

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



2D 게임 개발 환경 구성

■ 필수 환경

- Windows 10 64 bit
- Python 3.6.+
- Git / TortoiseGit

■ 설치할 것들

- SDL 라이브러리 - 2D 이미지 처리 라이브러리
- PySDL2 - SDL의 파이썬 버전
- Pico2d - PySDL2를 이용한 간단한 2D 라이브러리

Git pull 로 SDK 가져오기

The screenshot shows the TortoiseGit GUI with a file explorer view of the SDK directory. The directory path is 2017-2DGP > SDK. The SDK directory contains three files: pico2d-1.1.2015.win-amd64, PySDL2-0.9.5, and sdl2_dll, all marked as ZIP files. The TortoiseGit menu is open, showing options like Pull, Fetch, Push, and others. The SDK directory path is 2017-2DGP > SDK.

SDK Directory Contents:

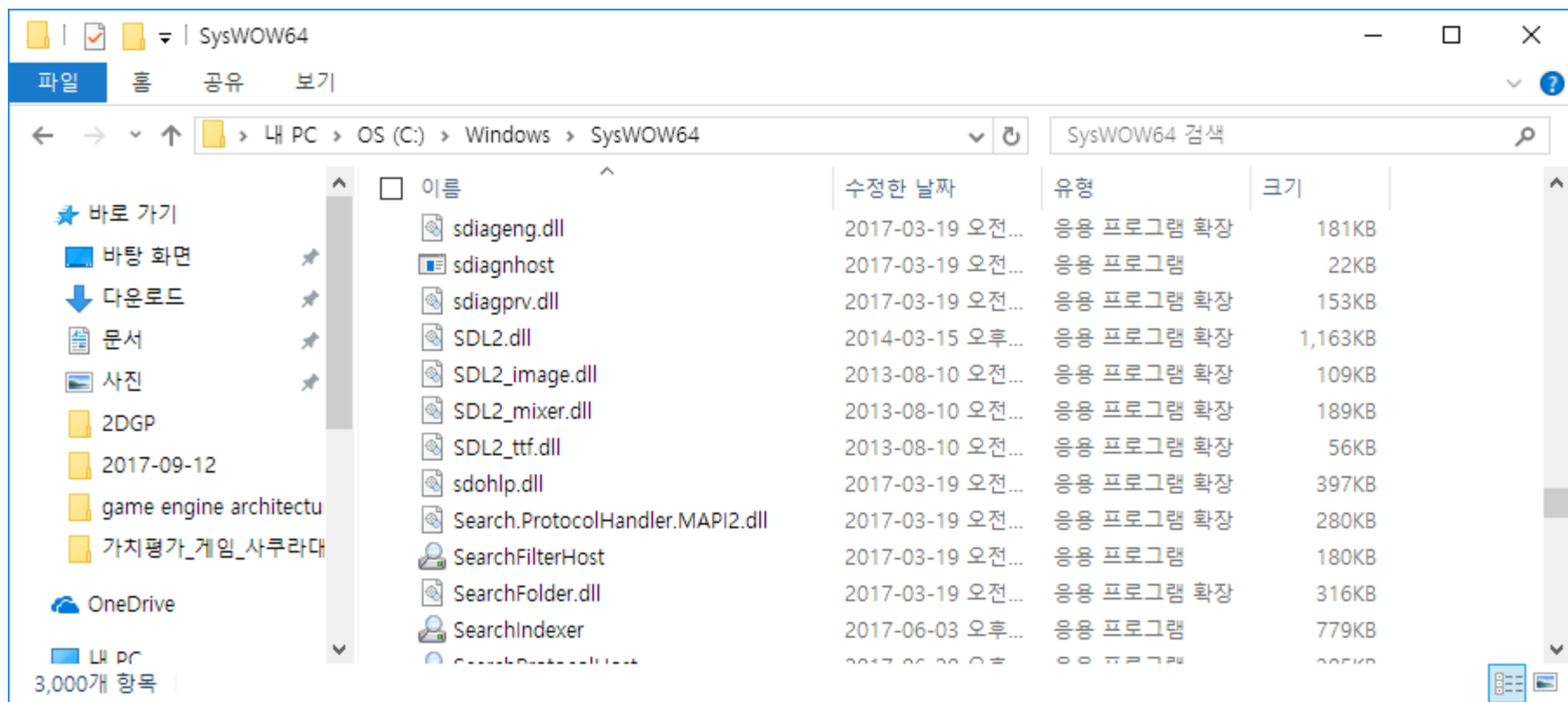
- pico2d-1.1.2015.win-amd64 (ZIP)
- PySDL2-0.9.5 (ZIP)
- sdl2_dll (ZIP)

TortoiseGit Menu Options:

