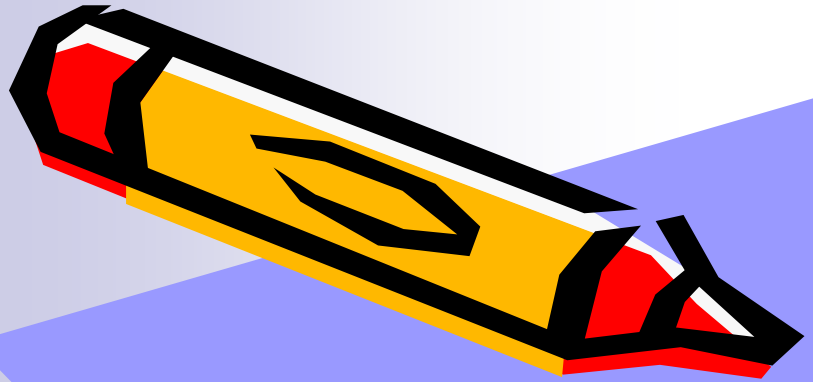


2D 게임 프로그래밍

제6강 인공지능

이대현
한국산업기술대학교

학습 내용

- 리소스 로딩 최적화
- 인공지능의 구현

1000명 전수
리소스 로딩 최적화

Pico2D Canvas (800x600) 9.09 FPS

boys_team_1000.py

초기 로딩하는 시간이
많이 걸린다?

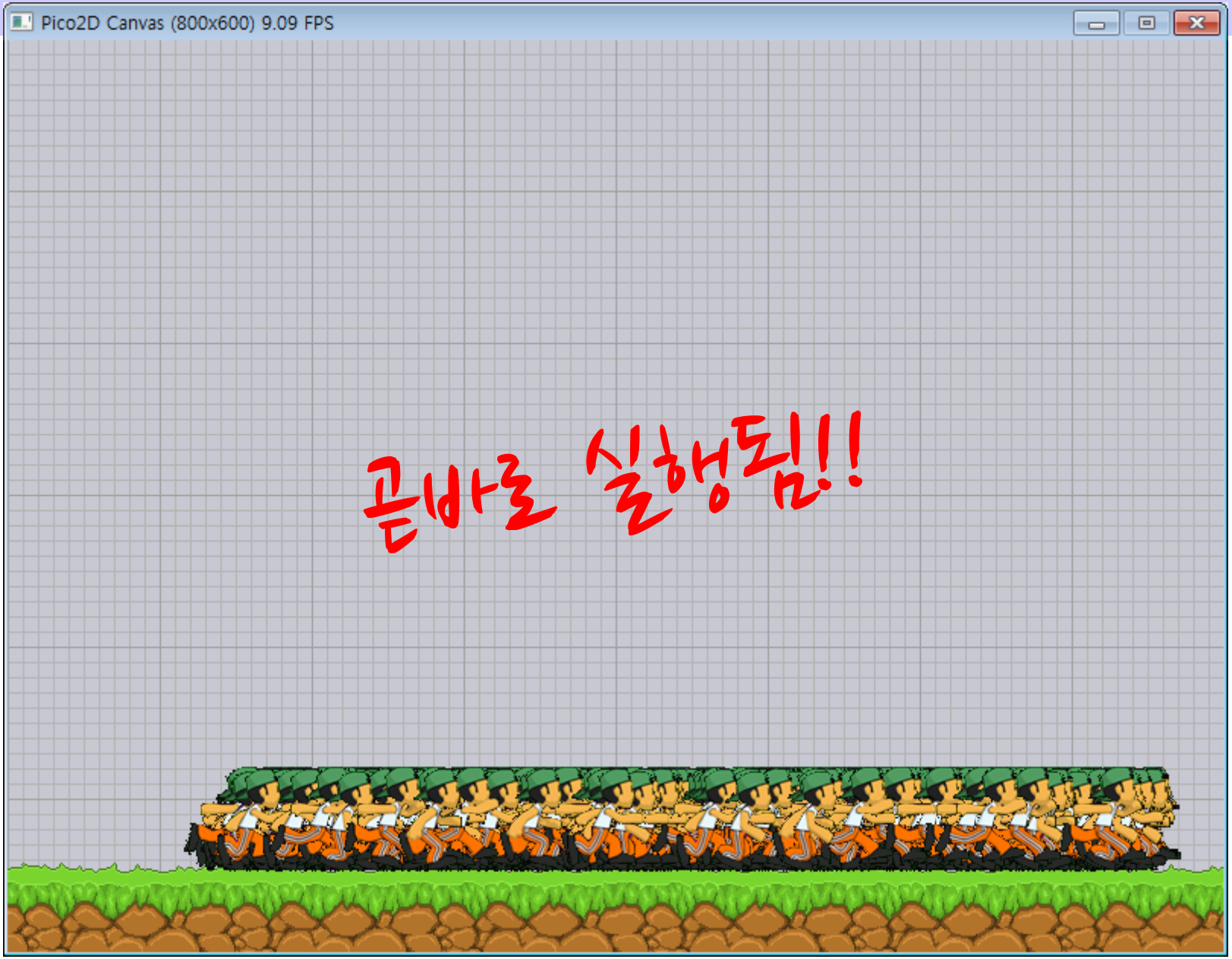
문제점은?

```
class Boy:

    def __init__(self):
        self.x, self.y = random.randint(100, 700), 90
        self.frame = random.randint(0, 7)
        self.image = load_image('run_animation.png')
```

