

# Lecture #12. 게임 월드

2D 게임 프로그래밍

이대현 교수

# 학습 내용

---

- 캐릭터 상태의 추가
- 특수 이벤트 처리
- 게임 월드 구성

시스템



캐릭터 컨트롤러 구현

(IDLE & RUN & SLEEP)

# boy.py – SLEEP\_TIMER 이벤트 추가



*# Boy Event*

RIGHT\_DOWN, LEFT\_DOWN, RIGHT\_UP, LEFT\_UP, **SLEEP\_TIMER = range(5)**

```
key_event_table = {  
    (SDL_KEYDOWN, SDLK_RIGHT): RIGHT_DOWN,  
    (SDL_KEYDOWN, SDLK_LEFT): LEFT_DOWN,  
    (SDL_KEYUP, SDLK_RIGHT): RIGHT_UP,  
    (SDL_KEYUP, SDLK_LEFT): LEFT_UP  
}
```

# boy.py – SLEEP 상태 함수 추가



```
class SleepState:
```

```
    @staticmethod
```

```
    def enter(boy, event):
        boy.frame = 0
```

```
    @staticmethod
```

```
    def exit(boy, event):
        pass
```

```
    @staticmethod
```

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

```
    @staticmethod
```

```
    def draw(boy):
        if boy.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,
                                           -3.141592 / 2, '', boy.x + 25, boy.y - 25, 100, 100)
```

# boy.py – Sleep 상태 변화 추가



```
next_state_table = {  
    IdleState: {RIGHT_UP: RunState, LEFT_UP: RunState,  
                RIGHT_DOWN: RunState, LEFT_DOWN: RunState,  
                SLEEP_TIMER: SleepState},  
    RunState: {RIGHT_UP: IdleState, LEFT_UP: IdleState,  
              LEFT_DOWN: IdleState, RIGHT_DOWN: IdleState},  
    SleepState: {LEFT_DOWN: RunState, RIGHT_DOWN: RunState,  
                LEFT_UP: RunState, RIGHT_UP: RunState}  
}
```

# boy.py – IdleState SLEEP\_TIMER 이벤트 처리



```
class IdleState:
```

```
    @staticmethod
```

```
    def do(boy):
```

```
        boy.frame = (boy.frame + 1) % 8
```

```
        boy.timer -= 1
```

```
        if boy.timer == 0:
```

```
            boy.add_event(SLEEP_TIMER)
```

`clip_composite_draw(left, bottom, width, height, rad, flip, x, y, w, h)`

rad: 회전각도(라디안값)

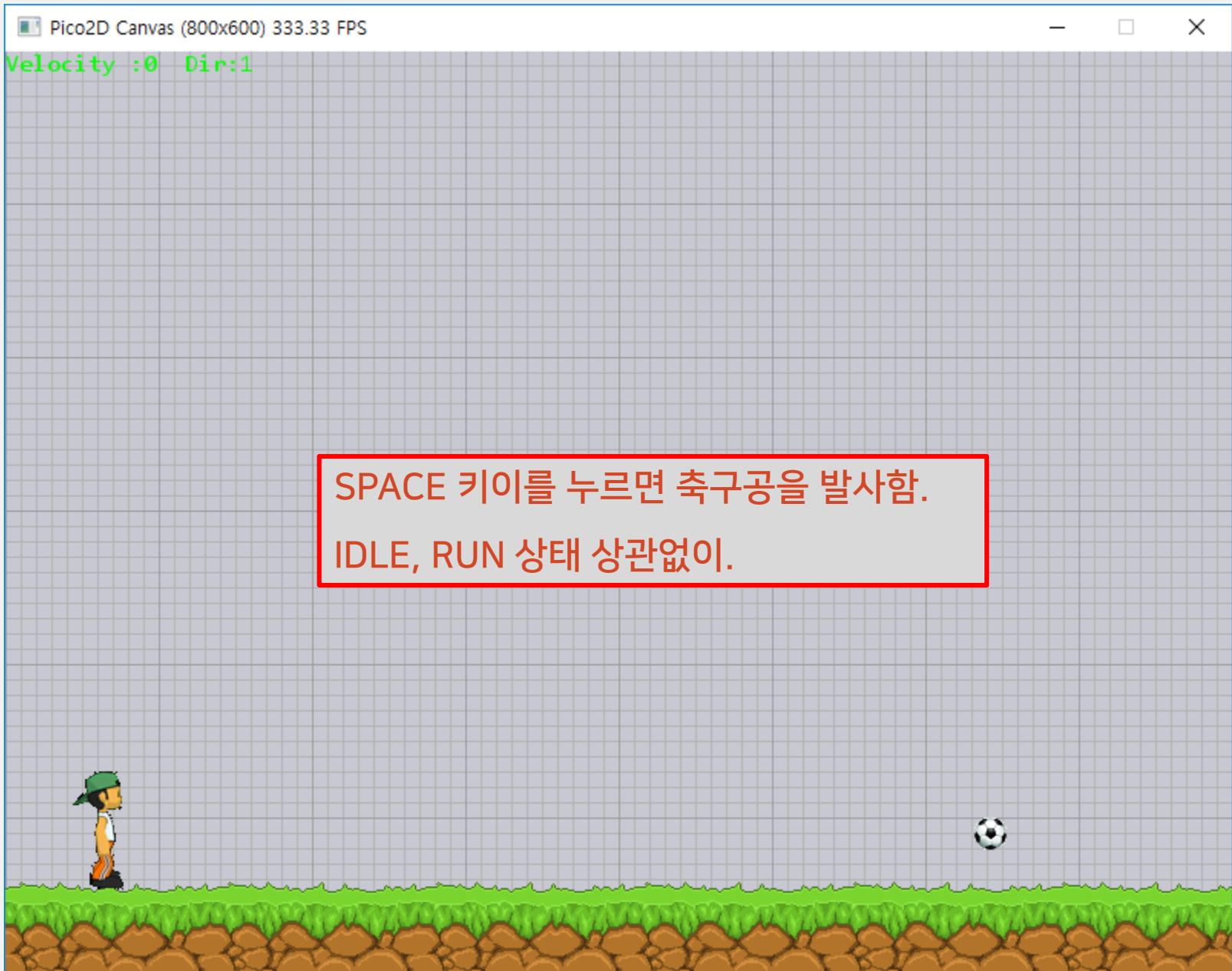
flip: 반전여부('h': 상하반전, 'v':좌우반전, 'hv': 상하좌우반전)



