

2D 게임 프로그래밍

제1강 2D 렌더링 기초

이대현
한국산업기술대학교



학습 내용

- 2D 게임의 기본 요소
- 필요 Tool 들의 설치
- 캐릭터 이미지의 렌더링과 이동

2D 게임?

■ 게임이란?

- “가상 월드에 존재하는 여러 객체들의 상호작용”

■ 게임의 기본 구성 요소

- 배경
- 캐릭터, 오브젝트
- UI - GUI, 입력(키,마우스,터치, ...)
- AI
- 사운드

■ 2D 게임?

- 현재 진행 중인 게임 가상 월드의 내용을 화면에 2D 그림으로 보여주는 것
- 배경,캐릭터(오브젝트)의 표현(렌더링)을 2D 이미지들의 조합으로 구성함!

GUI

배경(World)

캐릭터

오브젝트

2D 게임 개발 접근법

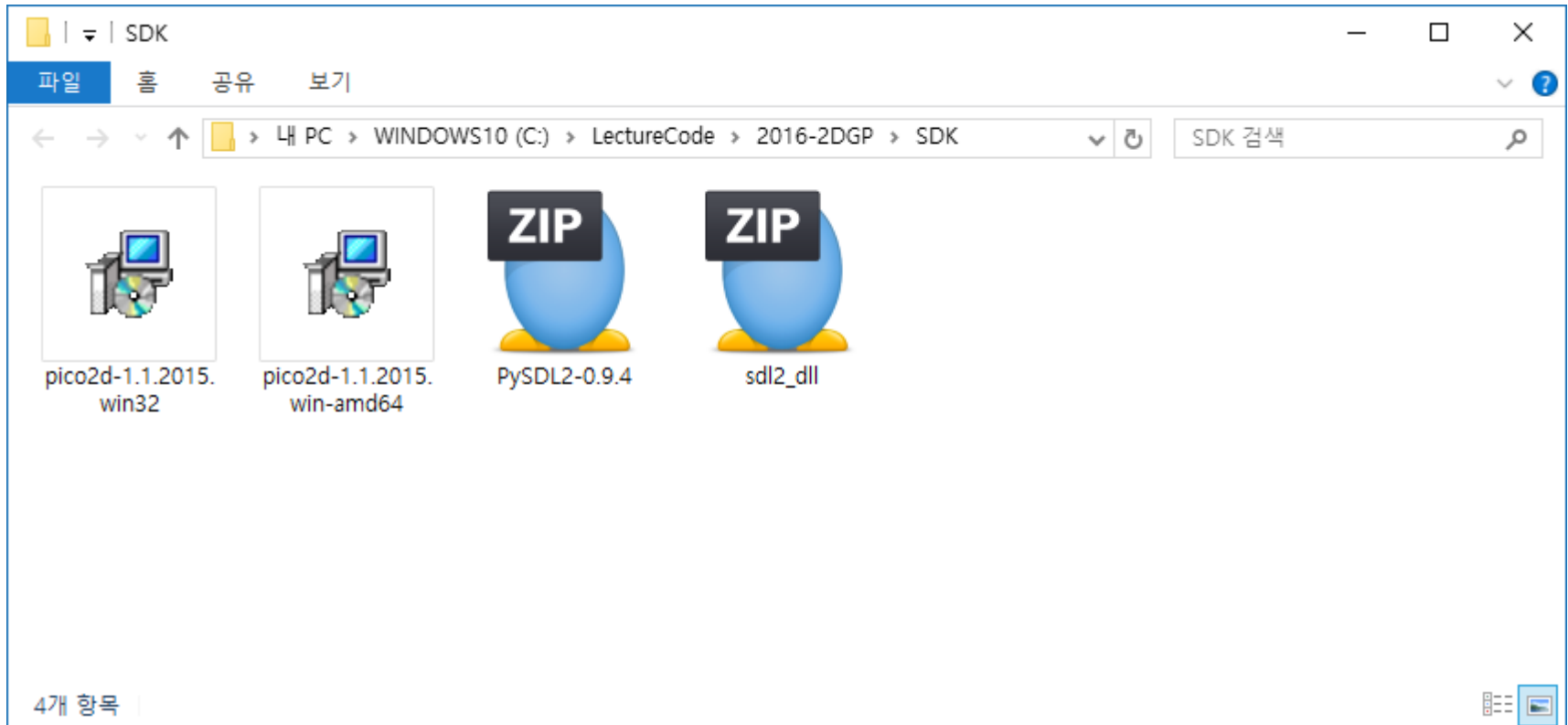
- 플랫폼 종속적 방법
 - Direct X
 - OpenGL
 - Simple Frame Buffer
- 플랫폼 독립적 방법, Cross Platform
 - Unity3D
 - COCOS2D
 - SDL
 - 그 외의 범용 2D 렌더링 라이브러리

SDL(Simple DirectMedia Layer)

- SDL이란?
 - 크로스 플랫폼 멀티미디어 라이브러리.
 - 비디오, 오디오 및 사용자 입력을 처리하는 API로 구성.
 - 기본적으로 2D 그래픽 라이브러리. 3D는 OpenGL을 통해서 지원.
- SDL이 지원하는 플랫폼
 - PC: Windows, Linux, Mac OS
 - Phone: Android, iOS,
- 라이선싱(SDL 2.0)
 - zlib license
 - 자유롭게 상용 게임을 개발할 수 있음.
 - SDL1.2 → GNU LGPL 라이선싱
- 홈페이지
 - www.libsdl.org

