

# Lecture #5. 애니메이션

2D 게임 프로그래밍

이대현 교수



한국공학대학교  
TECH UNIVERSITY OF KOREA

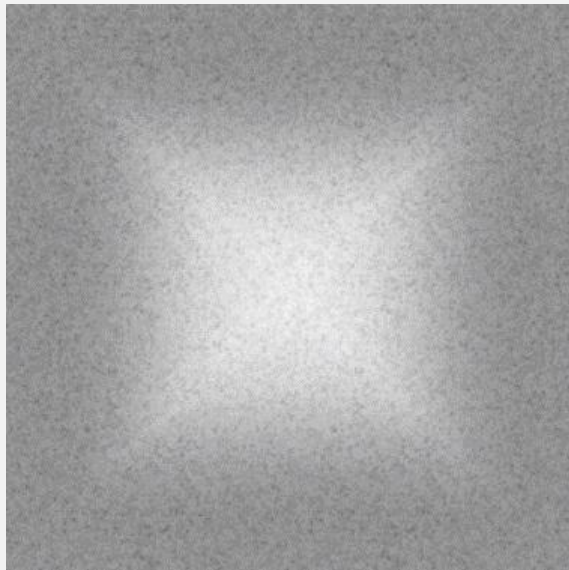
# 학습 내용

---

- 더블 버퍼링
- 폴리핑
- 스프라이트 애니메이션

# 지난 번 실습의 문제점? - 화면 플리커링

---