객체의 멤버변수는 객체마다 따로 만들어진다!

1000번의 로딩이 반복


```
class Boy:
    image = None

    def __init__(self):
        self.x, self.y = random.randint(100, 700), 90
        self.frame = random.randint(0, 7)
        if Boy.image == None:
            Boy.image = load_image('run_animation.png')
```



클래스 변수

클래스 자체에 할당되는 변수.
객체들은 공유하는 동일한 변수를 갖게 됨.

```
class Boy:
    image = None
…
…   def __do_some():
…
            Boy.image = …
```

```
class Boy:
    image = None

    def __init__(self):
        self.x, self.y = random.randint(100, 700), 90
        self.frame = random.randint(0, 7)
        if Boy.image == None:
            Boy.image = load_image('run_animation.png')
```

단 한번의 이미지 로딩만 수행.

이미지 리소스를 모든 객체가 공유하게 됨.

소년의 초간단 인공지능 1Q5

ai_boy.py - class Boy


```
def __init__(self):
    self.x, self.y = random.randint(100, 700), 90
    self.frame = random.randint(0, 7)
    self.dir = 1
    if Boy.image == None:
        Boy.image = load_image('run_animation.png')

def update(self):
    self.frame = (self.frame + 1) % 8
    self.x += (self.dir * 5)
    if self.x > 800:
        self.dir = -1
        self.x = 800
    elif self.x < 0:
        self.dir = 1
        self.x = 0
```

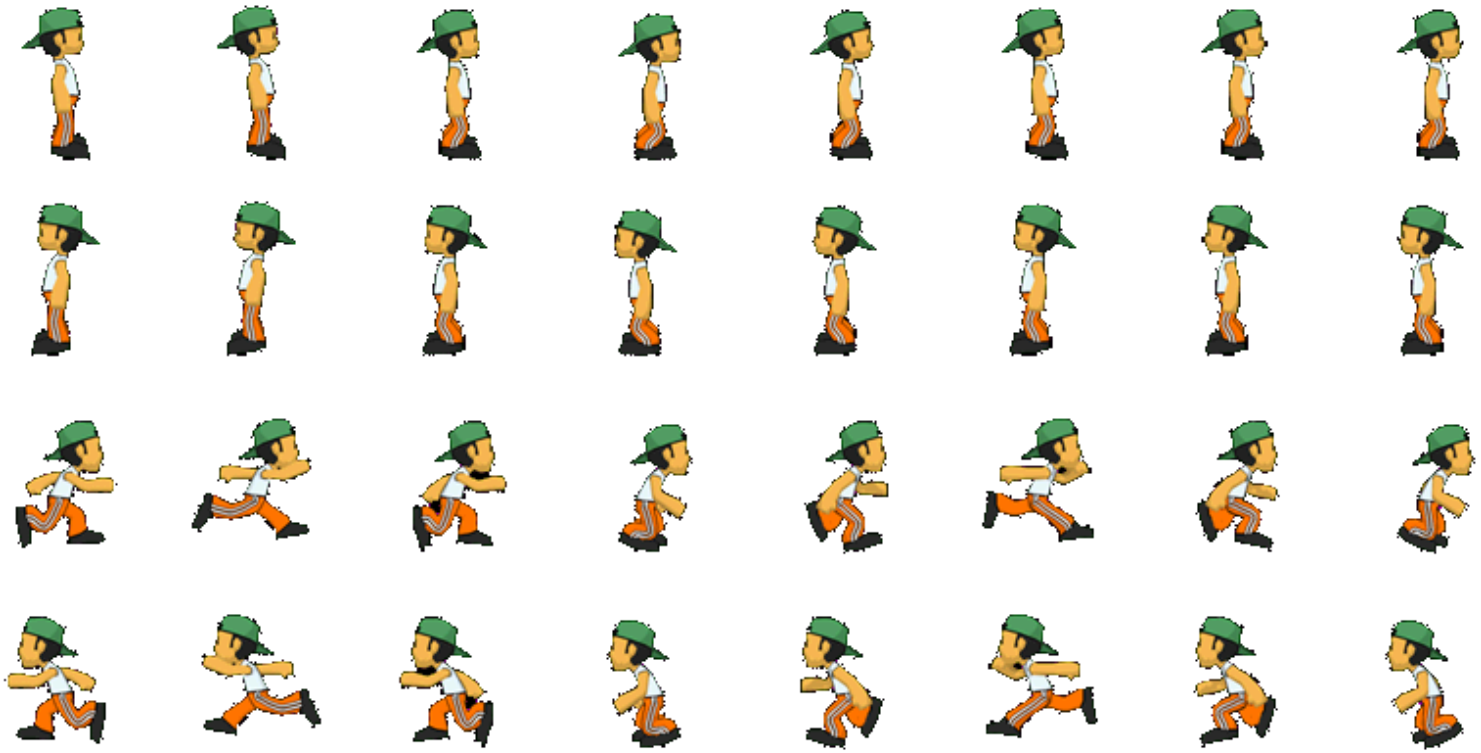
```
def __init__(self):
    self.x, self.y = random.randint(100, 700), 90
    self.frame = random.randint(0, 7)
    self.dir = 1
    if Boy.image == None:
        Boy.image = load_image('run_animation.png‘)
```

