

Lecture #13. 게임 월드

2D 게임 프로그래밍

이대현 교수



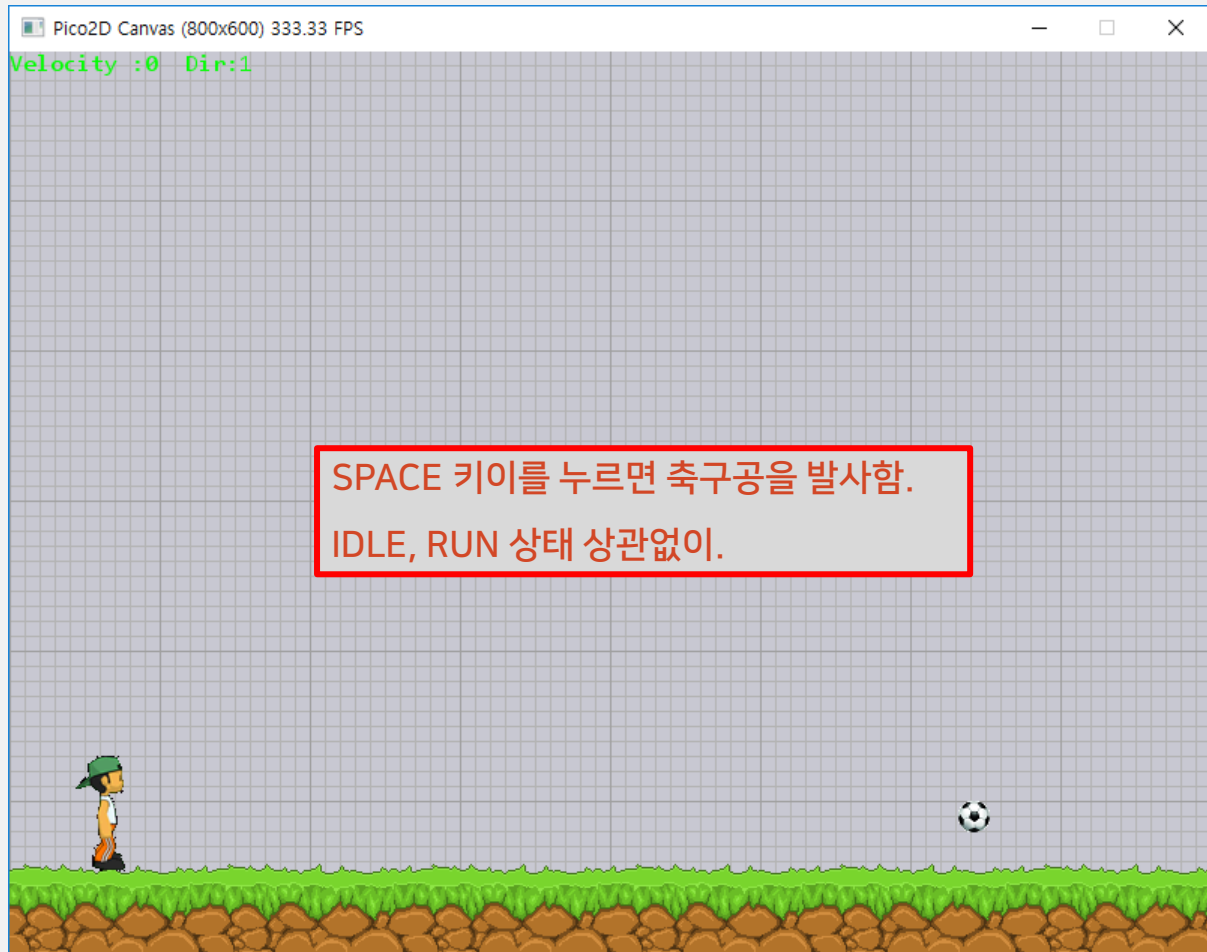
한국공학대학교
TECH UNIVERSITY OF KOREA

학습 내용

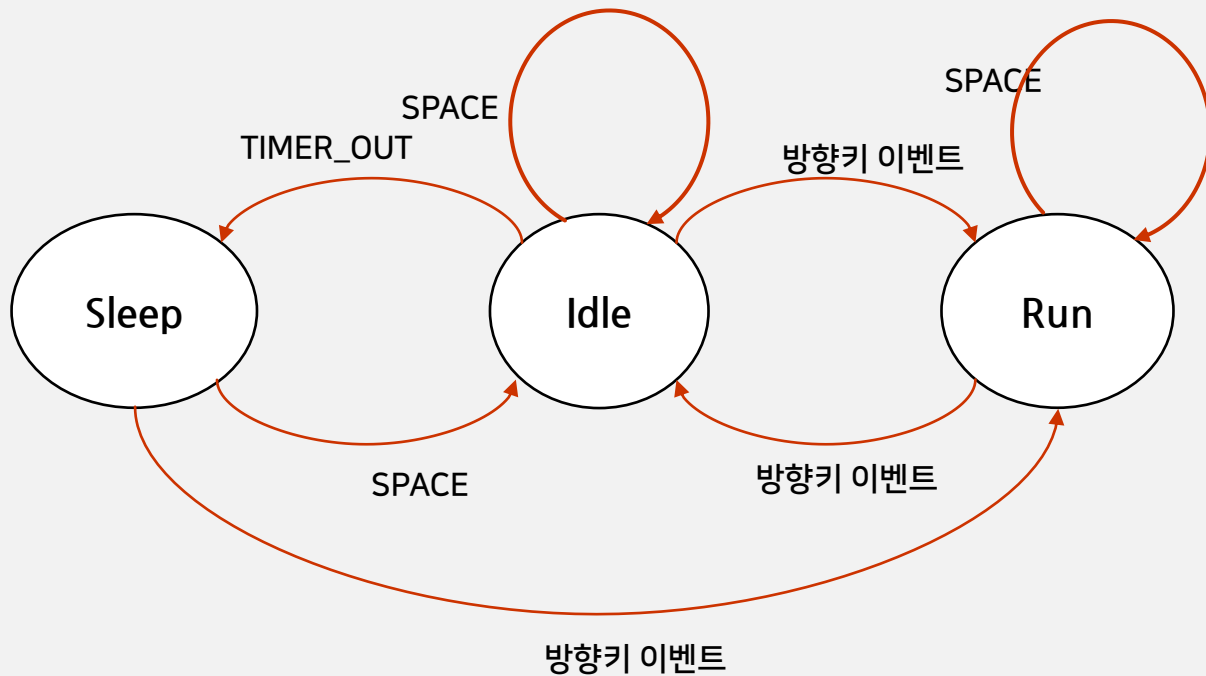
- 게임 월드 구성



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()
```




간단한 게임 월드 구현

ball.py

```
from pico2d import load_image

class Ball:
    image = None

    def __init__(self, x = 400, y = 300, velocity = 1):
        if Ball.image == None:
            Ball.image = load_image('ball21x21.png')
        self.x, self.y, self.velocity = x, y, velocity

    def draw(self):
        self.image.draw(self.x, self.y)

    def update(self):
        self.x += self.velocity
```

ball 오브젝트 생성과 추가



```
def fire_ball(self):  
    ball = Ball(self.x, self.y, self.face_dir*10)  
    game_world.add_object(ball)
```



```
world = []

def add_object(o):
    world.append(o)

def update():
    for o in world:
        o.update()

def render():
    for o in world:
        o.draw()
```

game_world 의 활용



```
def create_world():  
    global running  
    global grass  
    global team  
    global boy  
  
    running = True  
  
    grass = Grass()  
    game_world.add_object(grass)  
  
    boy = Boy()  
    game_world.add_object(boy)
```

```
def update_world():  
    game_world.update()  
  
def render_world():  
    clear_canvas()  
    game_world.render()  
    update_canvas()