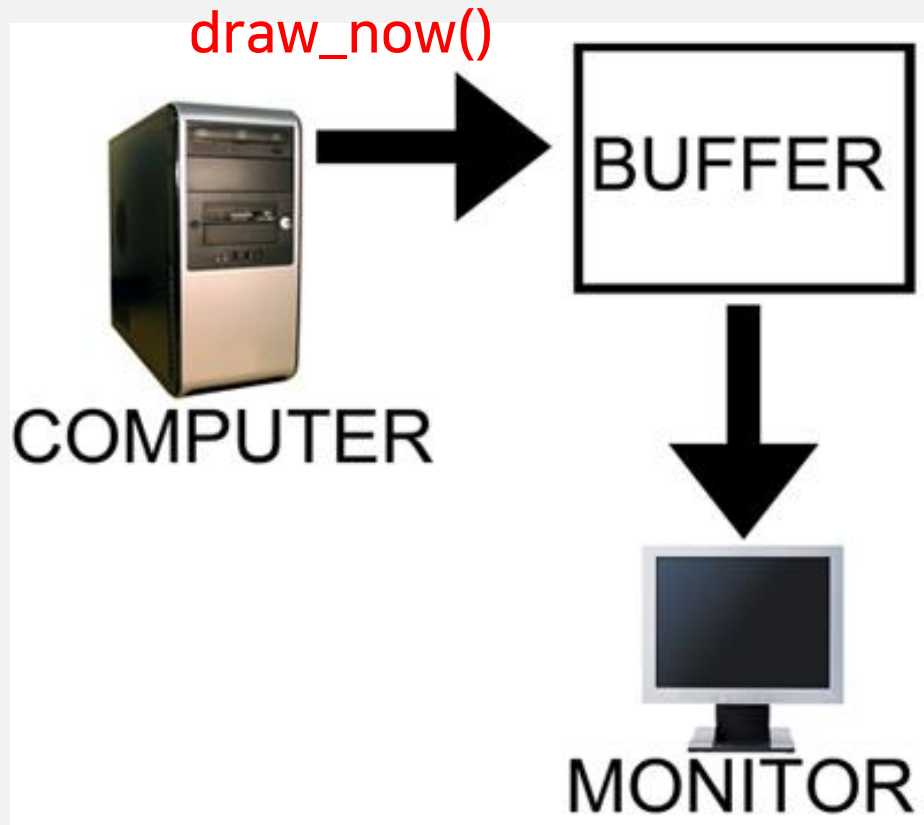


# 무대의 커튼은 왜 있을까?

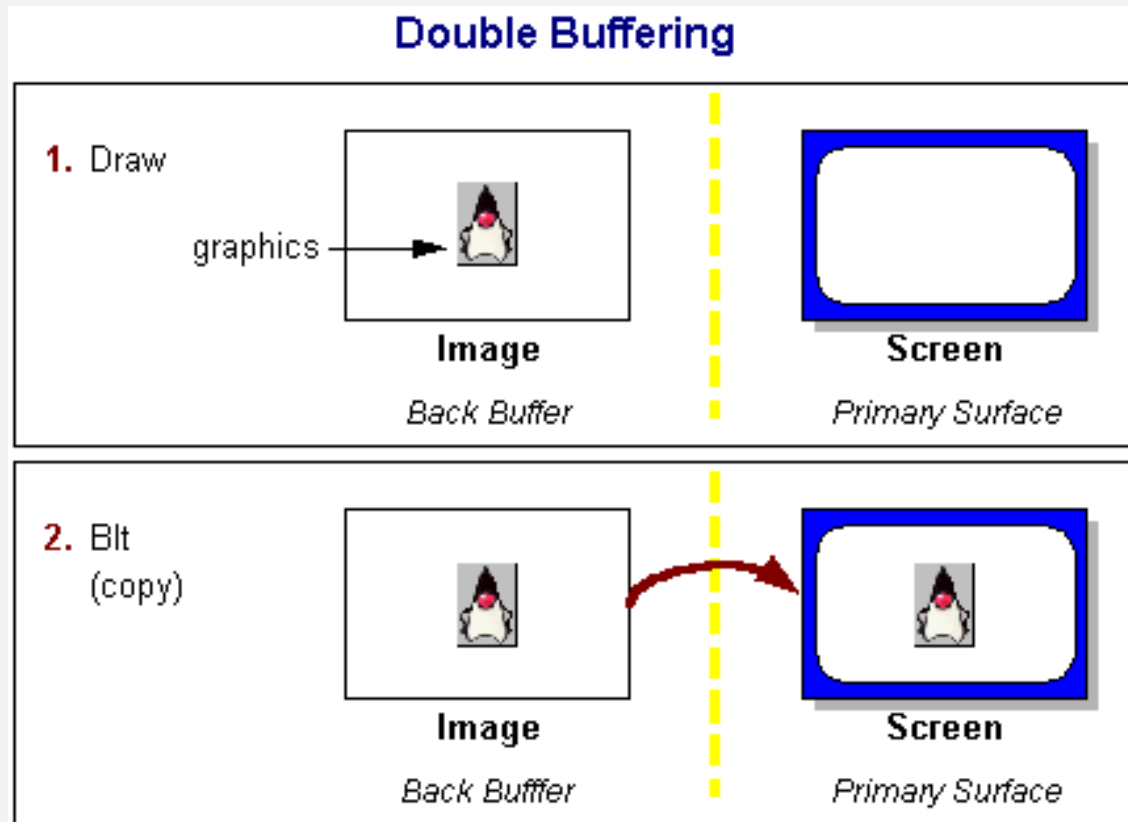
---



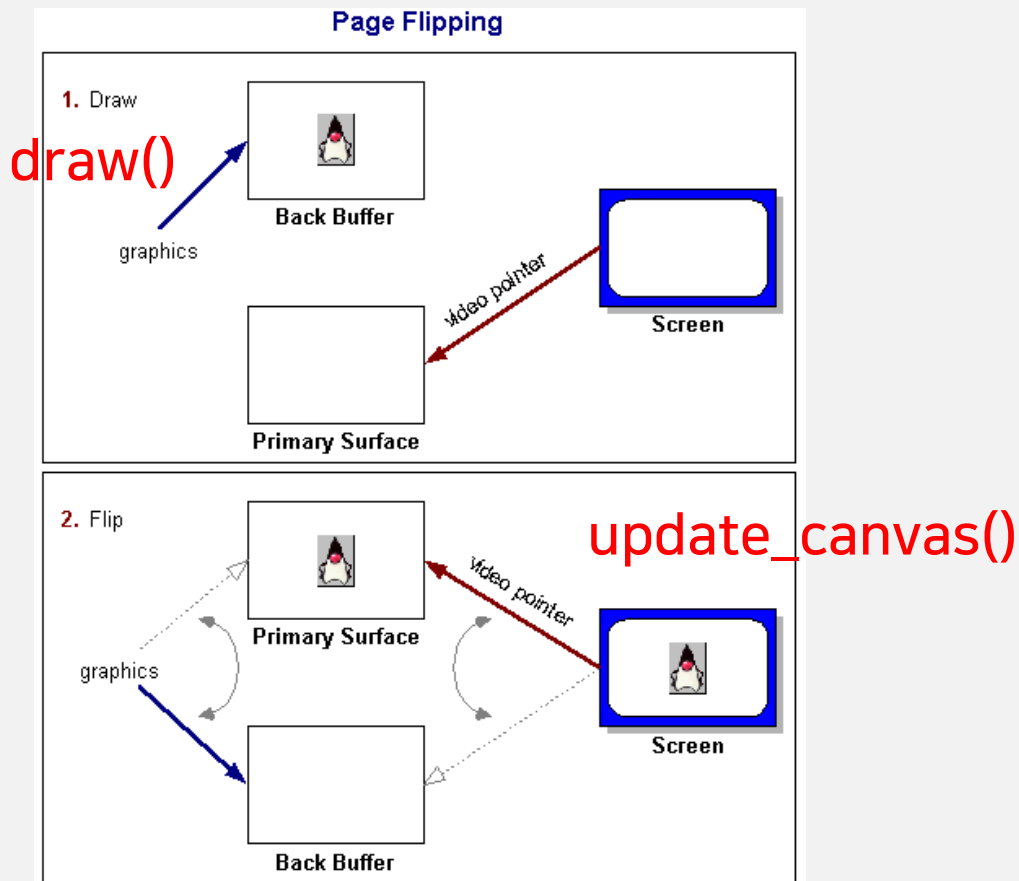
# 프레임 버퍼(Frame Buffer)



