

CS101 - 객체 (object)로 애니메이션 만들기

Lecture 23

School of Computing
KAIST

학습 목표:

- 지금까지 강의에서 사용한 객체의 상태와 동작을 이해할 수 있다.
- 닭과 여러 병아리들이 움직이는 애니메이션을 객체를 사용해 만들 수 있다.

객체: 상태와 동작

지금까지 우리는 여러 종류의 객체를 사용했습니다.

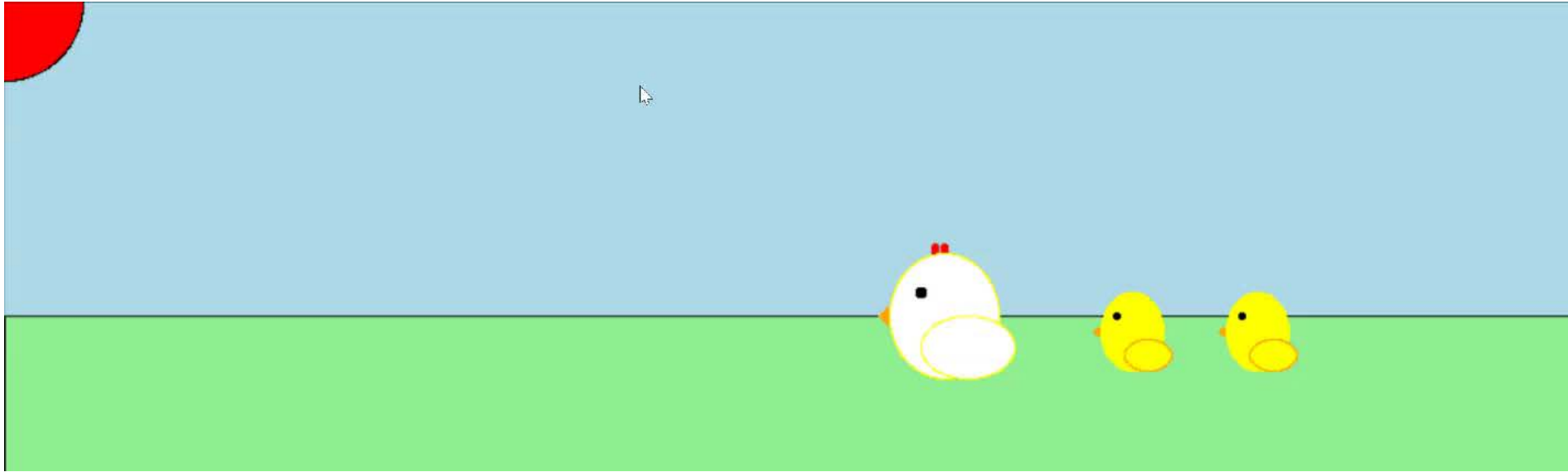
: 튜플, 문자열, 로봇, 이미지, 원이나 사각형 등의 그래픽 객체

객체는 **상태(State)**를 가지고 있고, **동작(Action)**을 수행할 수 있습니다.

Robot: 로봇의 상태는 로봇의 위치, 방향, 들고 있는 비퍼의 갯수 정보를 가지고 있습니다. 로봇은 움직임, 회전, 비퍼 줄기/들어올리기, 그리고 로봇의 상태를 파악하는 동작이 가능합니다.

Circle: 반지름, 위치, 깊이, 외곽선 색, 채워진 색 정보를 상태로 가지고 있습니다. 색 변경, 크기 변경, 위치 변경 등의 동작이 가능합니다.

Picture: 사진의 가로 길이, 세로 길이, 각 픽셀의 색 정보를 상태로 가지고 있습니다. 각 픽셀의 색을 읽거나, 변경하는 동작이 가능합니다.



위 애니메이션은 2010년도 신입생인 유정은, 송금현 학생의 작품입니다.

이 애니메이션은 3개의 레이어 객체를 사용하고 있습니다 (닭, 병아리1, 병아리2).
닭과 병아리는 모두 몸통, 날개, 눈, 부리를 가지고 있습니다.
닭은 벼슬을 가지고 있지만, 병아리는 벼슬이 없습니다.

닭과 병아리의 생김새는 거의 비슷합니다.
닭과 병아리의 차이점은 닭이 더 크고, 하얀색이고, 벼슬을 가지고 있다는 점입니다.

비슷한 객체들을 간단하게 만드는 방법 중 하나는
코드를 작성하고 그 코드를 복사해서 여러 번 붙여 넣는 것입니다.

이 방법은 큰 단점을 가지고 있습니다
: 코드에 버그가 있어서 고쳐야 할 때, 복사했던 모든 코드 역시 고쳐야 합니다.
동일한 코드가 어디에 있는지 파악하기 위해서는 시간과 노력을 많이 기울여야 합니다.

닭을 하나의 클래스로 만들어봅시다.

```
class Chicken(object):  
    """Graphic representation of a chicken."""  
    pass
```

Chicken 객체는 레이어, 몸통, 날개, 눈, 부리를 속성으로 가지게 됩니다.

함수 `make_chicken()` 은 (0,0) 위치에 Chicken 객체를 만듭니다.

```
def make_chicken(hen = False):  
    layer = Layer()  
    if hen:  
        body = Ellipse(70,80)  
        body.setFillColor("white")  
    else:  
        body = Ellipse(40,50)  
        body.setFillColor("yellow")  
        body.move(0, 10)  
    body.setBorderColor("yellow")  
    body.setDepth(20)  
    layer.add(body)  
    # similar for wing, eye, beak, dots
```

함수 `make_chicken()`은 (0,0) 위치에 `Chicken` 객체를 만듭니다.
객체의 생성이 끝나면 해당 객체를 반환합니다.

```
def make_chicken(hen = False):  
    # ... see previous page  
  
    ch = Chicken()  
    ch.layer = layer  
    ch.body = body  
    ch.wing = wing  
    ch.eye = eye  
  
    # return the Chicken object  
    return ch
```

Chicken 객체는 객체의 속성을 (예. `chick1.layer`) 이용해서 움직일 수 있습니다.

```
hen = make_chicken(True)
chick1 = make_chicken()
chick1.layer.move(120, 0)
```

```
herd = Layer()
herd.add(hen.layer)
herd.add(chick1.layer)
herd.move(600, 200)
```

```
chick2 = make_chicken()
chick2.layer.move(800, 200)
```

Chicken 객체도 비슷하게 바꿔봅시다.

객체에 생성자를 추가하고, 객체를 움직이는 멤버 함수를 만들어 봅시다.

```
class Chicken(object):  
    """Graphic representation of a chicken."""  
    def __init__(self, hen = False):  
        layer = Layer()  
        # make all the parts  
        self.layer = layer  
        self.body = body  
        self.wing = wing  
        self.eye = eye  
  
    def move(self, dx, dy):  
        self.layer.move(dx, dy)
```


정리 및 연습

본 강의 학습목표:

- 지금까지 강의에서 사용한 객체의 상태와 동작을 이해할 수 있다.
- 닭과 여러 병아리들이 움직이는 애니메이션을 객체를 사용해 만들 수 있다.

다음 강의 학습 목표:

- 프로그램 실행 속도에 영향을 주는 해석기와 (interpreter) 컴파일러의 차이를 이해할 수 있다.
- 프로그램 실행 속도에 보다 큰 영향을 미치는 좋은 알고리즘의 중요성을 이해할 수 있다.