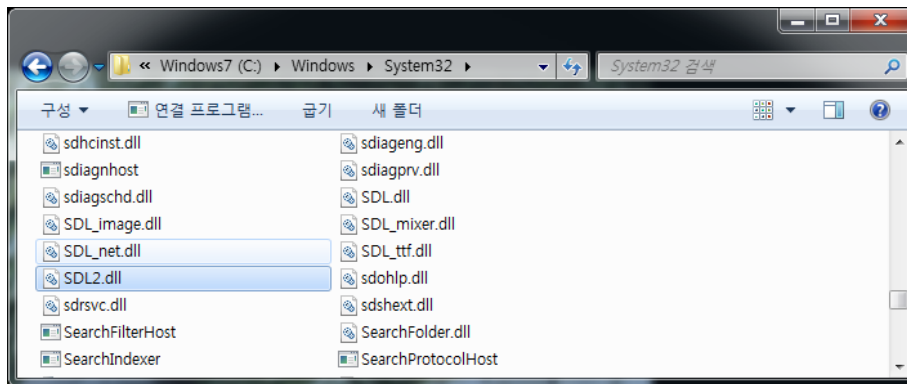


git pull



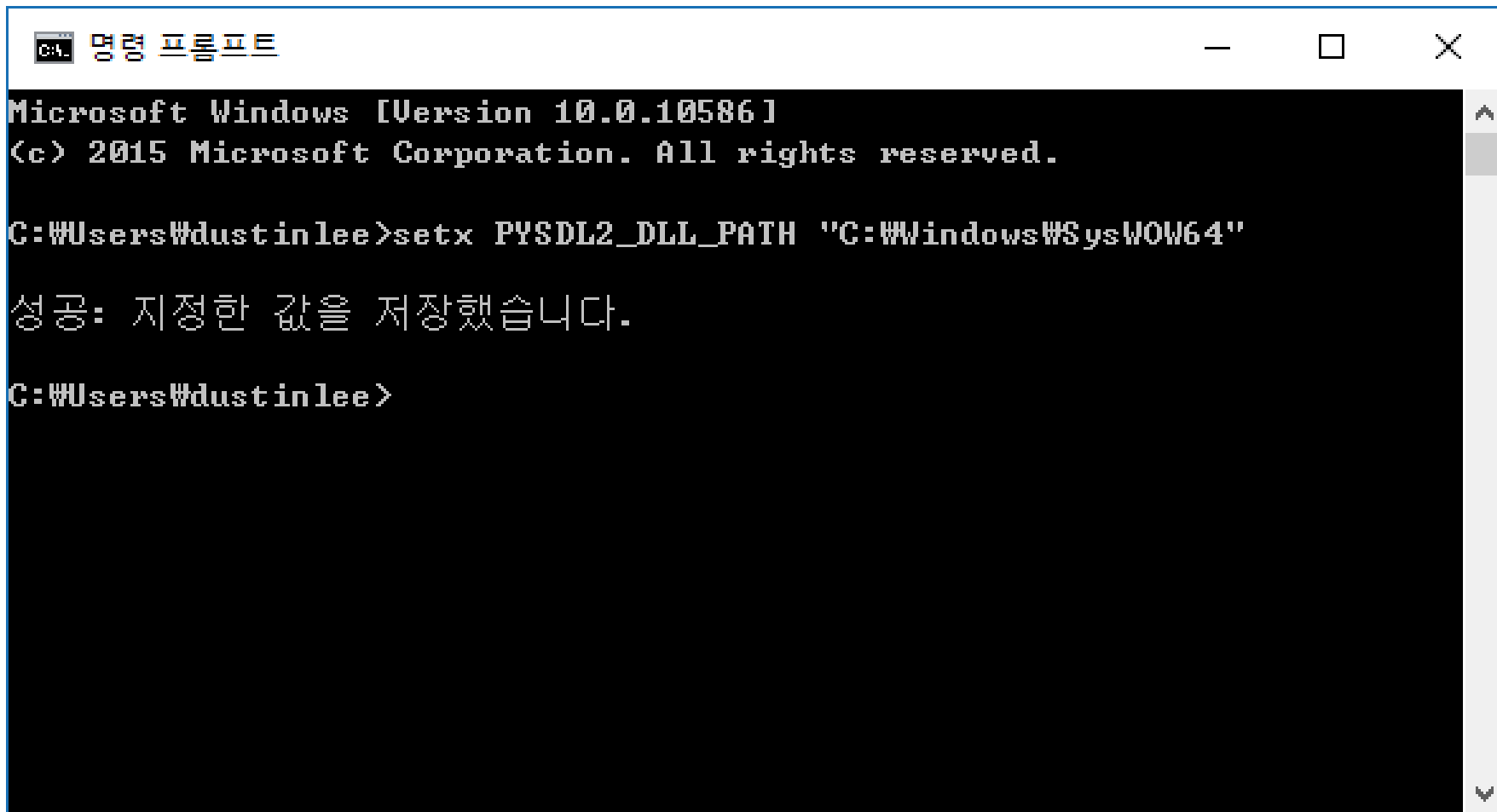
SDL 라이브러리 DLL 등록/복사

- SDL 의 DLL 파일들을 윈도우에 등록
 - 32bit로 개발하고자 하는 경우, sdl2_dll/x86/ 폴더 안의 모든.dll 파일들을 C:\Windows\System32 폴더에 복사.
 - 64bit로 개발하고자 하는 경우, sdl2_dll/x64/ 폴더안의 모든.dll 파일들을 C:\Windows\SysWOW64 폴더에 복사
 - 64bit PC에서, 32bit python으로 32bit app을 개발하려면, SysWOW64 폴더안에, 32bit SDL DLL을 복사해야 함...



64비트의 경우: 환경 변수 PYSDL2_DLL_PATH 의 설정

cmd 실행



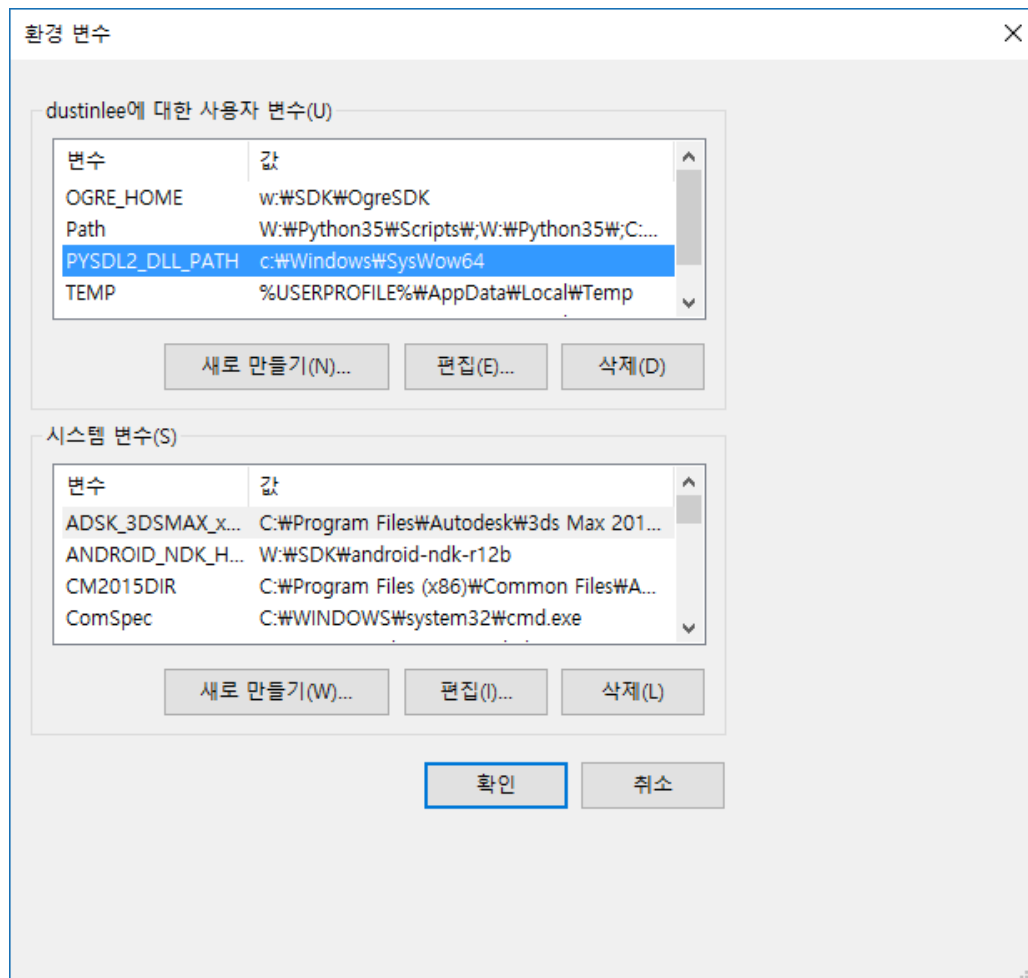
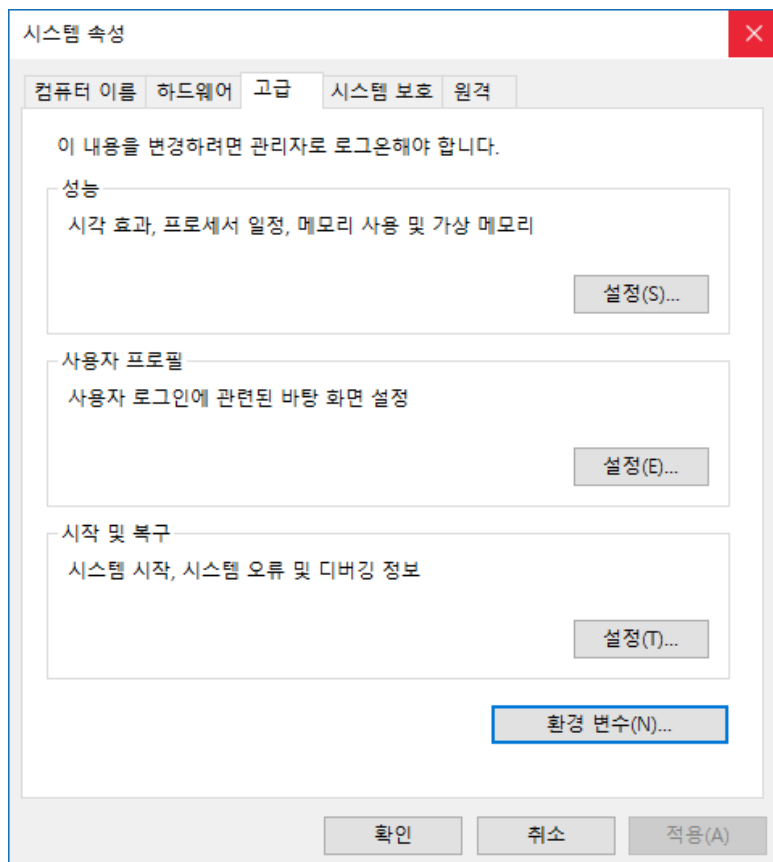
```
명령 프롬프트
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\dustinlee>setx PYSDL2_DLL_PATH "C:\Windows\SysWOW64"

성공: 지정한 값을 저장했습니다.

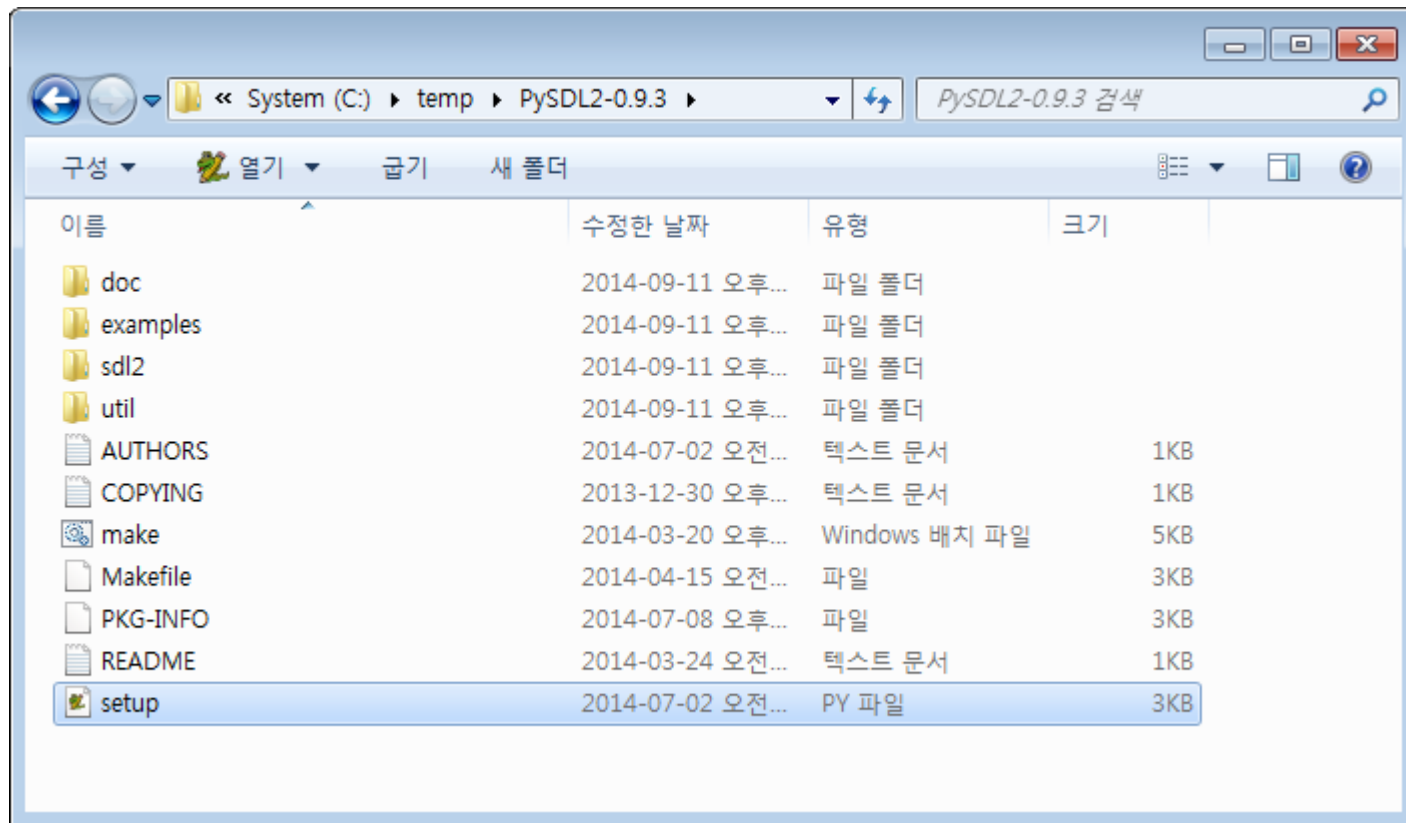
C:\Users\dustinlee>
```

