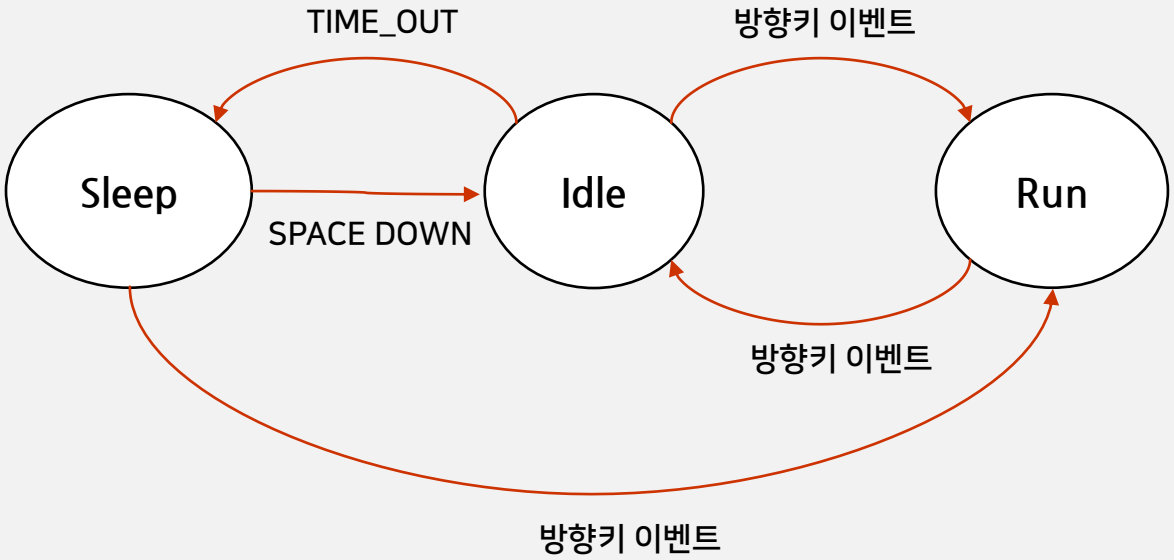
Lecture #12. 캐릭터 컨트롤러 (3)

2D 게임 프로그래밍

이대현 교수

한국공학대학교
TECH UNIVERSITY OF KOREA

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


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

```
self.state_machine.set_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

```
def start(self, state):
    self.cur_state = state
    self.cur_state.enter(self.o, ('START', 0))
    pass
```

시작 상태로 들어갈 때는, START 이벤트에 의해 상태 변화가 일어난 것으로 처리.

```
def handle_event(self, e):
    for event, next_state in self.transitions[self.cur_state].items():
        if event(e):
            self.cur_state.exit(self.o, e)
            self.cur_state = next_state
            self.cur_state.enter(self.o, e)
            return
```

exit와 enter에 e를 전달해줌. 상태 변화의 원인인 e를 알려줌으로써 필요한 작업을 할 수 있도록 하기 위함.

Sleep 상태 수정


```
class Sleep:
    @staticmethod
    def enter(boy, e):
        if start_event(e):
            boy.face_dir = 1
            boy.action = 3
        boy.frame = 0