dir 이라는 이름의 변수를 이용해, 좌우 방향을 나타내도록 합니다. 1이면 오른쪽, -1이면 왼쪽을 뜻합니다.

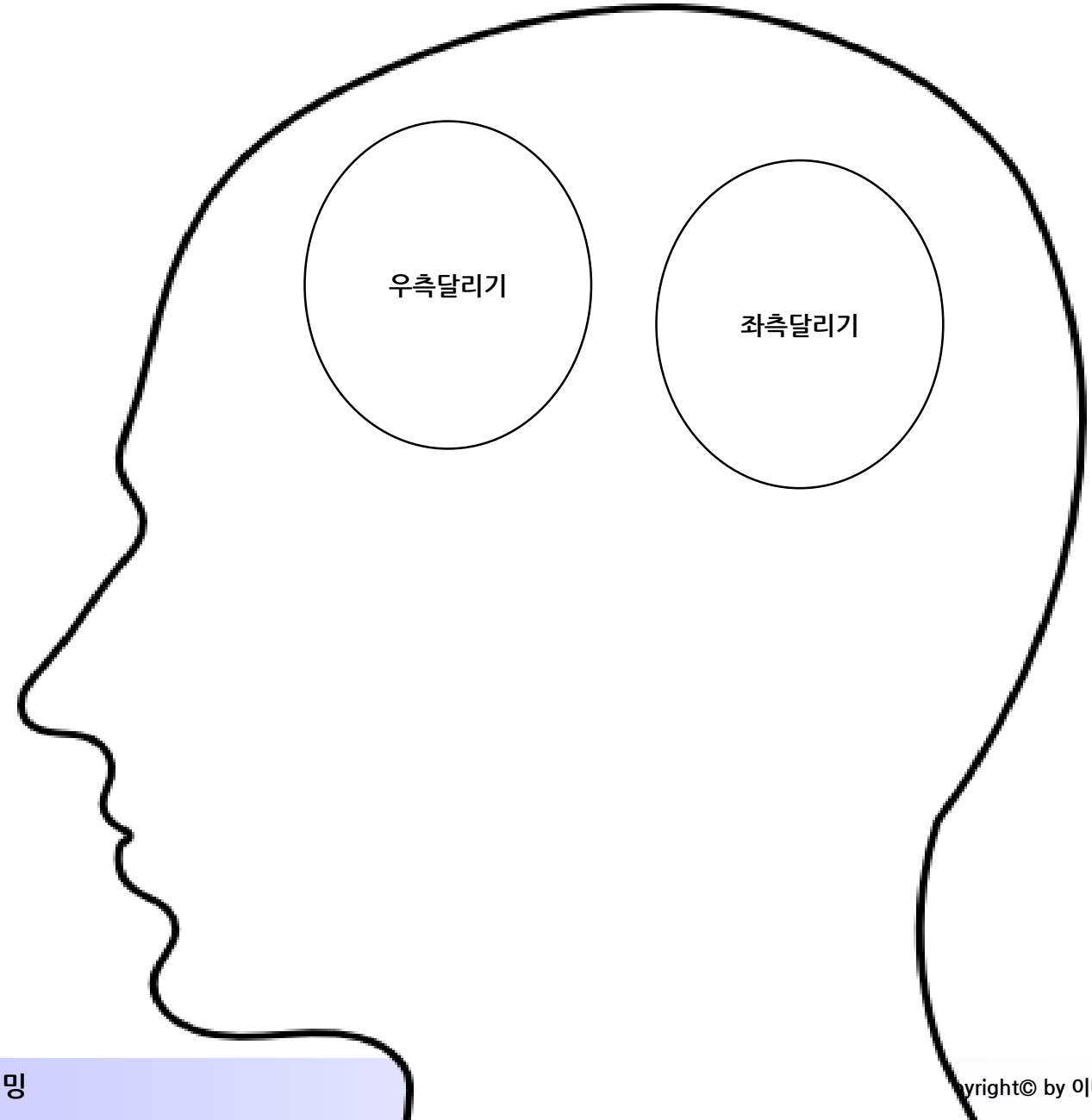
```
    def update(self):
        self.frame = (self.frame + 1) % 8
        self.x += (self.dir * 5)
        if self.x > 800:
            self.dir = -1
            self.x = 800
        elif self.x < 0:
            self.dir = 1
            self.x = 0
```