실습

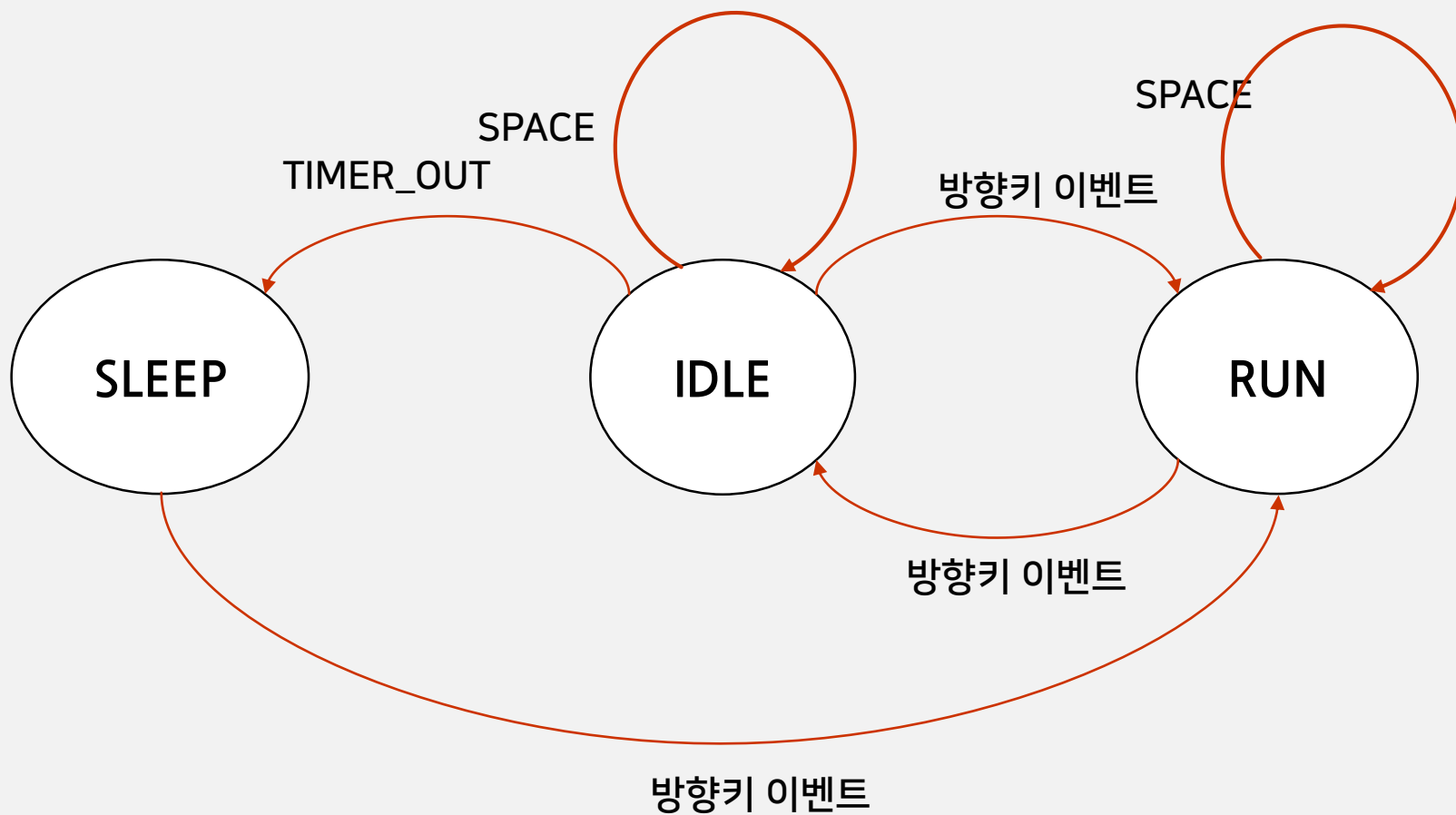


## 소년의 축구공 발사



# 상태 다이어그램

---



# boy.py – SPACE 이벤트 추가



```
# Boy Event
```

```
RIGHT_DOWN, LEFT_DOWN, RIGHT_UP, LEFT_UP, SLEEP_TIMER, SPACE = range(6)
```

```
key_event_table = {  
    (SDL_KEYDOWN, SDLK_RIGHT): RIGHT_DOWN,  
    (SDL_KEYDOWN, SDLK_LEFT): LEFT_DOWN,  
    (SDL_KEYUP, SDLK_RIGHT): RIGHT_UP,  
    (SDL_KEYUP, SDLK_LEFT): LEFT_UP,  
    (SDL_KEYDOWN, SDLK_SPACE): SPACE  
}
```



```
next_state_table = {  
    IdleState: {RIGHT_UP: RunState, LEFT_UP: RunState,  
                RIGHT_DOWN: RunState, LEFT_DOWN: RunState,  
                SLEEP_TIMER: SleepState, SPACE: IdleState},  
    RunState: {RIGHT_UP: IdleState, LEFT_UP: IdleState,  
               LEFT_DOWN: IdleState, RIGHT_DOWN: IdleState,  
               SPACE: RunState},  
    SleepState: {LEFT_DOWN: RunState, RIGHT_DOWN: RunState,  
                 LEFT_UP: RunState, RIGHT_UP: RunState}  
}
```

## boy.py – boy 의 fire\_ball 함수 추가



```
def fire_ball(self):  
    print('FIRE BALL')
```

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



```
class IdleState:

    @staticmethod
    def exit(boy, event):
        if event == SPACE:
            boy.fire_ball()
```

```
class RunState:

    @staticmethod
    def exit(boy, event):
        if event == SPACE:
            boy.fire_ball()
