9. 애니매이션 (Animation)

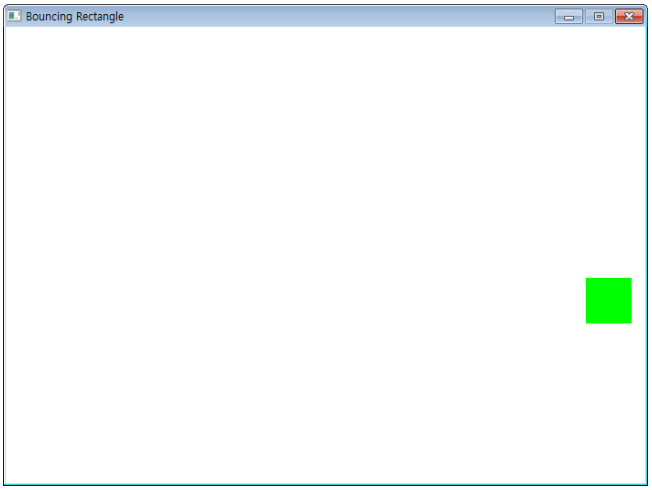
순천향대학교 컴퓨터공학과
이 상 정

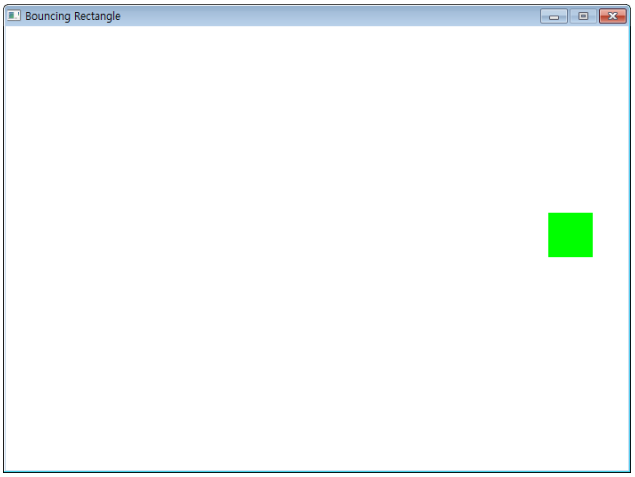
학습 내용

- 애니매이션 예 소개
- 사각형 이동 및 반동
- 클래스를 이용한 공 이동
- 눈 내리기
  - 리스트
  - 랜덤 함수

사각형 이동 및 반동: 예제 소개

- 흰색 배경에 녹색 사각형을 이동 하는 예
  - 윈도 경계에 도달 시 반사되어 이동(되튕기기, bounce)
  - x,y 방향 이동 간격(속도)은 5 픽셀





사각형 이동 및 반동: 사각형 이동 (1)

- 사각형의 좌표를 변경하여 이동
- 정적 사각형 그리기
  - pygame.draw.rect(screen,green,[50,50,50,50])
  - 좌표 (50,50)에 50x50 크기의 사각형 표시
  - width 인자가 생략되어 0이 되어 녹색으로 채움
- 사각형을 x 방향을 1 픽셀씩 이동
  - x 좌표를 변수로 하여 루프 수행

```
rect_x = 50
while done==False:
    ……
    pygame.draw.rect(screen,white,[rect_x,50,50,50])
    rect_x += 1
    ……
```

  - 더 빠르게 이동 => rect_x += 5

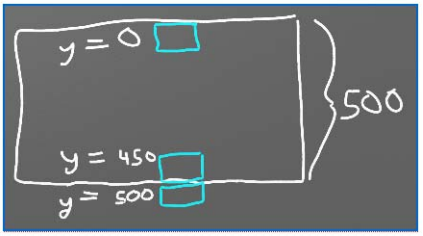
사각형 이동 및 반동: 사각형 이동 (2)

- 사각형을 x, y 방향으로 이동

```
# 사각형 시작점
rect_x = 50
rect_y = 50
# 사각형 이동 속도(간격) 및 방향
rect_change_x = 5
rect_change_y = 5
……
while done==False:
    ……
    pygame.draw.rect(screen,green,[rect_x,rect_y,50,50])
     rect_x += rect_change_x
     rect_y += rect_change_y
    ……
```