방향에 따른 x 값의 변화 계산

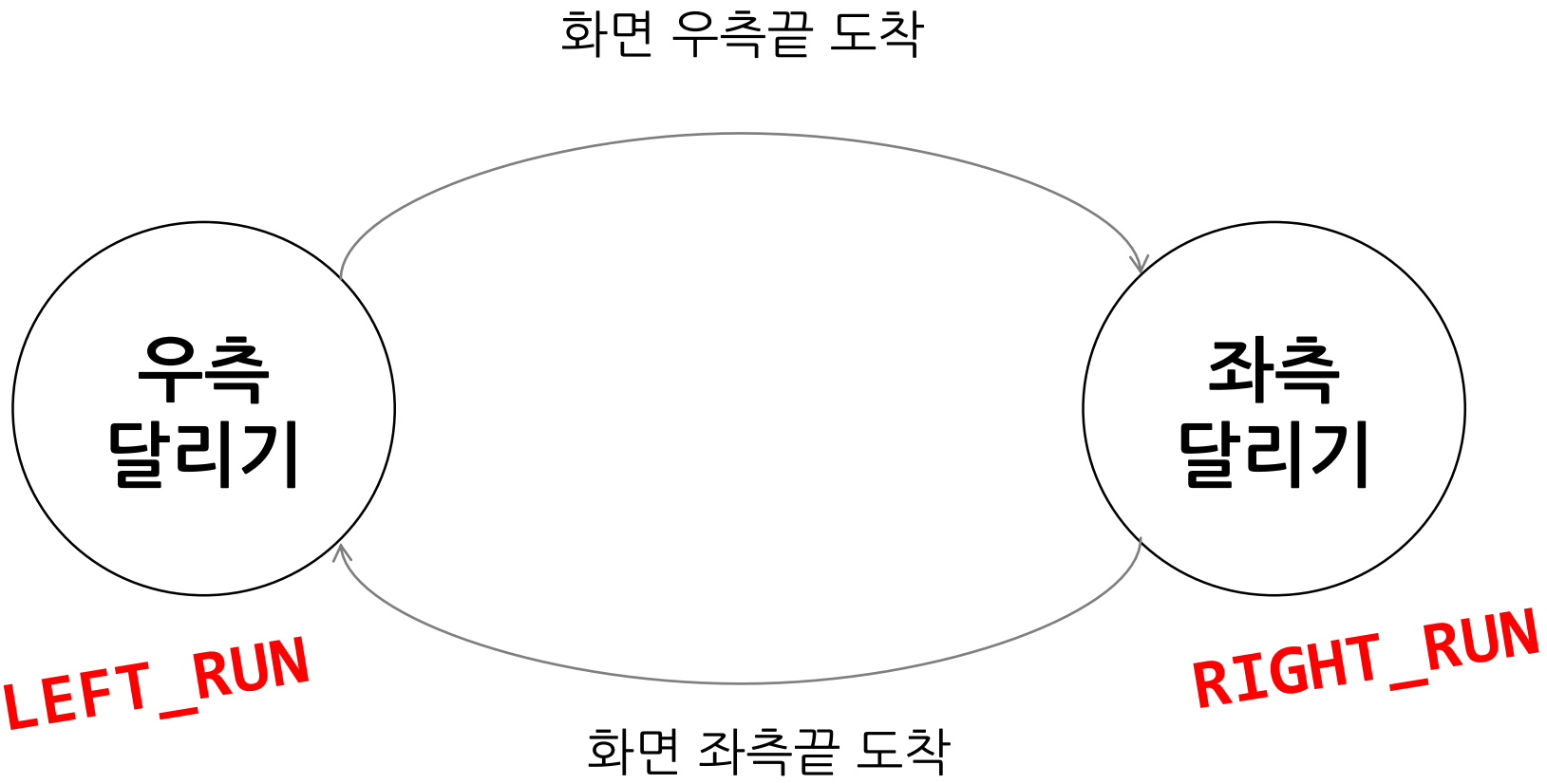
좌측 달리기를 추가할 필요…



소년의 머릿속을 보자…



머릿속을 분석하면…



class Boy 의 변경


```
class Boy:
    image = None

    LEFT_RUN, RIGHT_RUN = 0, 1

    def __init__(self):
        self.x, self.y = random.randint(100, 700), 90
        self.frame = random.randint(0, 7)
        self.dir = 1
        self.state = self.RIGHT_RUN
        if Boy.image == None:
            Boy.image = load_image('animation_sheet.png')
```

class Boy 의 변경


```
    def update(self):
        if self.state == self.RIGHT_RUN:
            self.frame = (self.frame + 1) % 8
            self.x += (self.dir * 5)
        elif self.state == self.LEFT_RUN:
            self.frame = (self.frame + 1) % 8