# 더블 버퍼링(Double Buffering)



# 페이지 플리핑(Page Flipping)



# 후면 버퍼(Back Buffer)에 그리기

---

```
>>> from pico2d import *  
>>> open_canvas()  
>>> character = load_image('character.png')  
>>> character.draw(100,100)  
>>> character.draw(200,200)
```



```
>>> update_canvas()
```

# 화면 지우기

---

```
>>> clear_canvas()  
>>> update_canvas()
```



## 부드러운 캐릭터 이동

# character\_moves.py



```
from pico2d import *
```

```
open_canvas()
```

```
grass = load_image('grass.png')
```

```
character = load_image('character.png')
```

```
for x in range(0, 800, 5):
```

```
    clear_canvas()
```

```
    grass.draw(400, 30)
```

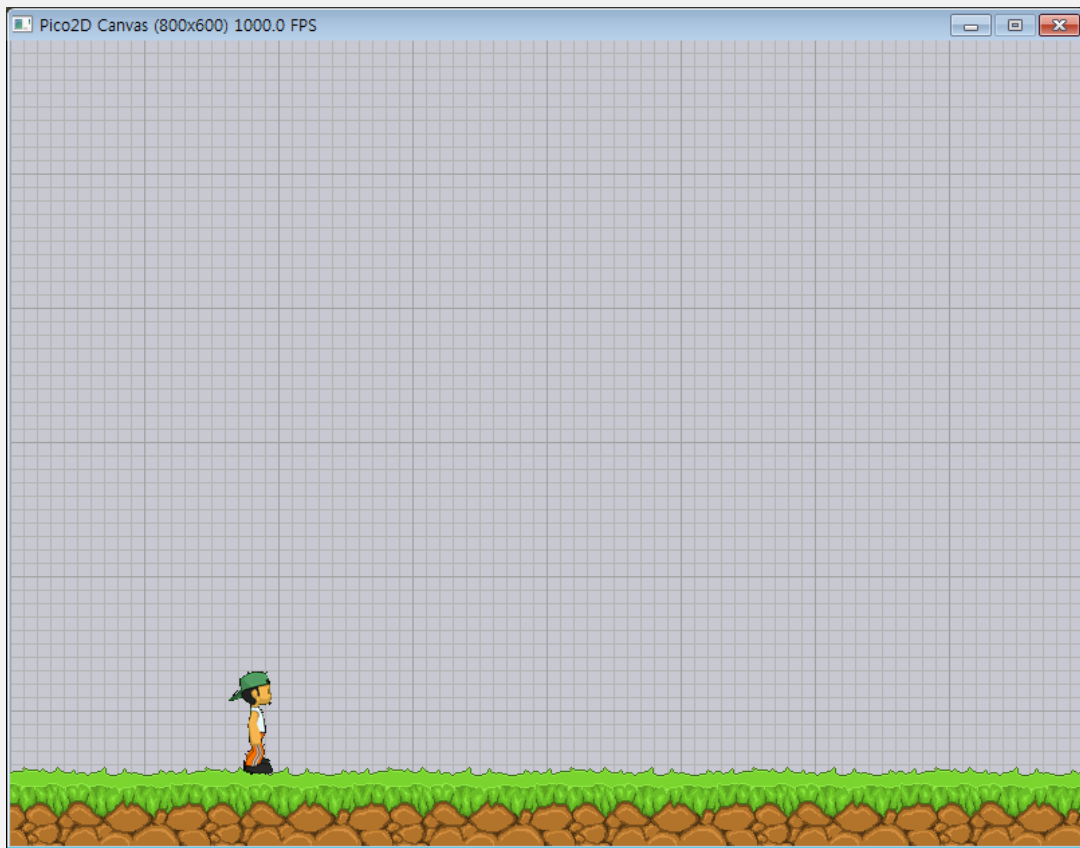
```
    character.draw(x, 90)
```

```
    update_canvas()
```

```
    delay(0.01)
```

```
close_canvas()
```