    @staticmethod
    def exit(boy, e):
        pass

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

    @staticmethod
    def draw(boy):
        if boy.face_dir == 1:
            boy.image.clip_composite_draw(boy.frame * 100, 300, 100, 100,
                                          3.141592 / 2, '', boy.x - 25, boy.y - 25, 100, 100)
        else:
            boy.image.clip_composite_draw(boy.frame * 100, 200, 100, 100,
```

enter, exit 에 이벤트 전달 추가

Idle 상태에서 캐릭터의 방향에 맞게 회전 구현.

Idle 상태 수정

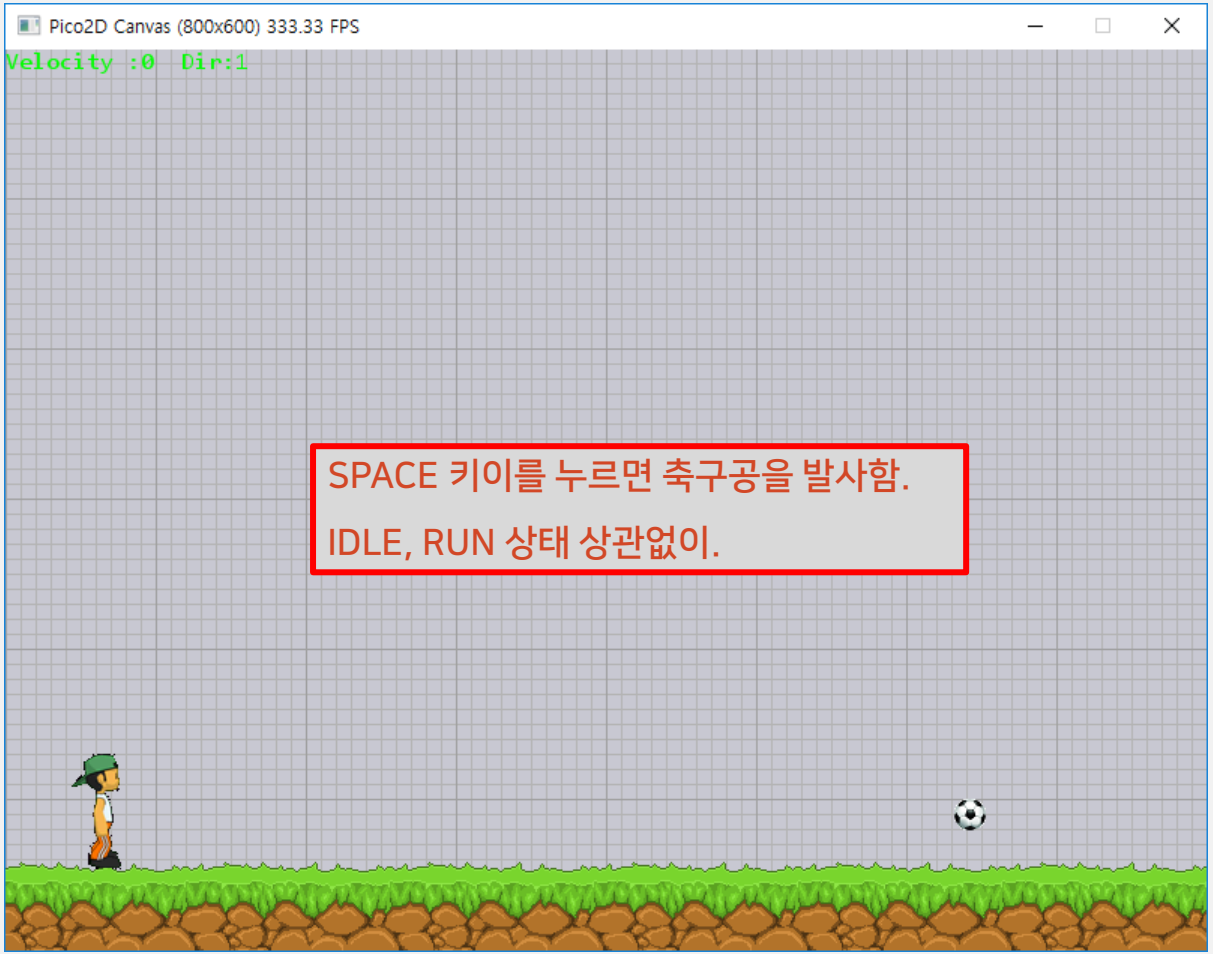
```
class Idle:
    @staticmethod
    def enter(boy, e):
        if start_event(e):
            boy.action = 2
            boy.face_dir = -1
        elif right_down(e) or left_up(e):
            boy.action = 2
            boy.face_dir = -1
        elif left_down(e) or right_up(e):
            boy.action = 3
            boy.face_dir = 1