사각형 이동 및 반동: 사각형 반동

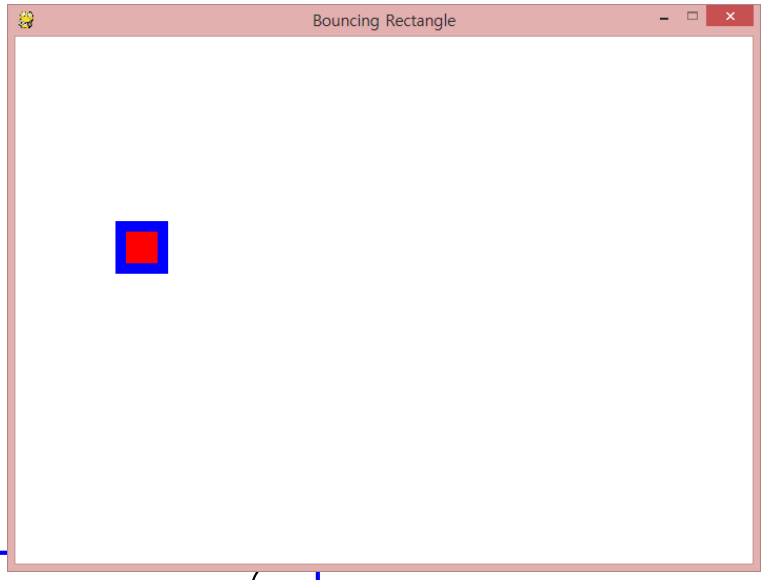
- 윈도우 경계에 도달 시 반사되어 이동(되튕기기, bounce)
  - 스크린의 경계 지점 감지하여 이동 방향 전환
  - 경계 지점은 스크린 크기에서 여백 고려하여 감지
  - 방향 전환은 이동 간격의 부호를 반전




```
screen=pygame.display.set_mode([700,500])
……
while done==False:
    ……
    # 되튕기기(bounce)
    if rect_y > 450 or rect_y < 0:                  # 아래 끝 or 위 끝 이면
        rect_change_y = rect_change_y * -1          # y 방향 전환
    if rect_x > 650 or rect_x < 0:                  # 오른쪽 끝 or 왼쪽 끝이면
        rect_change_x = rect_change_x * -1          # x 방향 전환
```

이중 사각형 이동

# 파란 사각형 안에 빨간 사각형 그리기
pygame.draw.rect(screen, blue, [rect_x,rect_y,50,50])
pygame.draw.rect(screen, red, [rect_x+10, rect_y+10,30,30])



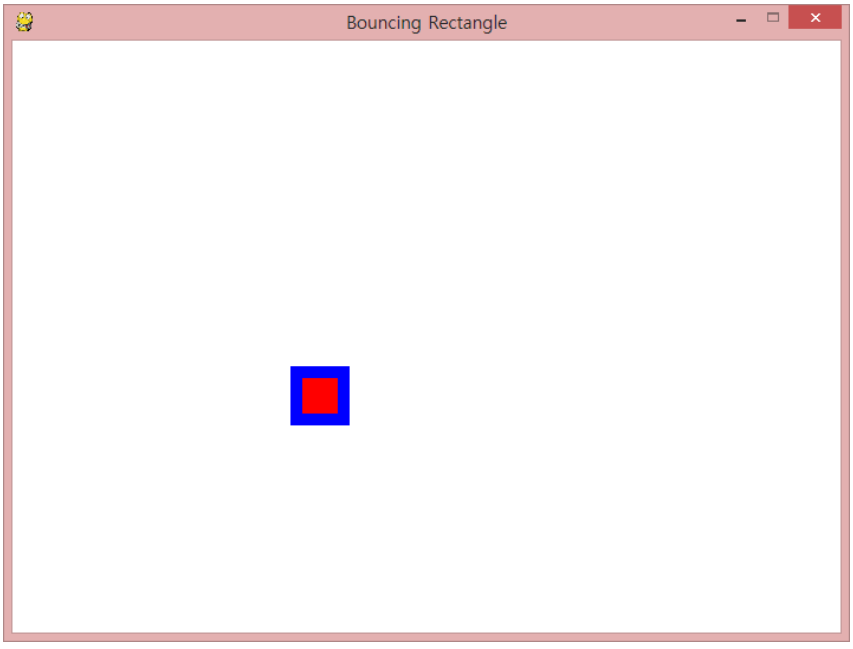
사각형 이동 및 반동 코드 (1)

```
import pygame

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

pygame.init()

# 윈도우 설정
size=[700,500]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("Bouncing Rectangle")
# 사용자 종료 클릭 하기 전까지 루프 수행
done=False
clock=pygame.time.Clock()

# 사각형 시작점
rect_x = 50
rect_y = 50

# 사각형 이동 속도(간격) 및 방향
rect_change_x = 5
rect_change_y = 5

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

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

 # 파란 사각형 안에 빨간 사각형 그리기
    pygame.draw.rect(screen, blue,
                     [rect_x,rect_y,50,50])
    pygame.draw.rect(screen, red,
                     [rect_x+10, rect_y+10,30,30])
    # 사각형 이동
    rect_x += rect_change_x
    rect_y += rect_change_y
```