            self.x += (self.dir * 5)

        if self.x > 800:
            self.dir = -1
            self.x = 800
            self.state = self.LEFT_RUN
        elif self.x < 0:
            self.dir = 1
            self.x = 0
            self.state = self.RIGHT_RUN
```

class Boy 의 변경


```
def draw(self):
    self.image.clip_draw(self.frame * 100, self.state * 100, 100,
                         100, self.x, self.y)
```

상태(State)

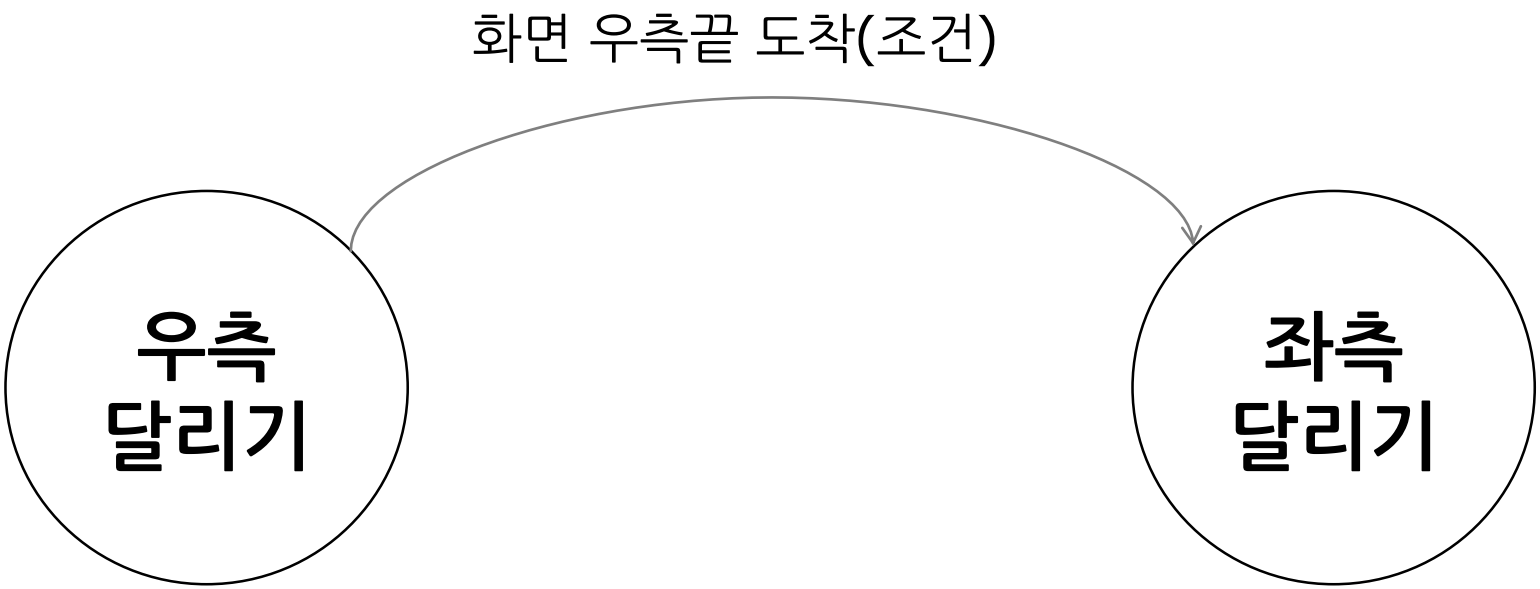
객체가 놓여있는 상황으로써, 요구되는 태스크를 지속적으로 수행함


```
if self.state == self.RIGHT_RUN:
    self.frame = (self.frame + 1) % 8
    self.x += (self.dir * 5)


self.image.clip_draw()
```

RIGHT_RUN

상태의 변화는 언제 일어나는가?

