) 사용
pygame.draw.arc(screen,black,[20,220,250,200], 0, pi/2, 2)
pygame.draw.arc(screen,green,[20,220,250,200], pi/2, pi, 2)
pygame.draw.arc(screen,blue, [20,220,250,200], pi,3*pi/2, 2)
pygame.draw.arc(screen,red, [20,220,250,200],3*pi/2, 2*pi, 2)

# 다각형 그리기
pygame.draw.polygon(screen,black,[[100,100],[0,200],[200,200]],5)

```

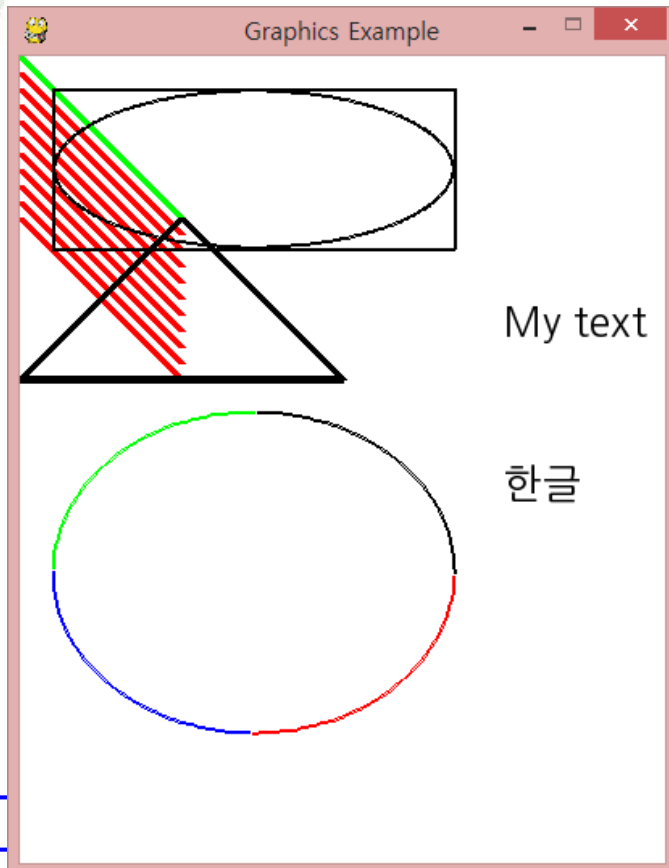
그래픽스 예: 통합 예 (3)

```
# 폰트 객체 생성
font = pygame.font.Font("nanumgothic.ttf", 25)
# 폰트 그리기
text1 = font.render("My text", True, black)
screen.blit(text1, [300, 150])
text2 = font.render("한글", True, black)
screen.blit(text2, [300, 250])
# 디스플레이 갱신 및 화면 표시
pygame.display.flip()

# 프로그램 종료
pygame.quit ()
```



시행주행



1. 앞에서 소개된 그래픽스 예를 작성하고 실행
2. 앞에서 배운 내용을 사용한 임의의 프로그램 작성
 - 프로그램 설명
 - 프로그램 소스
 - 실행 결과