# 실행 결과



# 스프라이트(Sprite)

## ■ 스프라이트란?

- 게임 장면안에서 보여지는 이미지 또는 애니메이션되는 오브젝트
- 2D 게임에서는 게임의 모든 캐릭터들과 이동하는 물체들을 표현하는 데 사용됨.
- 3D 게임에서는 2D로 표현될 수 있는 각종 오브젝트에 사용됨.
  - 불, 연기, 작은 물체들, UI 표시 등등.

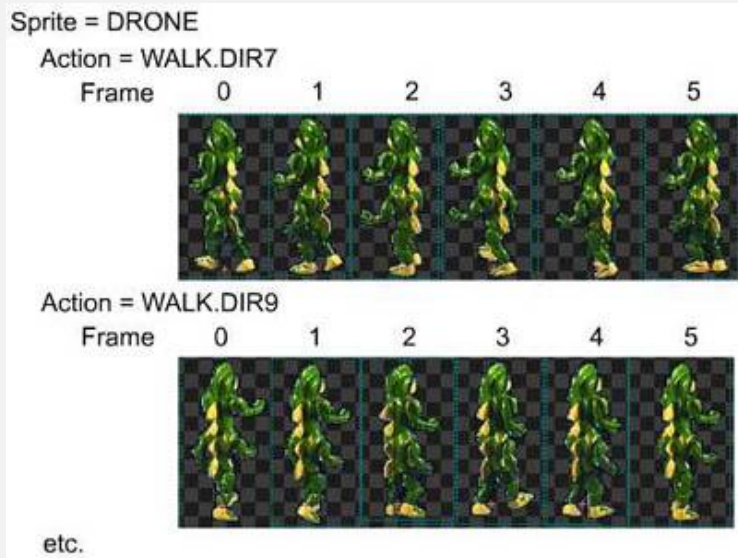


*Metal Slug 3*

# 애니메이션(Animation)

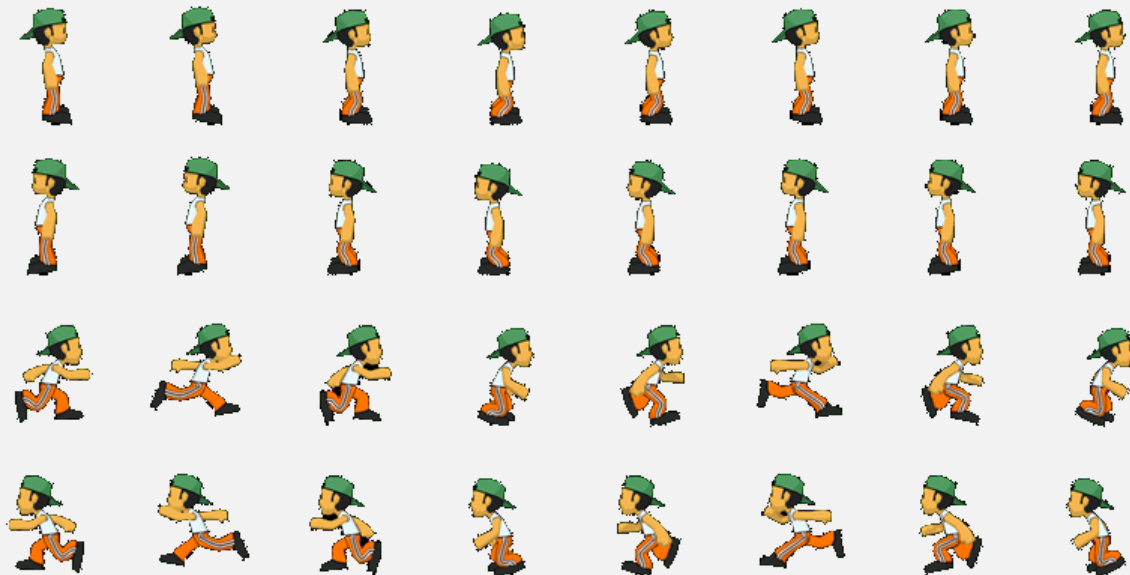
## ■ 애니메이션이란?