사각형 이동 및 반동 코드 (2)

```
# 되튕기기(bounce)
if rect_y > 450 or rect_y < 0:                  # 아래 끝 or 위 끝 이면
    rect_change_y = rect_change_y * -1          # y 방향 전환
if rect_x > 650 or rect_x < 0:                  # 오른쪽 끝 or 왼쪽 끝이면
    rect_change_x = rect_change_x * -1          # x 방향 전환

# 초당 20 프레임 화면 표시
clock.tick(20)

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


pygame.quit ()
```


시험주행



클래스를 이용한 공 이동

- 객체 지향 프로그래밍
  - 공의 인스턴스 생성 후 속성 초기화, 메서드 호출
  - 원 그리기
    - pygame.draw.circle(Surface, color, pos, radius, width=0): return Rect
- 클래스를 사용하여 공의 속성, 메서드 정의
  - 속성
    - 공의 위치, 이동 간격
    - 크기, 색
  - 메서드
    - 공그리기
    - 이동
    - 되튕기기
- 2개의 공 이동 예

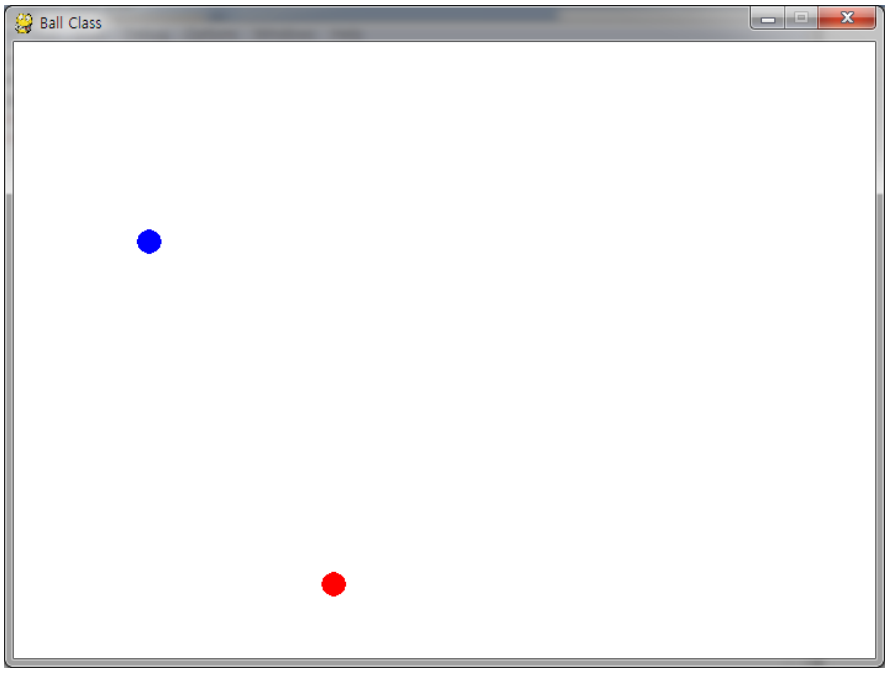
클래스를 이용한 공 이동 코드

```
import pygame

# 색 정의 [R, G, B]
WHITE = [255, 255, 255]
RED = [255, 0, 0]
BLUE = [0, 0, 255]

class Ball():
    # 공의 위치 좌표
    x=0
    y=0
    # 공의 이동 간격
    change_x=0
    change_y=0
    # 공의 크기
    size=10
    # 공의 색
    color=WHITE

    # 공 그리기
    def draw(self, screen):
        pygame.draw.circle(screen, self.color,
                           [self.x, self.y], self.size )

    # 공의 이동 좌표 계산
    def move(self):
        self.x += self.change_x
        self.y += self.change_y

    # 되튕기기(bounce)
    def bounce(self):
        if self.y > 490 or self.y < 10:
                        # 아래 끝 or 위 끝 이면
            self.change_y = self.change_y * -1
                        # y 방향 전환
        if self.x > 690 or self.x < 10:
                        # 오른쪽 끝 or 왼쪽 끝이면
            self.change_x = self.change_x * -1
                        # x 방향 전환

# 파이게임 초기화
pygame.init()

# 파이게임 윈도우 설정 (크기, 제목)
size = [700,500]
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Ball Class")
```