```

# 실행하고 SPACE 를 눌러보자?

---





```
next_state_table = {
    IdleState: {RIGHT_UP: RunState, LEFT_UP: RunState,
                RIGHT_DOWN: RunState, LEFT_DOWN: RunState,
                SLEEP_TIMER: SleepState, SPACE: IdleState},
    RunState: {RIGHT_UP: IdleState, LEFT_UP: IdleState,
              LEFT_DOWN: IdleState, RIGHT_DOWN: IdleState,
              SPACE: RunState},
    SleepState: {LEFT_DOWN: RunState, RIGHT_DOWN: RunState,
                LEFT_UP: RunState, RIGHT_UP: RunState,
                SPACE: IdleState}
}
```

# ball.py

---

```
from pico2d import *
import game_world

class Ball:
    image = None

    def __init__(self, x = 800, 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
```

# 게임 월드 game\_world.py

---

```
# Layer 0: Background Objects  
# Layer 1: Foreground Objects  
objects = [[],[]]
```

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

```
def add_object(o, layer):  
    objects[layer].append(o)
```

게임 월드에 객체 추가

```
def add_objects(l, layer):  
    objects[layer] += l
```

게임 월드에 객체들을 추가

# 게임 월드 game\_world.py

---

```
def remove_object(o):  
    for i in range(len(objects)):  
        if o in objects[i]:  
            objects[i].remove(o)  
            del o  
            break
```

게임 월드에서 객체 제거

```
def clear():  
    for o in all_objects():  
        del o
```