- 여러 개의 이미지를 일정한 시간 간격을 통해서 화면에 뿌림으로써, 물체가 움직이는 효과를 주는 것.
- 스프라이트는 여러 개의 action으로 구성됨.
  - Action: 달리기, 걷기, 제자리 동작 등과 같이 캐릭터의 움직임을 나타냄.
  - Action은 여러 개의 Frame으로 구성됨.
    - Frame은 한 개의 이미지



# 스프라이트 시트

---







## 캐릭터 애니메이션

# run\_animation.png

---



# character\_runs.py



```
from pico2d import *
```

```
open_canvas()
```

```
grass = load_image('grass.png')
```

```
character = load_image('run_animation.png')
```

```
frame = 0
```

```
for x in range(0, 800, 10):
```

```
    clear_canvas()
```

```
    grass.draw(400, 30)
```

```
    character.clip_draw(frame * 100, 0, 100, 100, x, 90)
```

```
    update_canvas()
```

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

```
    delay(0.05)
```

```
close_canvas()
```

# 현재 Edit 중인 파일의 실행(Ctrl+Shift+F10)

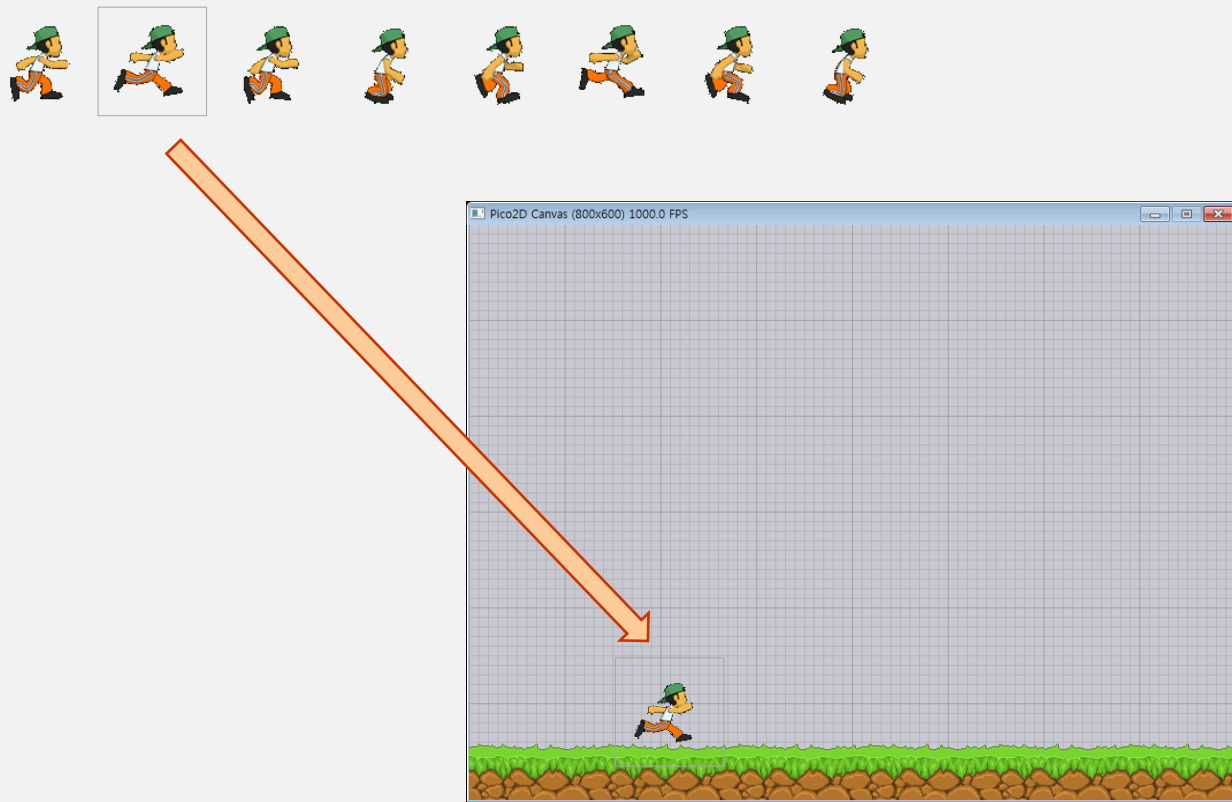
The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named `character_runs.py` with the following code:

```
1 from pico2d import *
2 open_canvas()
3 grass = load_image('grass.png')
4 character = load_image('run_animation.png')
5
6 def draw(frame * 100, 0, 100, 100, x, 90):
7     p_draw(frame * 100, 0, 100, 100, x, 130, 200, 200)
8     composite_draw(frame * 100, 0, 100, 100, 0, 'h', x, 130, 200, 200)
9
10 def main():
11     +
12
13 if __name__ == '__main__':
14     main()
```