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

# 공의 생성 및 속성 초기화
redBall = Ball()
blueBall = Ball()

# 공의 시작점
redBall.x = 50
redBall.y = 50
blueBall.x = 600
blueBall.y = 400

# 공의 이동속도와 방향
redBall.change_x = 5
redBall.change_y = 5
blueBall.change_x = -7
blueBall.change_y = -7

# 공의 색
redBall.color = RED
blueBall.color = BLUE
```

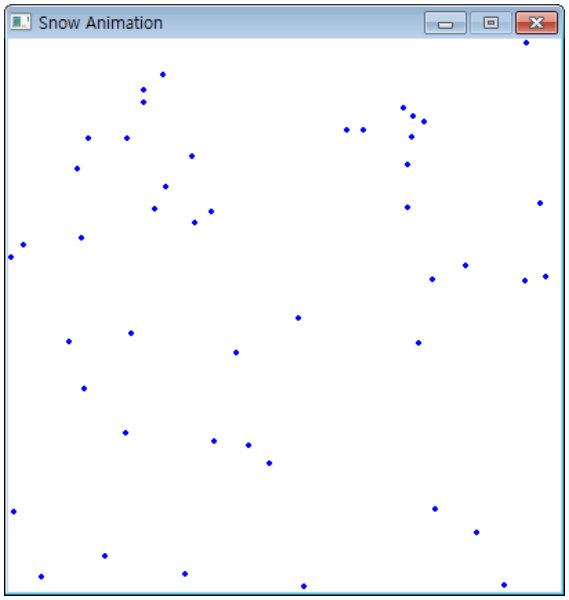
```
# 파이게임 이벤트 루프 실행
while done==False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

    # 스크린 클리어 및 배경색 설정
    screen.fill(WHITE)
    # 윈도우에 공을 그림
    redBall.draw(screen)
    blueBall.draw(screen)
    # 공을 이동
    redBall.move()
    blueBall.move()
    # 되튕기기(bounce)
    redBall.bounce()
    blueBall.bounce()
    # 초당 20 프레임 화면 표시
    clock.tick(20)

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

pygame.quit ()
```


시험주행



눈 내리기 예

- 흰색 배경에 파란 눈이 아래로 뿌려지는 예



눈 내리기 예: 임의의 위치에 눈 표시

- 화면 임의의 위치에 50개의 눈을 표시
  - 원 그리기
    - pygame.draw.circle(Surface, color, pos, radius, width=0): return Rect
  - 난수(random number) 생성
    - import random
      random.randrange([*start*], *stop*[, *step*])

```
# 50개의 눈의 임의의 위치에 눈을 표시
for i in range(50):
    x = random.randrange(0, 400)
    y = random.randrange(0, 400)
    pygame.draw.circle(screen, blue, [x,y], 2)
```

눈 내리기 예: 눈 이동 (1)

- 50개 눈의 위치를 리스트 변수에 저장한 후 이동
  - y좌표 값을 1픽셀 증가시켜 아래로 이동
  - 아래 끝에 도달하면 y값 위 끝으로 설정

- 리스트 변수 star_list에 50개 위치 저장