어떤 조건이 만족되거나, 어떤 사건(Event)이 일어날 경우...




```
if self.x > 800:
    self.dir = -1
    self.x = 800
    self.state = self.LEFT_RUN
```

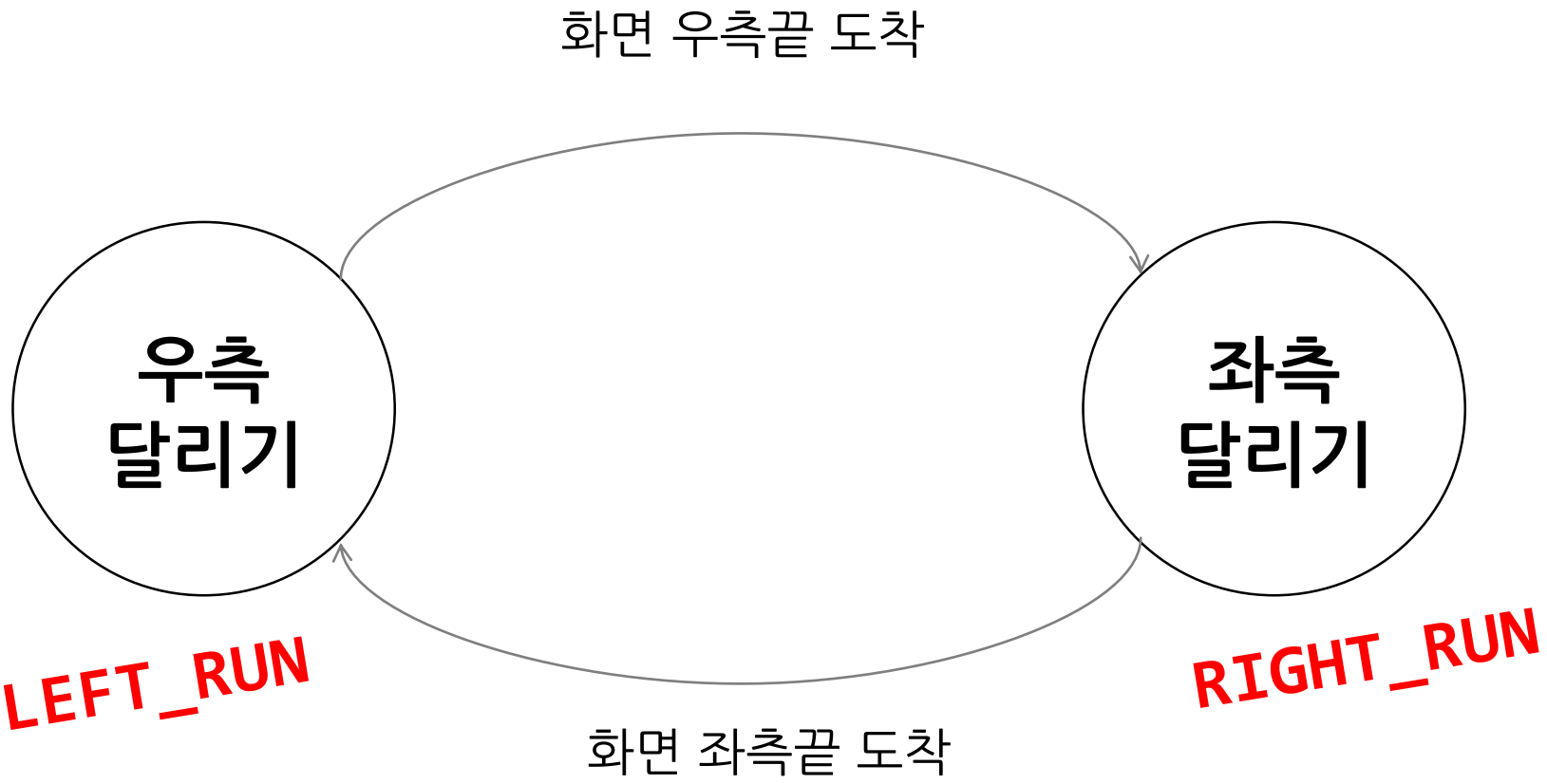
좌측달리기 상태에서는?

```
if self.state == self.LEFT_RUN:
    self.frame = (self.frame + 1) % 8
    self.x += (self.dir * 5)


self.image.clip_draw()
```