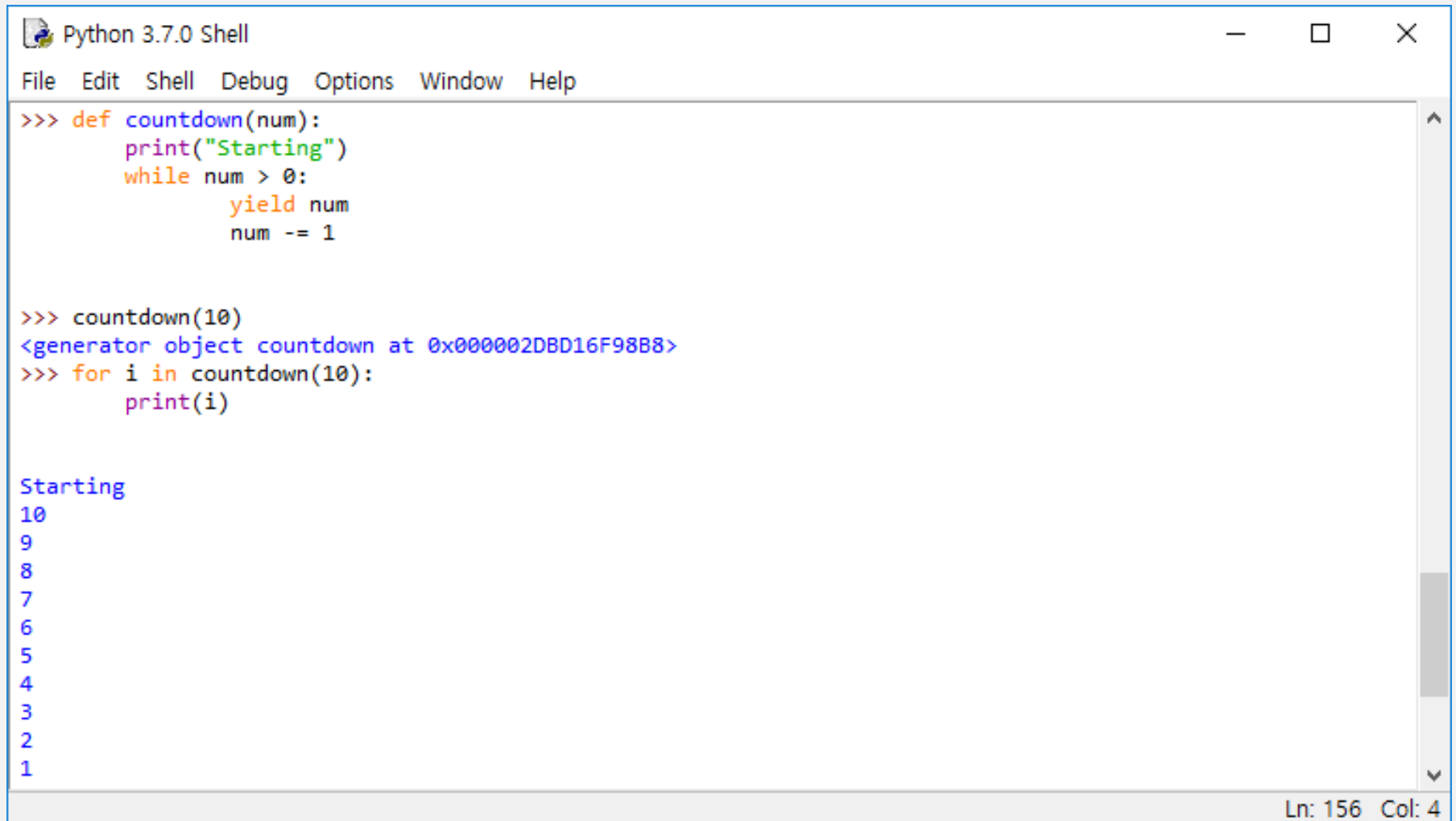
게임 월드의 모든 객체 제거

```
def all_objects():  
    for i in range(len(objects)):  
        for o in objects[i]:  
            yield o
```

게임 월드의 모든 객체들을 하나씩 꺼내오기

# Python Generator

- 객체들을 하나씩 만들어서(발전) 넘겨주는 기능
- for 문 등에서 효과적으로 사용.



