Lecture #11. 캐릭터 컨트롤러 (2)

2D 게임 프로그래밍

이대현 교수

한국공학대학교
TECH UNIVERSITY OF KOREA

#4. 소년 객체의 전달

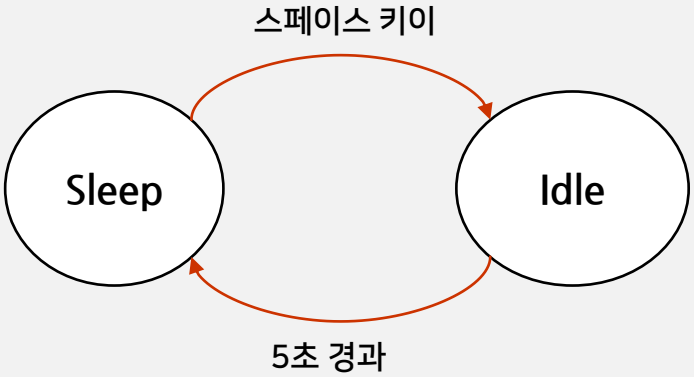

```
class StateMachine:
    def __init__(self, boy):
        self.boy = boy
        self.cur_state = Idle

    def start(self):
        self.cur_state.enter(self.boy)

    def update(self):
        self.cur_state.do(self.boy)

    def draw(self):
        self.cur_state.draw(self.boy)
```

```
class Boy:
    def __init__(self):
        self.x, self.y = 400, 90
        self.frame = 0
        self.action = 3
        self.image = load_image('animation_sheet.png')
        self.state_machine = StateMachine(self)
        self.state_machine.start()
```

```
class Idle:

    @staticmethod
    def enter(boy):
        boy.action = 3
        boy.frame = 0
        pass

    @staticmethod
    def exit(boy):
        pass

    @staticmethod
    def do(boy):
        boy.frame = (boy.frame + 1) % 8

    @staticmethod
    def draw(boy):
        boy.image.clip_draw(boy.frame * 100, boy.action * 100, 100, 100, boy.x, boy.y)
```

캐릭터 컨트롤러 구현(Sleep & Idle)

상태 변환 구현 목표



어떤 상태가 있는가?

어떤 이벤트가 있는가?

Sleep 상태 구현


```
class Sleep:

    @staticmethod
    def enter(boy):
        boy.frame = 0

    @staticmethod
    def exit(boy):
        pass

    @staticmethod
    def do(boy):
        boy.frame = (boy.frame + 1) % 8

    @staticmethod
    def draw(boy):
        boy.image.clip_composite_draw(boy.frame * 100, 300, 100, 100,
                                      3.141592 / 2, '', boy.x - 25, boy.y - 25, 100, 100)
```

상태 이벤트 구현 (1)

- 상태 이벤트는 pico2d 의 get_events 를 통해서 얻는 "입력" 이벤트와는 다름.
  - ex) TIME_OUT – 시간 초과 이벤트 등등이 필요함.

- 튜플을 이용해서 상태 이벤트를 나타내도록 함.
  - (상태 이벤트 종류, 실제 이벤트값)
  - ('INPUT', 실제입력이벤트값)
  - ('TIME_OUT', 0)
  - ('NONE', 0)

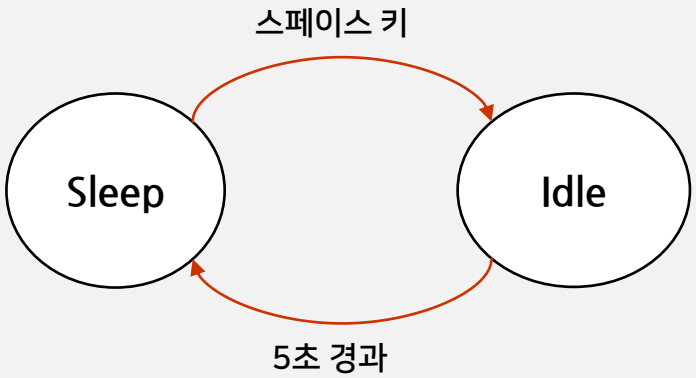
```
class Boy:
    def handle_event(self, event):
        self.state_machine.handle_event(('INPUT', event))
```