시스템 환경 변수 확인

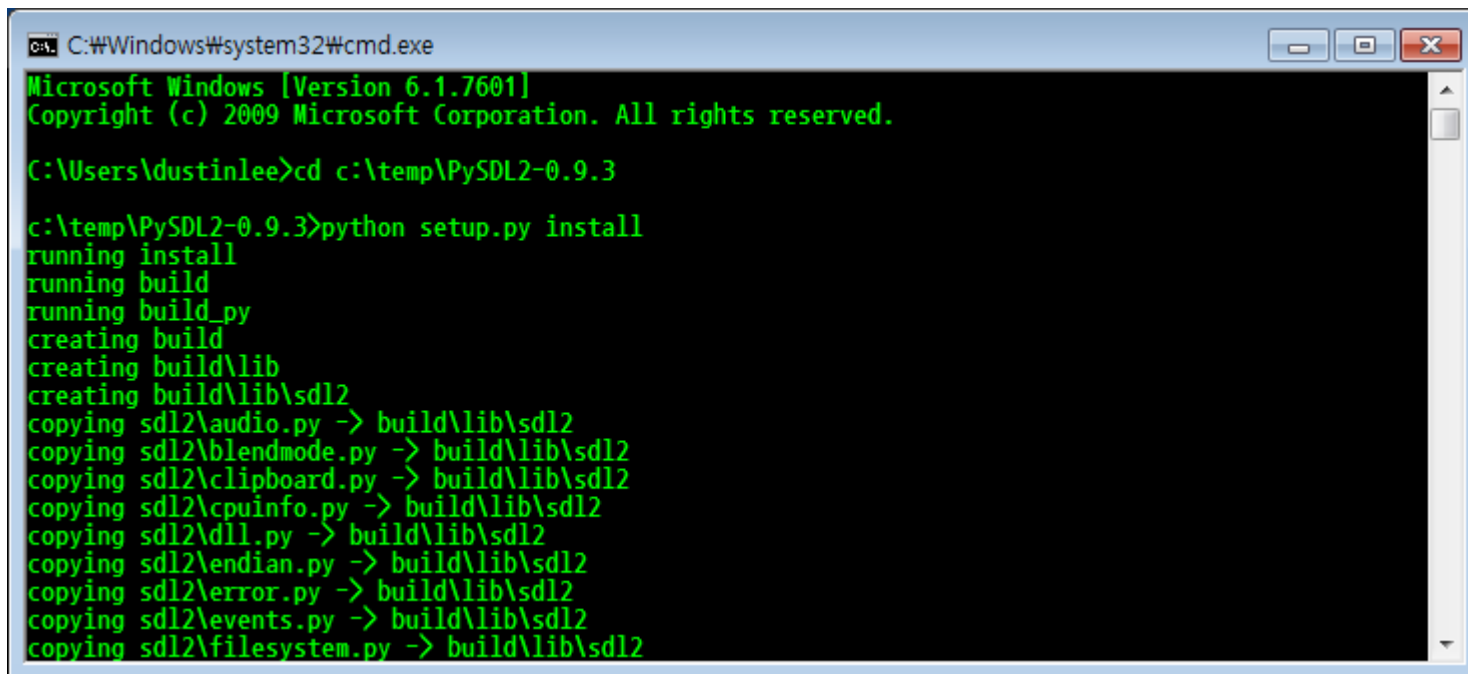


PySDL 설치

- 적당한 곳에 압축 해제
 - C:\temp



- cmd (코맨드) 창을 열고
- 폴더를 c:\temp\PySDL2-0.9.3 으로 변경한 후,
- python setup.py install



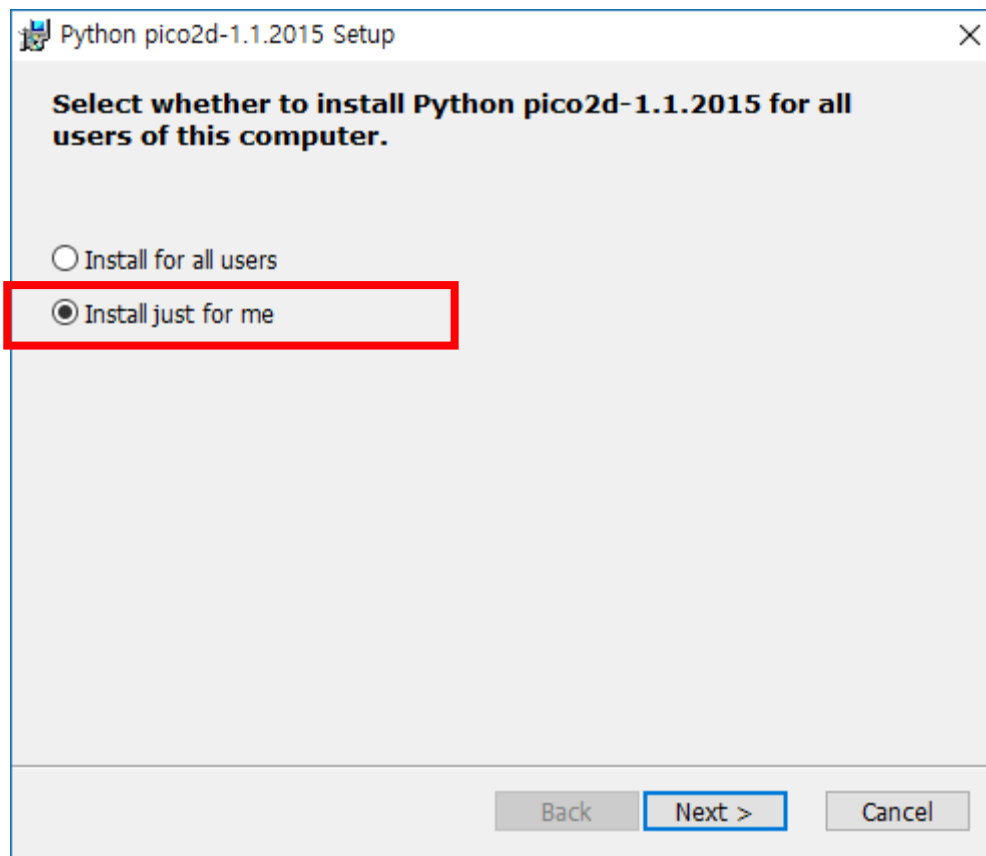
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