```



ball.py

```
def update(self):  
    self.x += self.velocity  
  
    if self.x < 25 or self.x > 800 - 25:  
        game_world.remove_object(self)
```

game_world.py

```
def remove_object(o):  
    world.remove(o)
```



본격 게임 월드 구현

game_world.py (1)



게임 월드에 담겨있는 모든 객체들을 담고 있는 리스트. Drawing Layer 에 따라서 분류. 필요에 따라 Layer를 추가하면 됨. 현재는 두개의 Layer만.

```
# layer 0: Background Objects
# layer 1: Foreground Objects

world = [[], []]
```

```
def add_object(o, depth = 0):
    world[depth].append(o)
```

게임 월드에 객체 추가

```
def add_objects(oL, depth = 0):
    world[depth] += oL
```

게임 월드에 객체'들'을 추가

```
def remove_object(o):
    for layer in world:
        if o in layer:
            layer.remove(o)
            return
```

게임 월드에서 객체 제거

```
raise ValueError('Cannot delete non existing object')
```


game_world.py (2)



```
def update():  
    for layer in world:  
        for o in layer:  
            o.update()  
  
def render():  
    for layer in world:  
        for o in layer:  
            o.draw()
```

게임 월드 depth 를 이용하도록 수정



control_boy.py

```
def create_world():  
  
    # 중략...  
  
    grass = Grass()  
    game_world.add_object(grass, 0)  
  
    boy = Boy()  
    game_world.add_object(boy, 1)
```

ball.py

```
def fire_ball(self):  
    ball = Ball(self.x, self.y, self.face_dir*10)  
    game_world.add_object(ball, 1)
```