LEFT_RUN

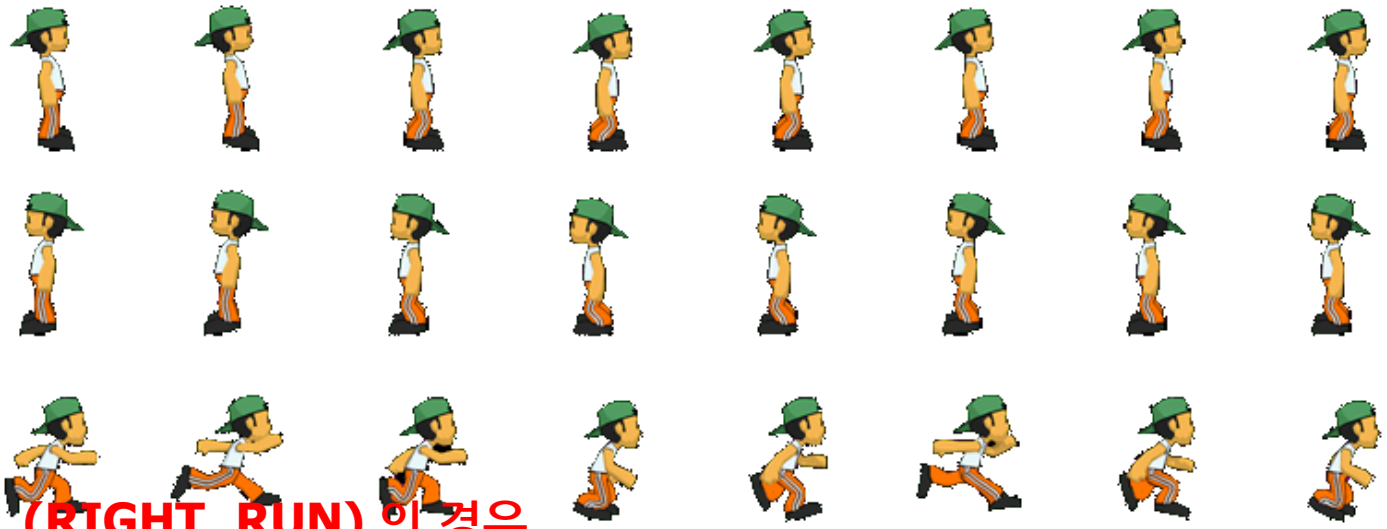
```
elif self.x < 0:
    self.dir = 1
    self.x = 0
    self.state = self.RIGHT_RUN
```

머릿속을 분석하면…



state와 frame에 따른 클립 이미지 선택

(frame*100, state*100)