```
# 리스트 생성
star_list=[]
# 50개의 눈의 임의의 위치 리스트를 추가
for i in range(50):
    x=random.randrange(0, 400)
    y=random.randrange(0, 400)
    star_list.append([x, y])
```


눈 내리기 예: 눈 이동 (2)

- 리스트 변수 참조 예

```
print( star_list[0] )       # 첫 위치의 [x, y]를 출력
print( star_list[0][1] )    # 첫 위치의 y를 출력
print( star_list[i][1] )    # i 위치의 y를 출력
print( star_list[i][0] )    # i 위치의 x를 출력
```

- 눈 그리기 및 이동

```
# 리스트의 위치에 눈을 표시하고 이동
for i in range(len(star_list)):
    # 눈 그리기
    pygame.draw.circle(screen, blue, star_list[i], 2)

     # 아래로 1픽셀 이동
    star_list[i][1]+=1
```

눈 내리기 예: 눈 이동 (3)

- 눈의 스크린 아래 끝에 도달 감지 및 리셋

```
# 눈의 스크린 아래 끝에 도달 감지 및 리셋
if star_list[i][1] > 400:          # y 값이 스크린 경계 보다 크면
    # 스크린 위 끝으로 리셋
    y=random.randrange(-50,-10)
    star_list[i][1]=y
    # 새로운 임의의 x값 설정
    x=random.randrange(0,400)
    star_list[i][0]=x
```


눈 내리기 코드 (1)

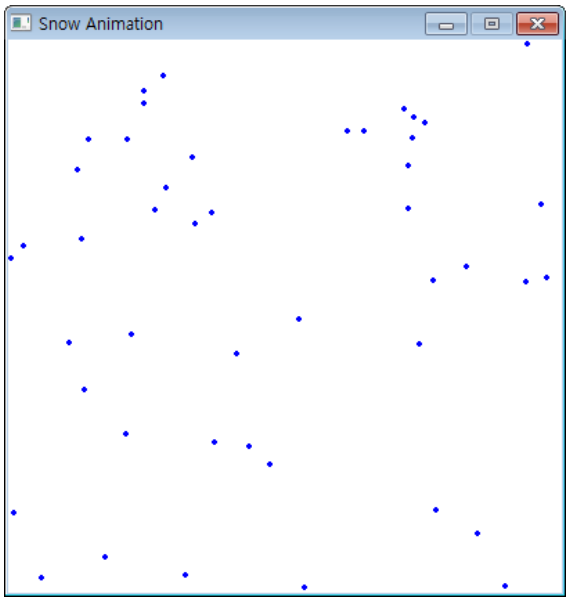
```
import pygame
import random

pygame.init()

blue = [ 0, 0, 255]
white = [255,255,255]

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

# 리스트 생성
star_list=[]

# 50개의 눈의 임의의 위치 리스트를 추가
for i in range(50):
    x=random.randrange(0,400)
    y=random.randrange(0,400)
    star_list.append([x,y])
```

```
# 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)

    # 리스트의 위치에 눈을 표시하고 이동
    for i in range(len(star_list)):
        # 눈 그리기
        pygame.draw.circle(screen,blue,star_list[i],2)

        # 아래로 1픽셀 이동
        star_list[i][1]+=1
```

눈 내리기 코드 (2)

```
    # 눈의 스크린 아래 끝에 도달 감지 및 리셋
    if star_list[i][1] > 400:     # y 값이 스크린 경계 보다 크면
        # 스크린 위 끝으로 리셋
        y=random.randrange(-50,-10)
        star_list[i][1]=y
        # 새로운 임의의 x값 설정
        x=random.randrange(0,400)
        star_list[i][0]=x

# 초당 20 프레임 화면 표시
clock.tick(20)

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


pygame.quit ()
```


시험주행



과제

1. 앞에서 소개된 클래스 공 이동과 눈내리기 애니매이션 예를 작성하고 실행

2. 앞에서 배운 내용을 사용한 임의의 프로그램 작성
   - 프로그램 설명
   - 프로그램 소스
   - 실행 결과