상태 이벤트 구현 (2)

- 이벤트 체크 함수를 이용해서 어떤 이벤트인지 판단할 수 있도록 함.

```
def space_down(e):
    return e[0] == 'INPUT' and e[1].type == SDL_KEYDOWN and e[1].key == SDLK_SPACE

def time_out(e):
    return e[0] == 'TIME_OUT'

# 이렇게 쓸수도 있음.
# time_out = lambda e : e[0] == 'TIME_OUT'
```

상태 변환 구현 (1)



현재 상태	이벤트	다음 상태
Sleep	스페이스 키	Idle
Idle	5초 경과	Sleep

상태 변환 테이블

현재상태와 이벤트로부터 다음 상태를 계산

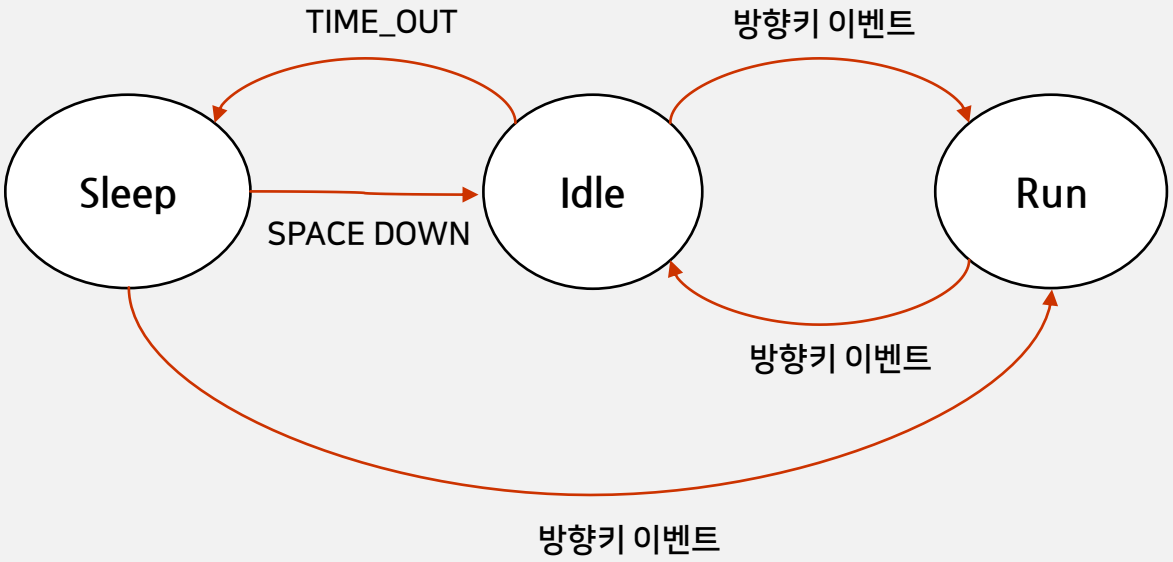
상태 변환 구현 (2)


```
class StateMachine:

    def __init__(self, boy):
        self.boy = boy
        self.cur_state = Sleep
        self.transitions = {
            Sleep: {space_down: Idle},
            Idle: {time_out: Sleep}
        }

    def handle_event(self, e):
        for check_event, next_state in self.transitions[self.cur_state].items():
            if check_event(e):
                self.cur_state.exit(self.boy)
                self.cur_state = next_state
                self.cur_state.enter(self.boy)
                return True
        return False
```