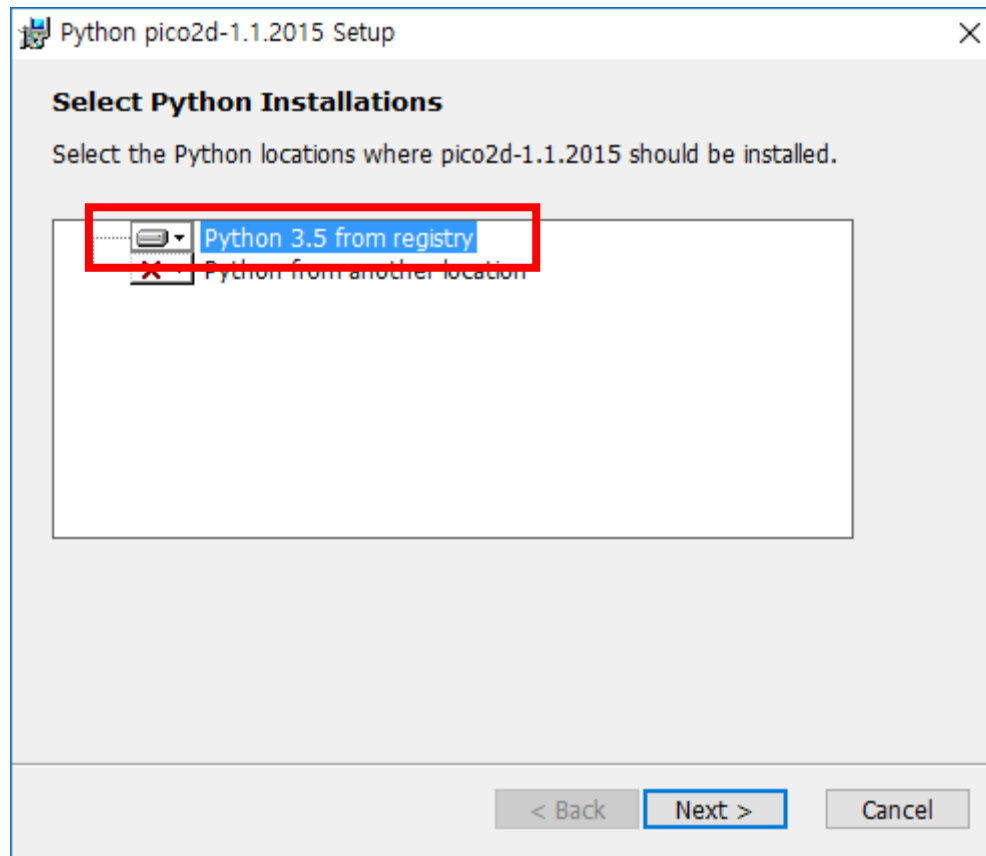
C:\Users\dustinlee>cd c:\temp\PySDL2-0.9.3