state == 1 (RIGHT_RUN) 인 경우

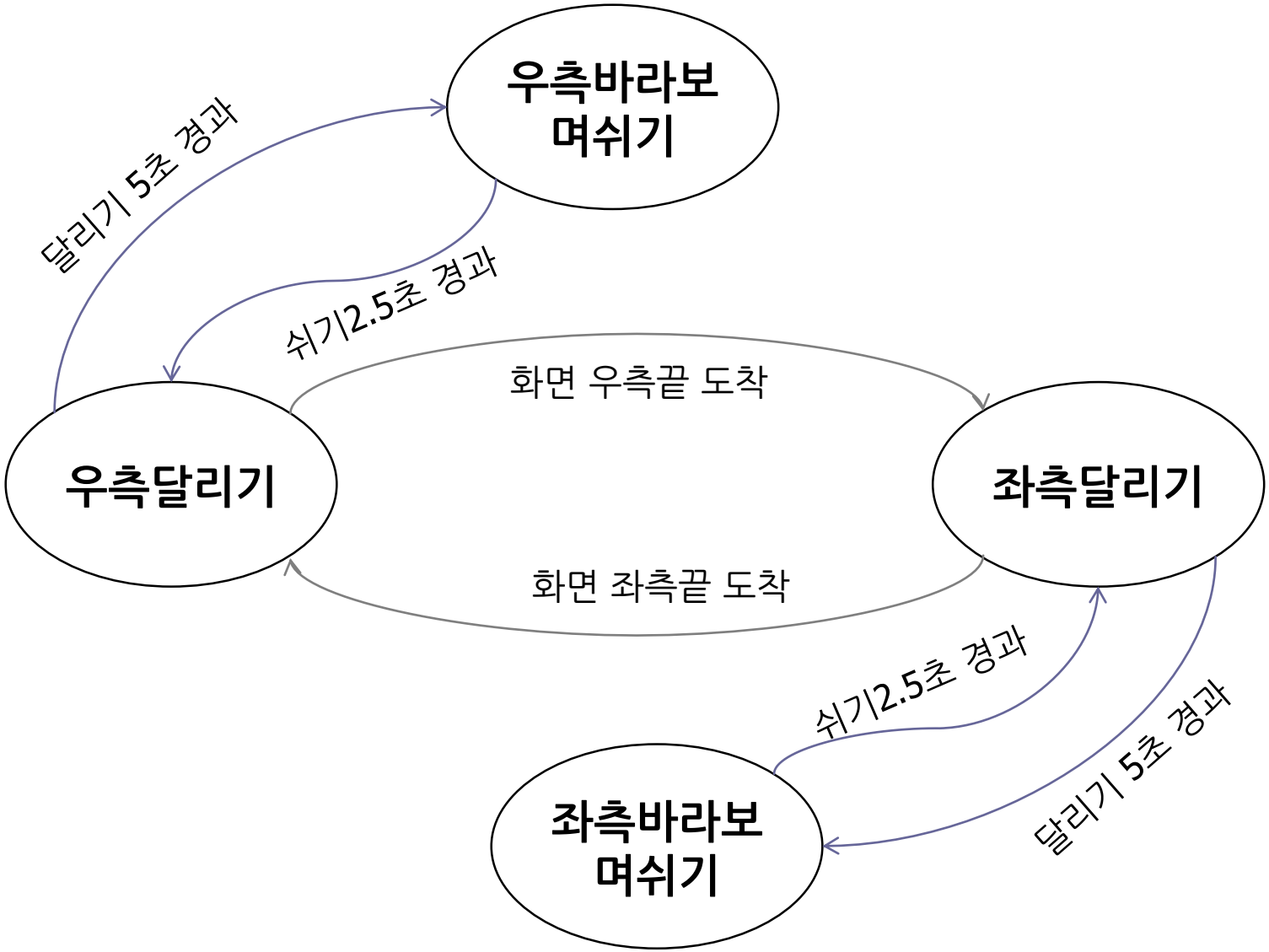
state == 0 (LEFT_RUN) 인 경우

main() 함수의 도입

```
def main():

    open_canvas()
    boy = Boy()
    grass = Grass()

    global running
    running = True
    while running:
        handle_events()

        boy.update()

        clear_canvas()
        grass.draw()
        boy.draw()
        update_canvas()

        delay(0.05)

    close_canvas()


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

1Q-10 인공지능 소년구현

피곤하면 쉬는 소년의 인공 지능



ai_boy_IQ10.py


```
def handle_left_run(self):
    self.x -= 5
    self.run_frames += 1
    if self.x < 0:
        self.state = self.RIGHT_RUN
        self.x = 0
    if self.run_frames == 100:
        self.state = self.LEFT_STAND
        self.stand_frames = 0

def handle_left_stand(self):
    self.stand_frames += 1
    if self.stand_frames == 50:
        self.state = self.LEFT_RUN
        self.run_frames = 0
```

ai_boy_IQ10.py


```
def handle_right_run(self):
    self.x += 5
    self.run_frames += 1
    if self.x > 800:
        self.state = self.LEFT_RUN
        self.x = 800
    if self.run_frames == 100:
        self.state = self.RIGHT_STAND
        self.stand_frames = 0

def handle_right_stand(self):
    self.stand_frames += 1
    if self.stand_frames == 50:
        self.state = self.RIGHT_RUN
        self.run_frames = 0
```

ai_boy_IQ10.py


```
handle_state = {
            LEFT_RUN: handle_left_run,
            RIGHT_RUN: handle_right_run,
            LEFT_STAND: handle_left_stand,
            RIGHT_STAND: handle_right_stand
}

def update(self):
    self.frame = (self.frame + 1) % 8
    self.handle_state[self.state](self)
```

상태

LEFT_RUN,
RIGHT_RUN,
LEFT_STAND,
RIGHT_STAND
= 0, 1, 2, 3

RIGHT_RUN 상태

```
def handle_right_run(self):
        self.x += 5
        self.run_frames += 1
        if self.x > 800:
            self.state = self.LEFT_RUN
            self.x = 800
        if self.run_frames == 100:
            self.state = self.RIGHT_STAND
            self.stand_frames = 0
```

RIGHT_STAND 상태

```
def handle_right_stand(self):
        self.stand_frames += 1
        if self.stand_frames == 50:
            self.state = self.RIGHT_RUN
            self.run_frames = 0
```

함수의 연결

```
handle_state = {
    LEFT_RUN: handle_left_run,
    RIGHT_RUN: handle_right_run,
    LEFT_STAND: handle_left_stand,
    RIGHT_STAND: handle_right_stand
}
```

함수의 실행

```
def update(self):
    self.frame = (self.frame + 1) % 8
    self.handle_state[self.state](self)
```

피곤하면 쉬는 소년…