        boy.frame = 0
        boy.wait_time = get_time()
```

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

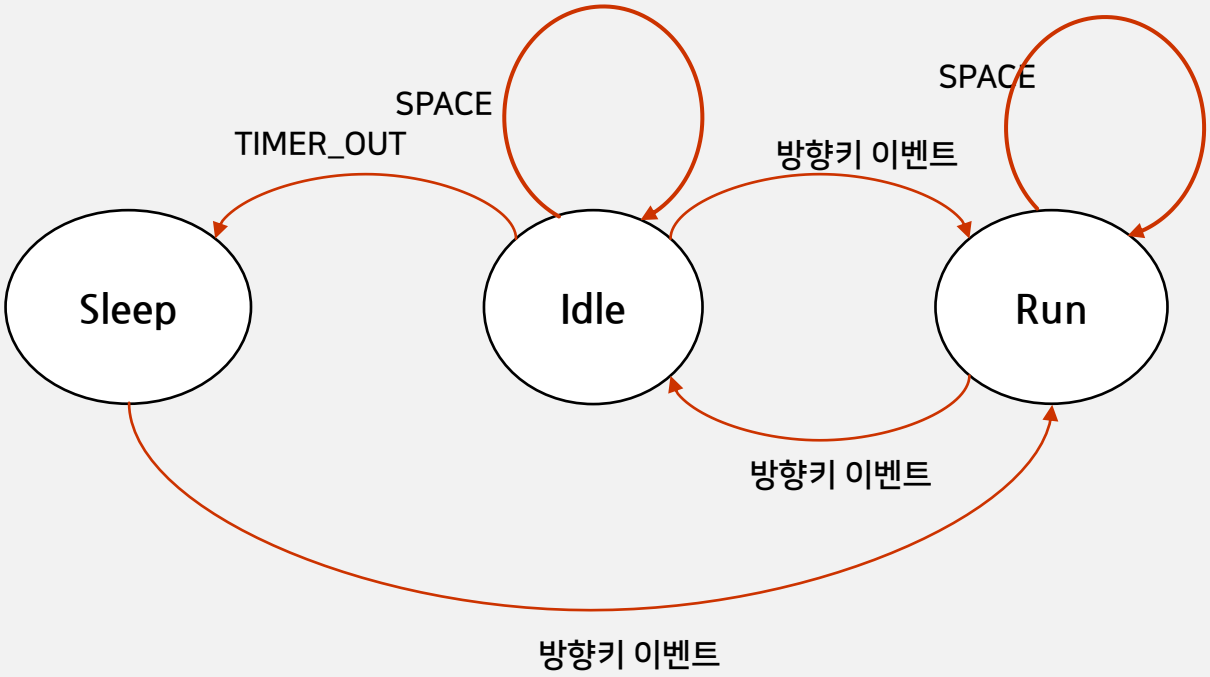
디버깅 Tips

```
class StateMachine:

    def start(self, state):
        self.cur_state = state
        print(f'Enter into {state}')
        self.cur_state.enter(self.o, ('START', 0))

    def add_event(self, e):
        print(f'    DEBUG: New event {e} added to event Que')
        self.event_que.append(e)

    def handle_event(self, e):
        for event, next_state in self.transitions[self.cur_state].items():
            if event(e):
                print(f'Exit from {self.cur_state}')
                self.cur_state.exit(self.o, e)
                self.cur_state = next_state
                print(f'Enter into {self.cur_state}')
                self.cur_state.enter(self.o, e)
                return

        print(f'        WARN: Event [{e}] at State [{self.cur_state}] not handled')
```

소년의 축구공 발사



상태 다이어그램



boy.py – 상태 변화 추가


```
self.state_machine.set_transitions(
    {
        Idle: {right_down: Run, left_down: Run, left_up: Run, right_up: Run, time_out: Sleep, space_down: Idle},
        Run: {right_down: Idle, left_down: Idle, right_up: Idle, left_up: Idle, space_down: Run},
        Sleep: {right_down: Run, left_down: Run, right_up: Run, left_up: Run, space_down: Idle}
    }
)
```

boy.py – boy 의 fire_ball 함수 추가


```
def fire_ball(self):
    if self.face_dir == -1:
        print('FIRE BALL LEFT')
    elif self.face_dir == 1:
        print('FIRE BALL RIGHT')
```

boy.py – RunState, IdleState의 exit() 함수 조정


```
class Idle:

    @staticmethod
    def exit(boy, e):
        if space_down(e):
            boy.fire_ball()
```

```
class Run:

    @staticmethod
    def exit(boy, e):
        if space_down(e):
            boy.fire_ball()
```