c:\temp\PySDL2-0.9.3>python setup.py install
running install
running build
running build_py
creating build
creating build\lib
creating build\lib\sdl2
copying sdl2\audio.py -> build\lib\sdl2
copying sdl2\blendmode.py -> build\lib\sdl2
copying sdl2\clipboard.py -> build\lib\sdl2
copying sdl2\cpuinfo.py -> build\lib\sdl2
copying sdl2\dll.py -> build\lib\sdl2
copying sdl2\endian.py -> build\lib\sdl2
copying sdl2\error.py -> build\lib\sdl2
copying sdl2\events.py -> build\lib\sdl2
copying sdl2\filesystem.py -> build\lib\sdl2
```

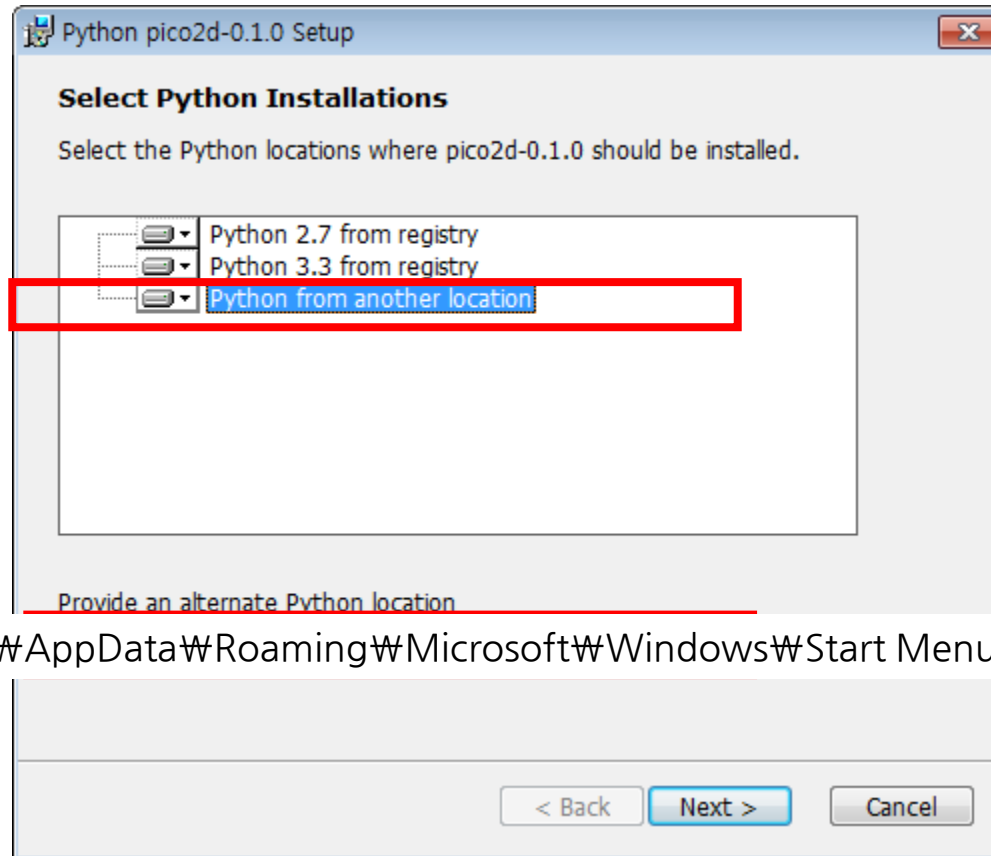
Pico2d 설치

- 32 bit
 - pico2d 32bit 설치
- 64 bit
 - pico2d 64bit 설치



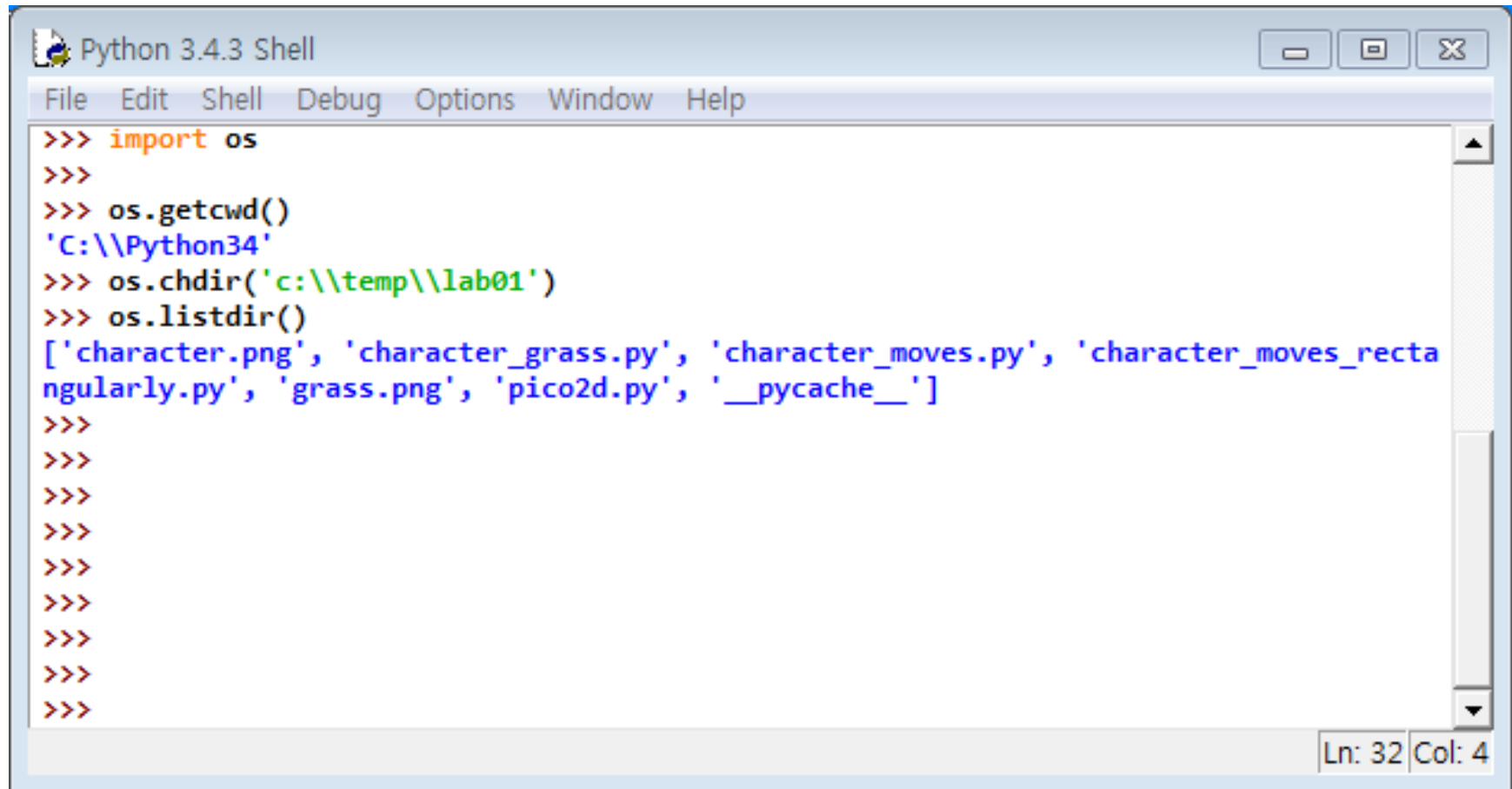


만약 python 3.5 from registry 가 나오지 않으면...



C:\Users\dustinlee\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Python 3.5

OS 모듈을 이용한 Working Directory 설정

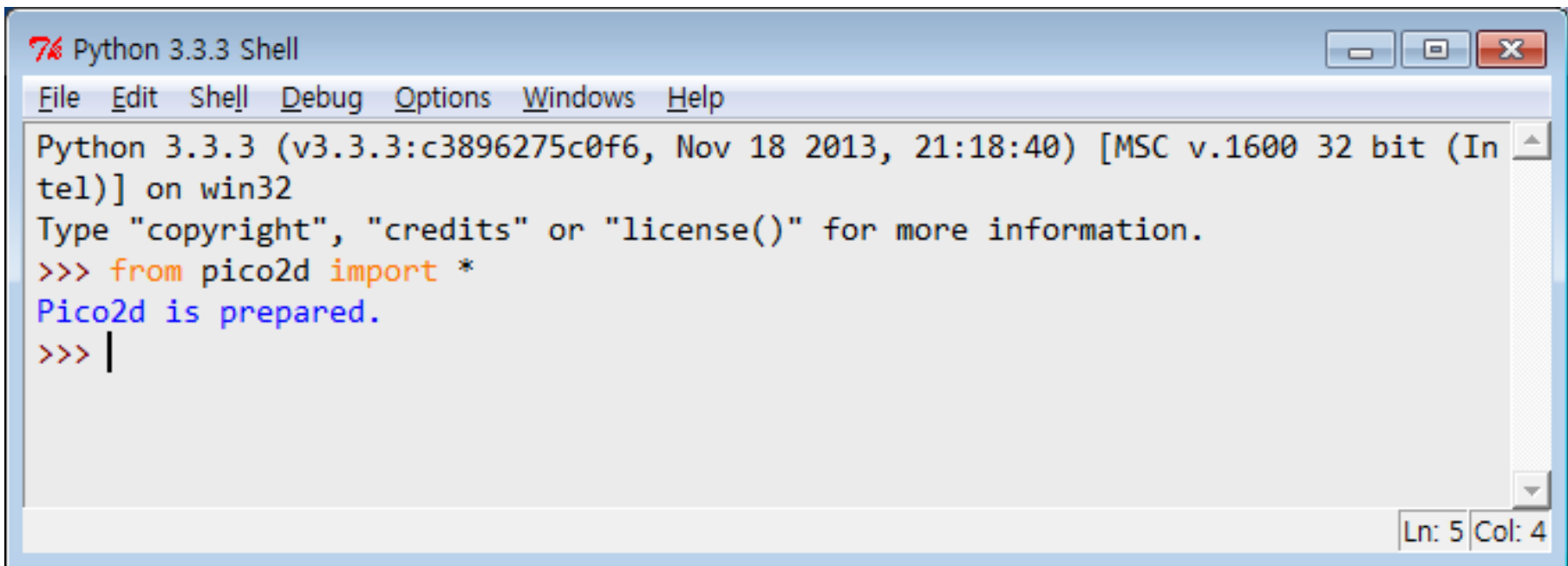


The screenshot shows a Python 3.4.3 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and a command prompt interface. The user has entered several commands to check and change the current working directory. The output shows the current directory is 'C:\\Python34', and after changing to 'c:\\temp\\lab01', the directory listing includes various files like 'character.png', 'character_grass.py', etc.

```
>>> import os
>>>
>>> os.getcwd()
'C:\\Python34'
>>> os.chdir('c:\\temp\\lab01')
>>> os.listdir()
['character.png', 'character_grass.py', 'character_moves.py', 'character_moves_recta
ngularly.py', 'grass.png', 'pico2d.py', '__pycache__']
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```