A context menu is open over the code, showing various actions. The **Run** option, which includes the keyboard shortcut **Ctrl+Shift+F10**, is highlighted with a red box. Below the menu, a run configuration window is visible, showing the command `C:\Python\python.exe W:\WorkCodingLive\2023-2DGP\2023-2DGP\character_runs.py` and the status `Pico2d is prepared.` and `Process finished with exit code 0`.

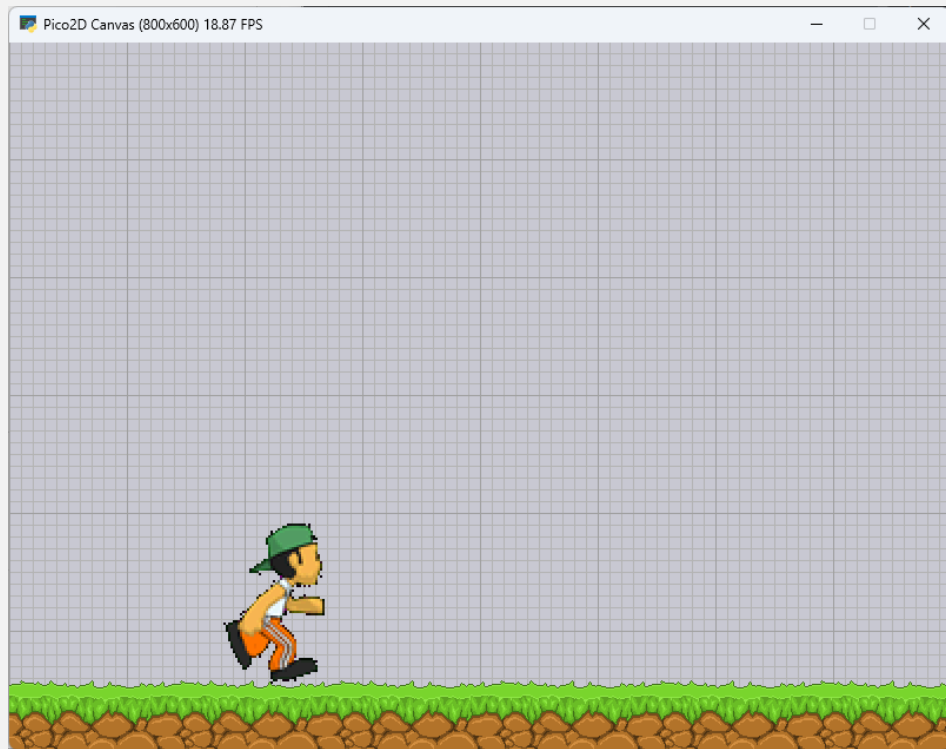
**Ctrl + Shift + F10 으로 실행.**  
이후에는, Shift + F10 을 하면, 지속적으로 앞서 실행한 코드를 실행할 수 있음.

# clip\_draw(left, bottom, width, height, x, y)



# 확대

```
character.clip_draw(frame * 100, 0, 100, 100, x, 130, 200, 200)
```



# 좌우반전

```
# clip_composite_draw(self, left, bottom, width, height, rad, flip, x, y, w=None, h=None)
```

```
character.clip_composite_draw(frame * 100, 0, 100, 100, 0, 'h', x, 130, 200, 200)
```