TIME_OUT 이벤트 발생과 처리


```
class Idle:

    @staticmethod
    def enter(boy):
        boy.action = 3
        boy.dir = 0
        boy.frame = 0
        boy.wait_time = get_time() # pico2d import 필요
        pass

    @staticmethod
    def do(boy):
        boy.frame = (boy.frame + 1) % 8
        if get_time() - boy.wait_time > 2:
            boy.state_machine.handle_event(('TIME_OUT', 0))
```

캐릭터 컨트롤러 구현
(Sleep & Idle & Run)



방향키 이벤트 체크 함수


```
def right_down(e):
    return e[0] == 'INPUT' and e[1].type == SDL_KEYDOWN and e[1].key == SDLK_RIGHT


def right_up(e):
    return e[0] == 'INPUT' and e[1].type == SDL_KEYUP and e[1].key == SDLK_RIGHT


def left_down(e):
    return e[0] == 'INPUT' and e[1].type == SDL_KEYDOWN and e[1].key == SDLK_LEFT


def left_up(e):
    return e[0] == 'INPUT' and e[1].type == SDL_KEYUP and e[1].key == SDLK_LEFT
```

Run 상태 구현


```
class Run:

    @staticmethod
    def enter(boy, e):
        if right_down(e) or left_up(e): # 오른쪽으로 RUN
            boy.dir, boy.action = 1, 1
        elif left_down(e) or right_up(e): # 왼쪽으로 RUN
            boy.dir, boy.action = -1, 0

    @staticmethod
    def exit(boy, e):
        pass

    @staticmethod
    def do(boy):
        boy.frame = (boy.frame + 1) % 8
        boy.x += boy.dir * 5
        pass

    @staticmethod
    def draw(boy):
        boy.image.clip_draw(boy.frame * 100, boy.action * 100, 100, 100, boy.x, boy.y)
```

상태 변환 규칙

```
class StateMachine:
    def __init__(self, boy):
        self.boy = boy
        self.cur_state = Idle
        self.transitions = {
            Idle: {right_down: Run, left_down: Run, left_up: Run, right_up: Run, time_out: Sleep},
            Run: {right_down: Idle, left_down: Idle, right_up: Idle, left_up: Idle},
            Sleep: {right_down: Run, left_down: Run, right_up: Run, left_up: Run, space_down: Idle}
        }
```

enter, exit 액션에서 이벤트 전달 추가

```
class StateMachine:

    def start(self):
        self.cur_state.enter(self.boy, ('NONE', 0))


    def handle_event(self, e):
        for check_event, next_state in self.transitions[self.cur_state].items():
            if check_event(e):
                self.cur_state.exit(self.boy, e)
                self.cur_state = next_state
                self.cur_state.enter(self.boy, e)
                return True

        return False
```

Sleep 상태 수정


```
class Sleep:

    @staticmethod
    def enter(boy, e):
        boy.frame = 0
        pass

    @staticmethod
    def exit(boy, e):
        pass

    @staticmethod
    def do(boy):
        boy.frame = (boy.frame + 1) % 8

    @staticmethod
    def draw(boy):
        if boy.action == 2:
            boy.image.clip_composite_draw(boy.frame * 100, 200, 100, 100,
                                          -3.141592 / 2, '', boy.x + 25, boy.y - 25, 100, 100)
        else:
            boy.image.clip_composite_draw(boy.frame * 100, 300, 100, 100,
                                          3.141592 / 2, '', boy.x - 25, boy.y - 25, 100, 100)
```

enter, exit 에 이벤트 전달 추가

Idle 상태에서 캐릭터의 방향에 맞게 회전 구현.

Idle 상태 수정

```
class Idle:

    @staticmethod
    def enter(boy, e):
        if boy.action == 0:
            boy.action = 2
        elif boy.action == 1:
            boy.action = 3
        boy.dir = 0
        boy.frame = 0
        boy.wait_time = get_time() # pico2d import 필요
        pass

    @staticmethod
    def exit(boy, e):
        pass
```

이전에 달리고 있는 상황이었으면, 그 때 이동 방향을 반영해서 정지 상태의 방향 결정.