Ln: 32 Col: 4

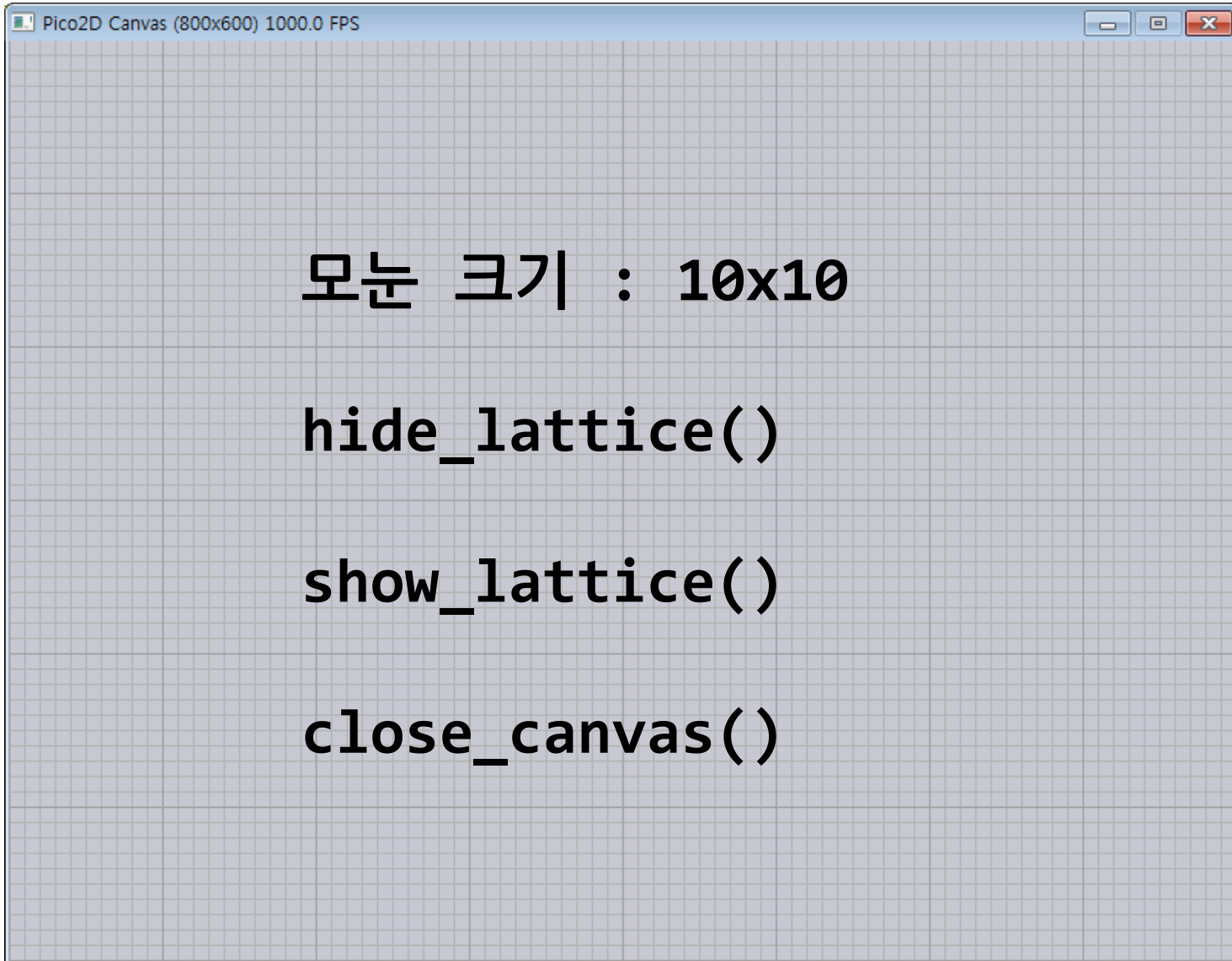
pico2d 라이브러리 준비



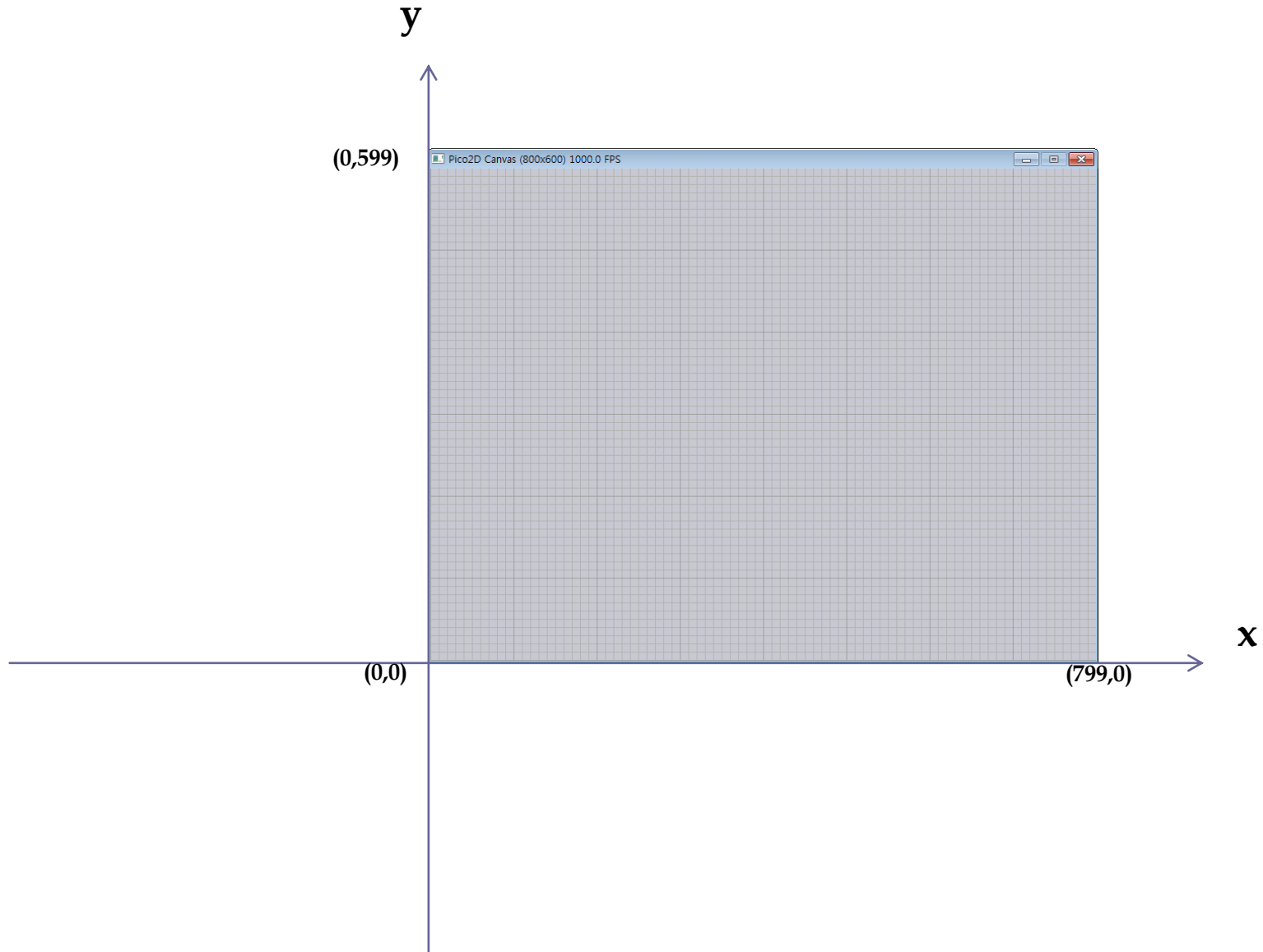
```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> from pico2d import *
Pico2d is prepared.
>>> |
```

Ln: 5 Col: 4

open_canvas(800, 600)