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

>>> def countdown(num):
    print("Starting")
    while num > 0:
        yield num
        num -= 1

>>> countdown(10)
<generator object countdown at 0x000002DBD16F98B8>
>>> for i in countdown(10):
    print(i)

Starting
10
9
8
7
6
5
4
3
2
1

Ln: 156 Col: 4
```



```
import game_world
```

```
def fire_ball(self):  
    ball = Ball(self.x, self.y, self.dir*3)  
    game_world.add_object(ball, 1)
```



```
boy = None
```

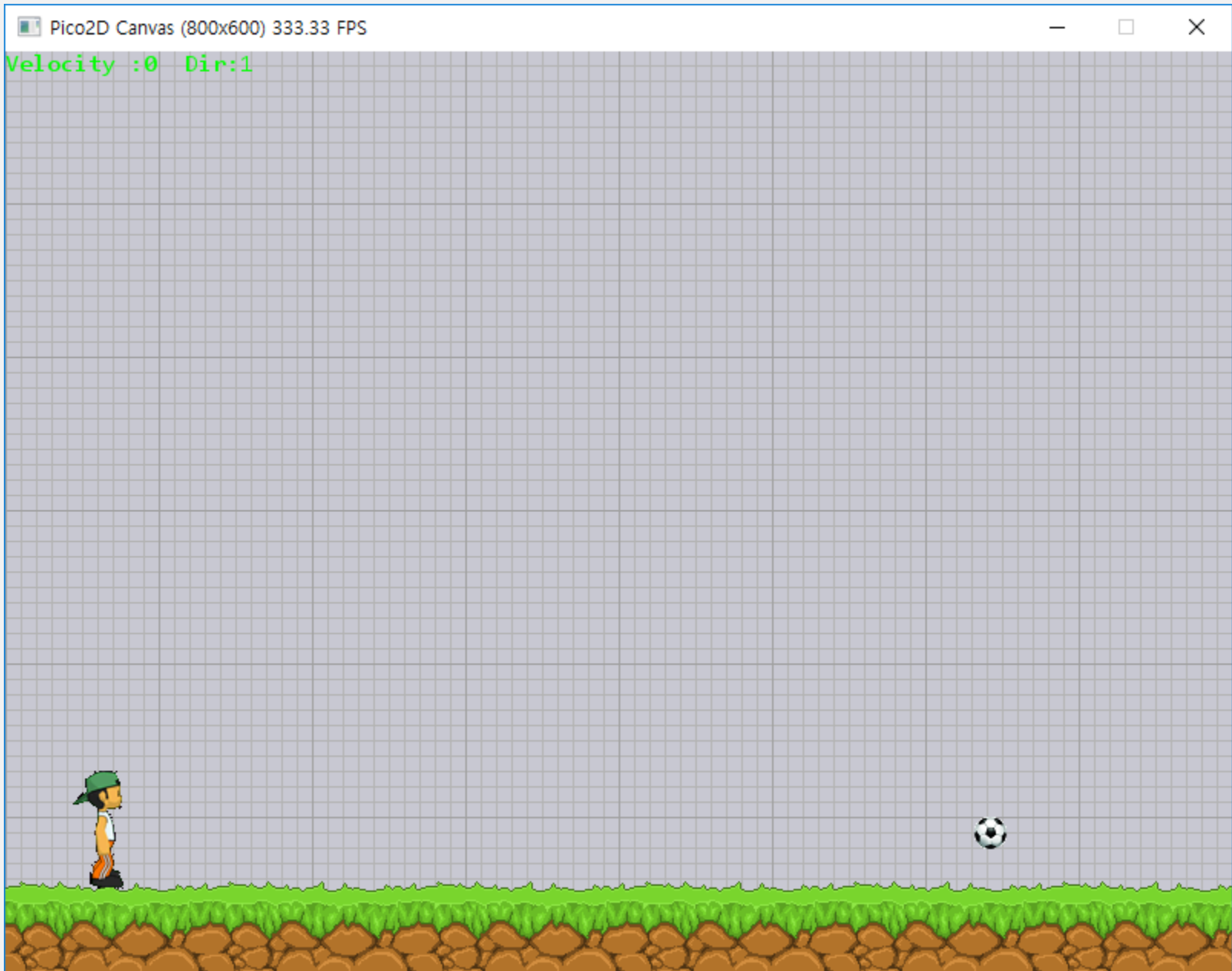
```
def enter():  
    global boy  
    boy = Boy()  
    grass = Grass()  
    game_world.add_object(grass, 0)  
    game_world.add_object(boy, 1)
```

```
def exit():  
    game_world.clear()
```



```
def update():  
    for game_object in game_world.all_objects():  
        game_object.update()  
  
def draw():  
    clear_canvas()  
    for game_object in game_world.all_objects():  
        game_object.draw()  
    update_canvas()
```





# ball.py – ball의 제거



```
from pico2d import *
import game_world

class Ball:
    image = None

    def __init__(self, x = 800, 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

        if self.x < 25 or self.x > 1600 - 25:
            game_world.remove_object(self)
```