캔버스의 좌표계

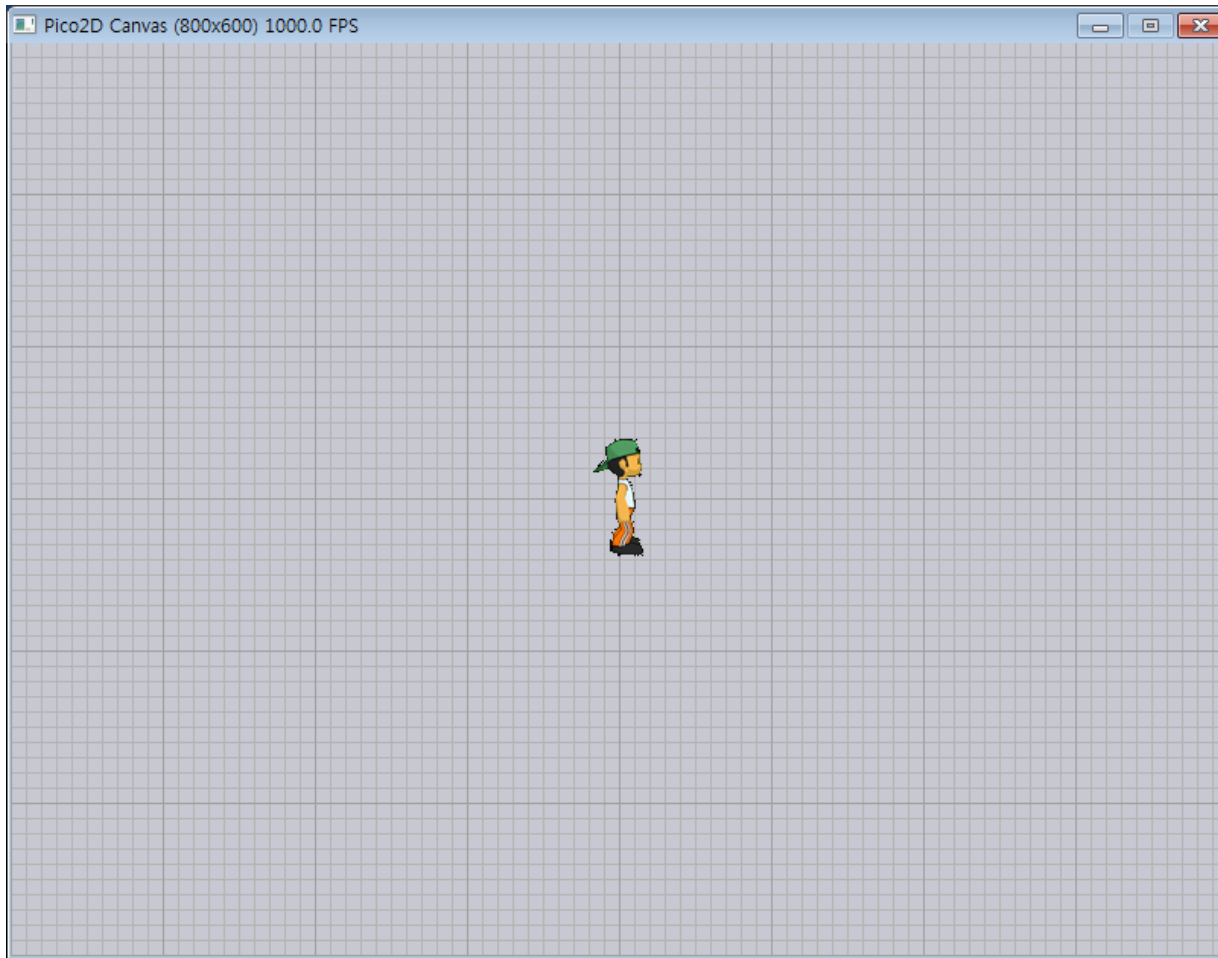


JPG vs PNG

우리의 주인공

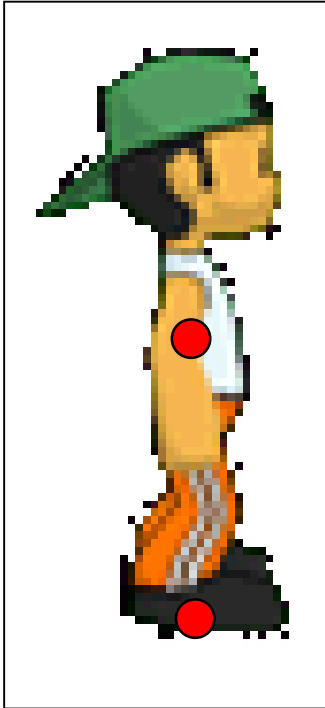


```
>>> image = load_image('character.png')
```



```
>>> image.draw_now(400, 300)
```

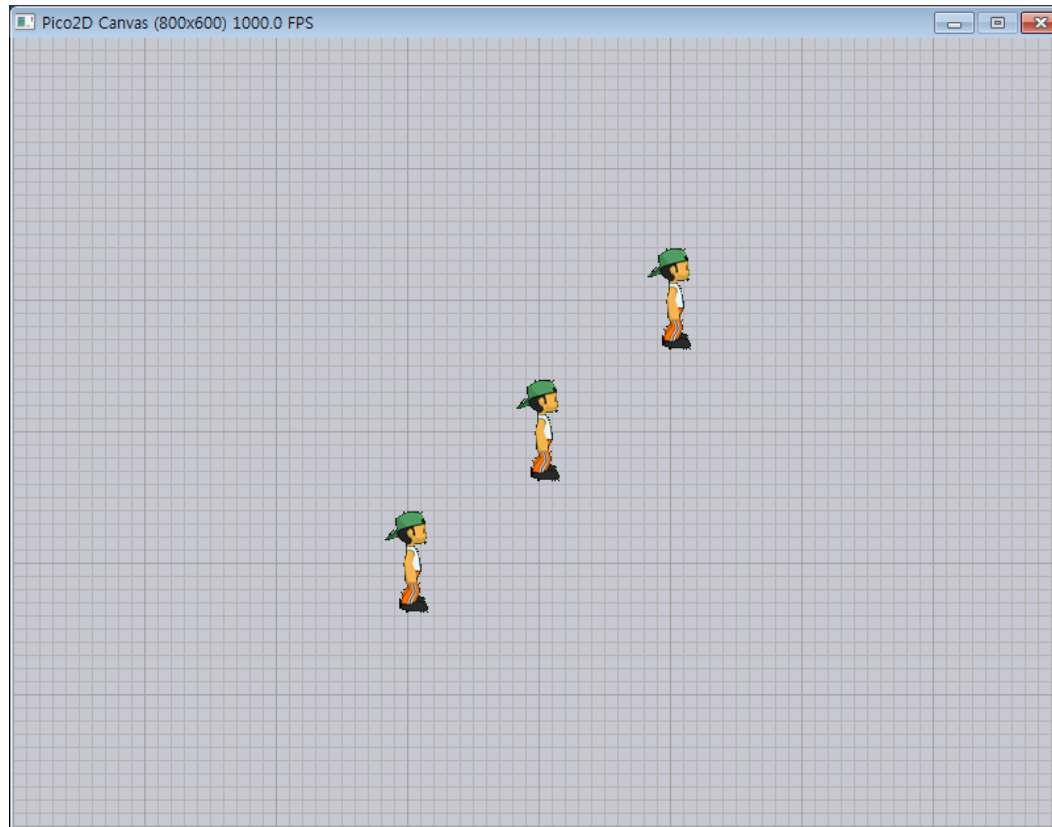
피봇(Pivot)



여기가 피봇입니다.

이 점을 피봇으로 삼기도 합니다

몇 명 더 그려 봅시다~

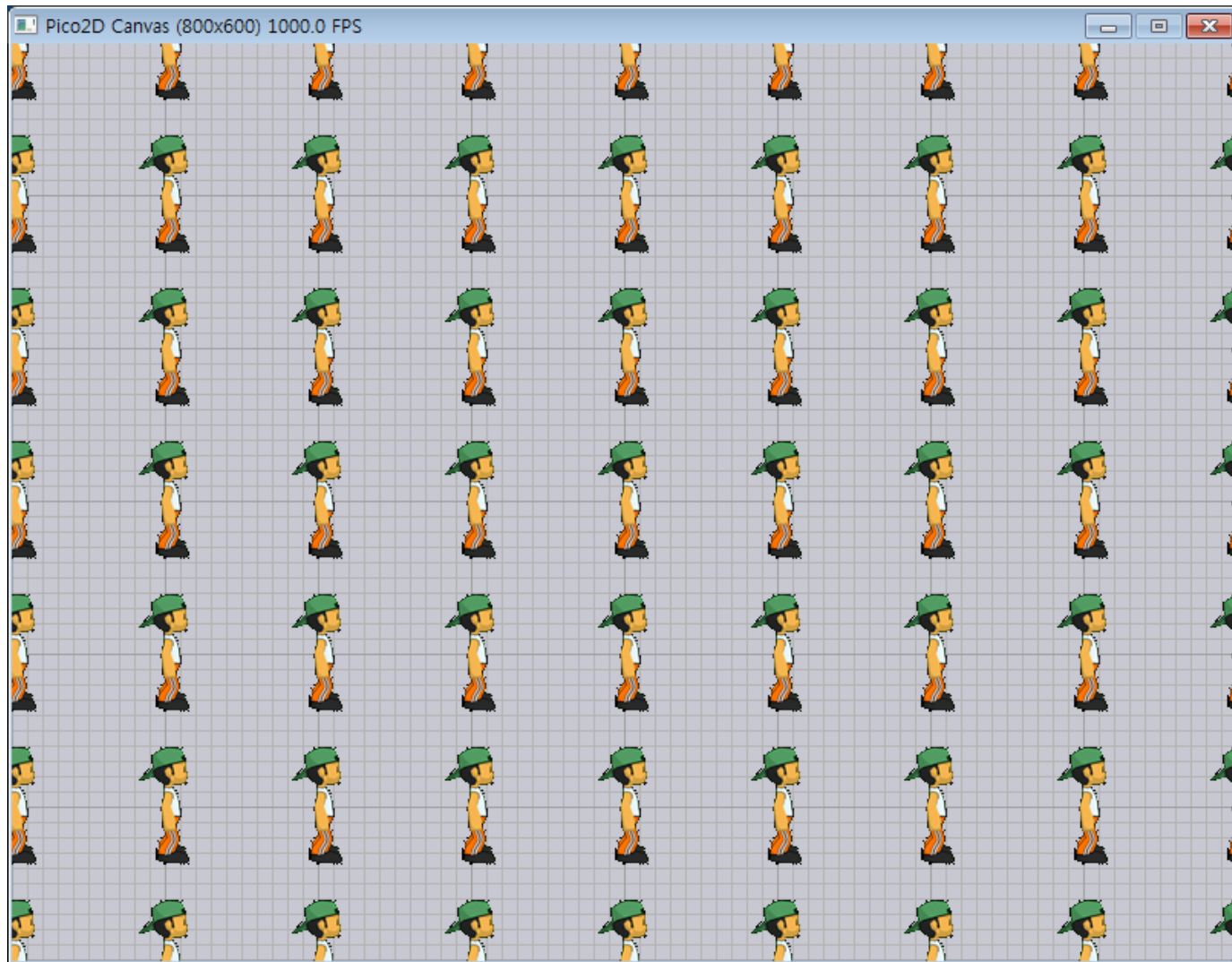


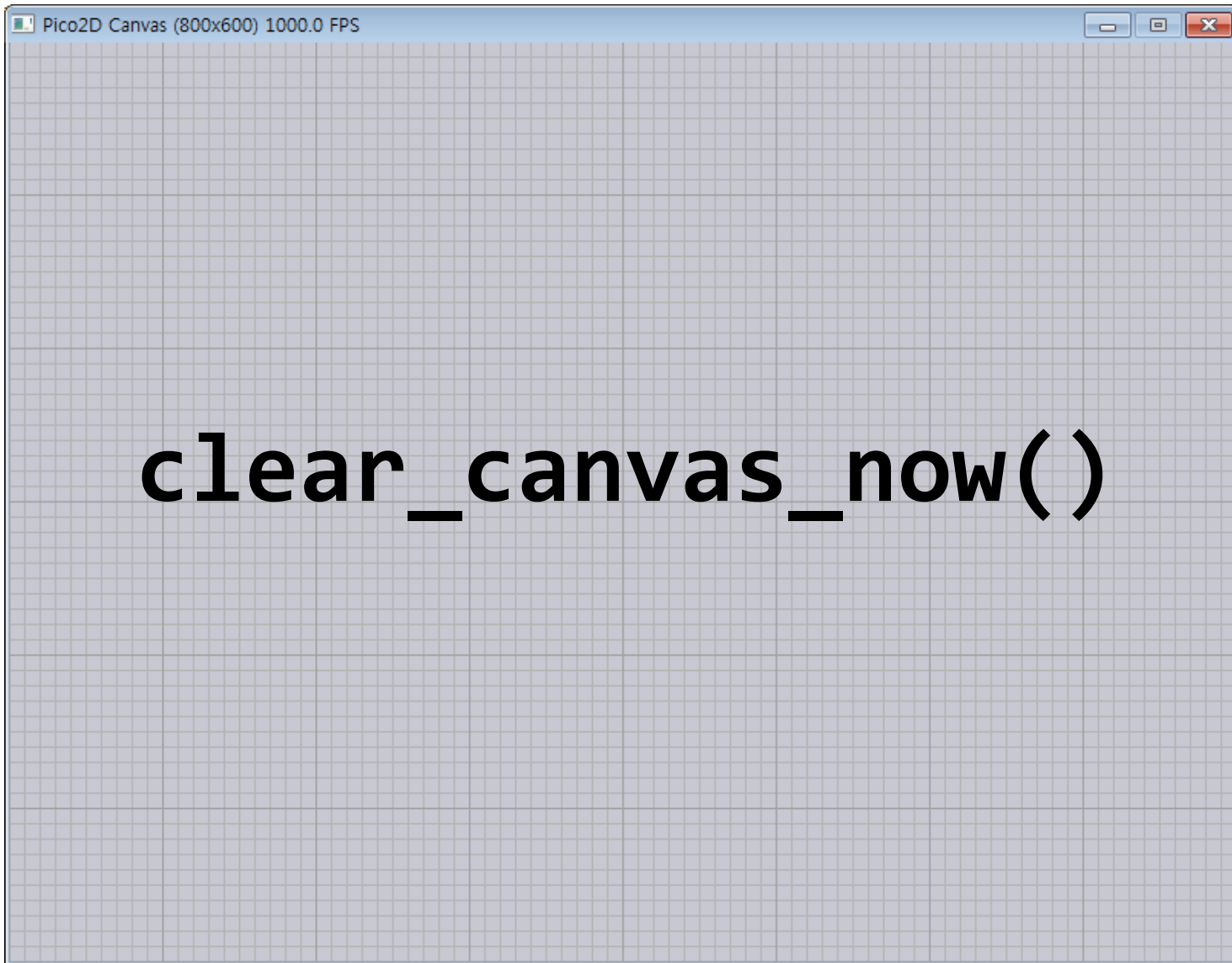
```
>>> image.draw_now(300,200)  
>>> image.draw_now(500,400)
```

떼로 그리기

```
>>> for x in range(0,9):  
    for y in range (0, 7):  
        image.draw_now(x * 100, y * 100)
```

캐릭터 떼!







Character_moves.py

캐릭터 이동

character_grass.py

```
from pico2d import *
```

```
open_canvas()
```

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

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

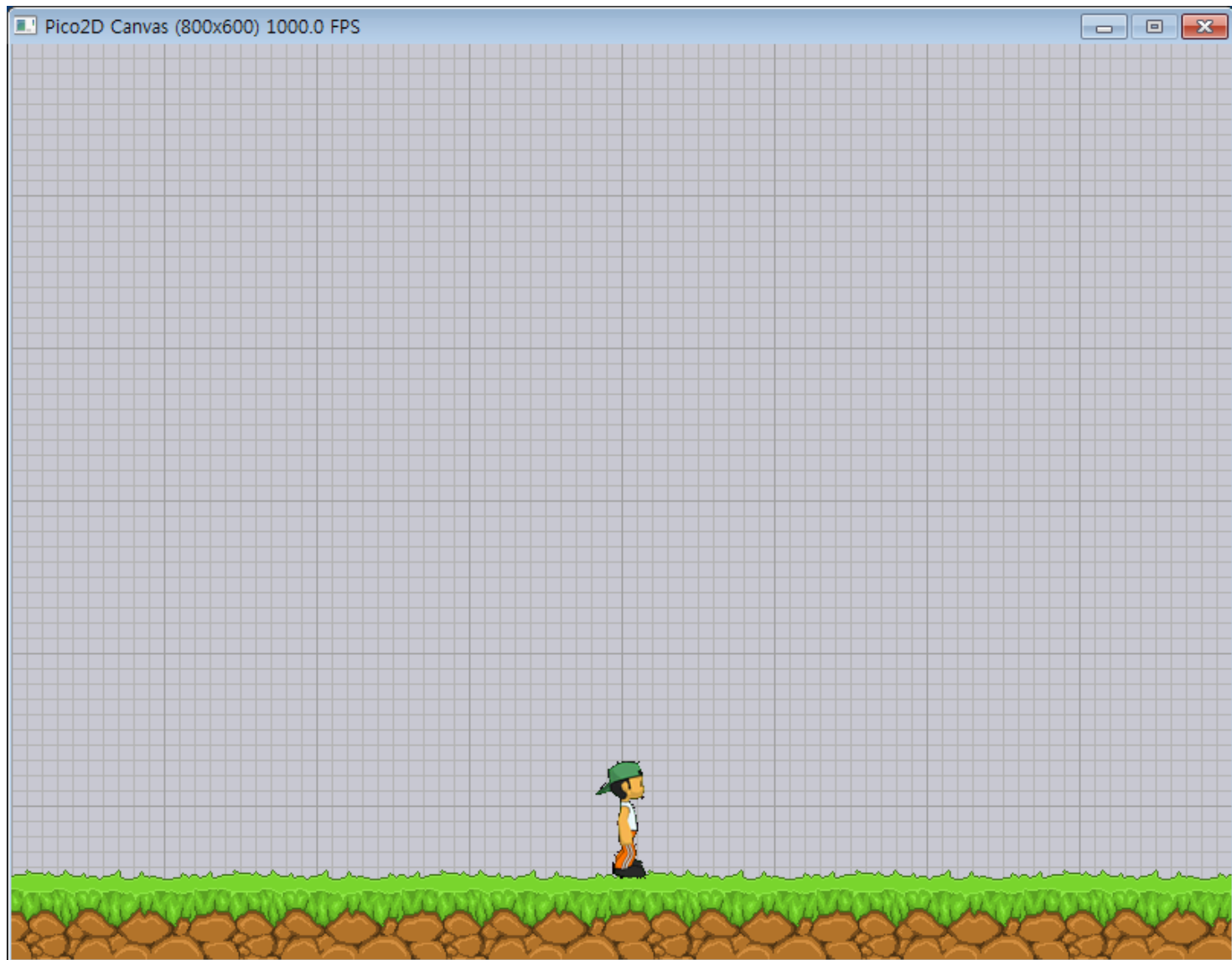
```
grass.draw_now(400, 30)
```

```
character.draw_now(400, 90)
```

```
delay(5)
```

```
close_canvas()
```





character_moves.py



```
from pico2d import *

open_canvas()

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

x = 0
while (x < 800):
    clear_canvas_now()
    grass.draw_now(400, 30)
    character.draw_now(x, 90)
    x = x + 2
    delay(0.01)

close_canvas()
```



```
x = 0
while (x < 800):
    clear_canvas_now()
    grass.draw_now(400, 30)
    character.draw_now(x, 90)
    x = x + 2
    delay(0.01)
```