- Pull...
- Fetch...
- Push...
- Diff
- Diff with previous version
- Show log
- Show Reflog
- Browse References
- Daemon
- Revision graph
- Repo-browser
- Check for modifications
- Rebase...
- Stash Save
- Bisect start
- Resolve...
- Revert...
- Clean up...
- Switch/Checkout...
- Merge...
- Create Branch...
- Create Tag...
- Export...
- Add...
- Submodule Add...
- Create Patch Serial...
- Apply Patch Serial...
- Settings
- Help
- About

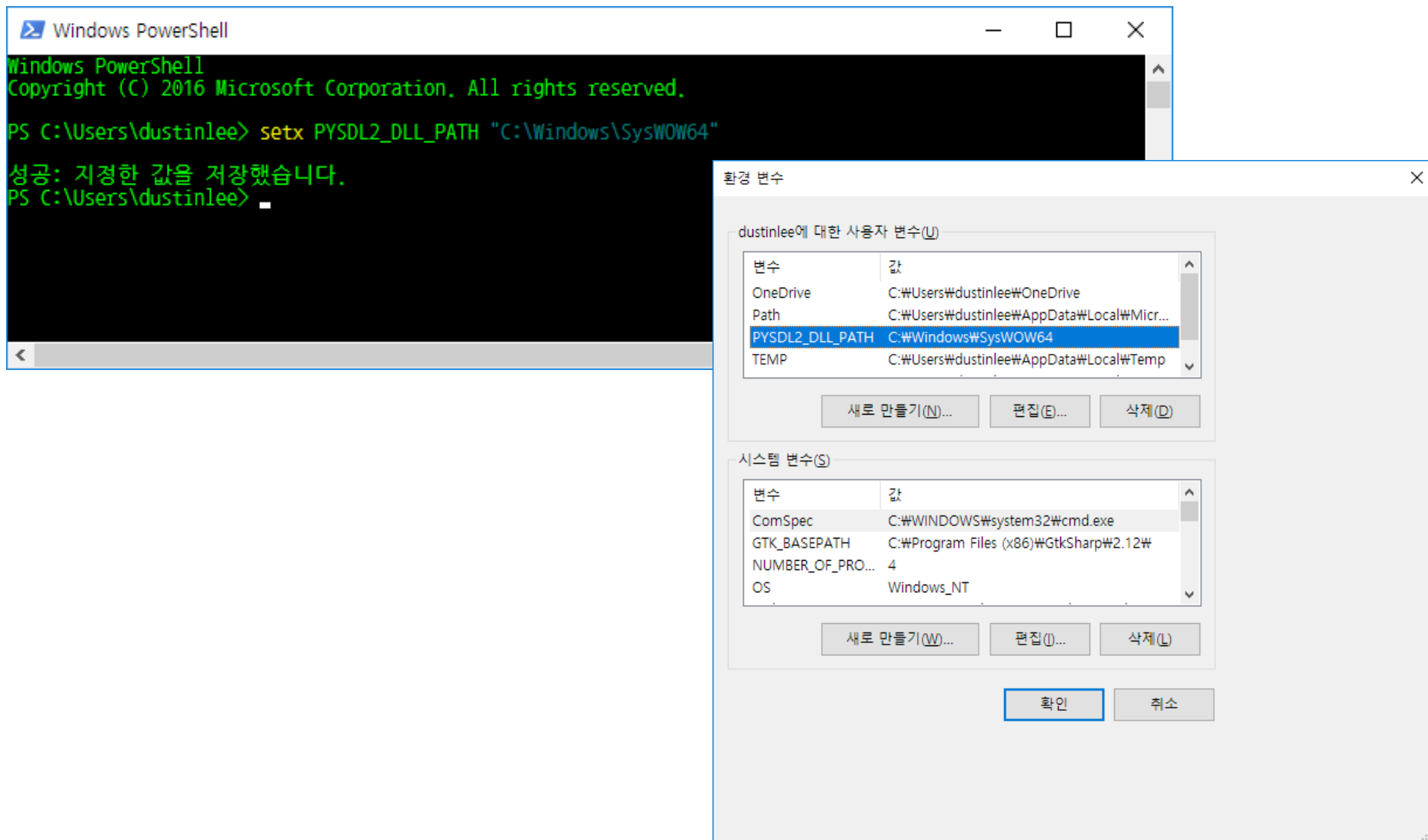
SDL 라이브러리 DLL 등록/복사

- SDL 의 DLL 파일들을 윈도우 시스템 폴더에 복사
 - 64bit로 개발하고자 하는 경우, sdl2_dll/x64/ 폴더안의 모든.dll 파일들을 C:\Windows\SysWOW64 폴더에 복사

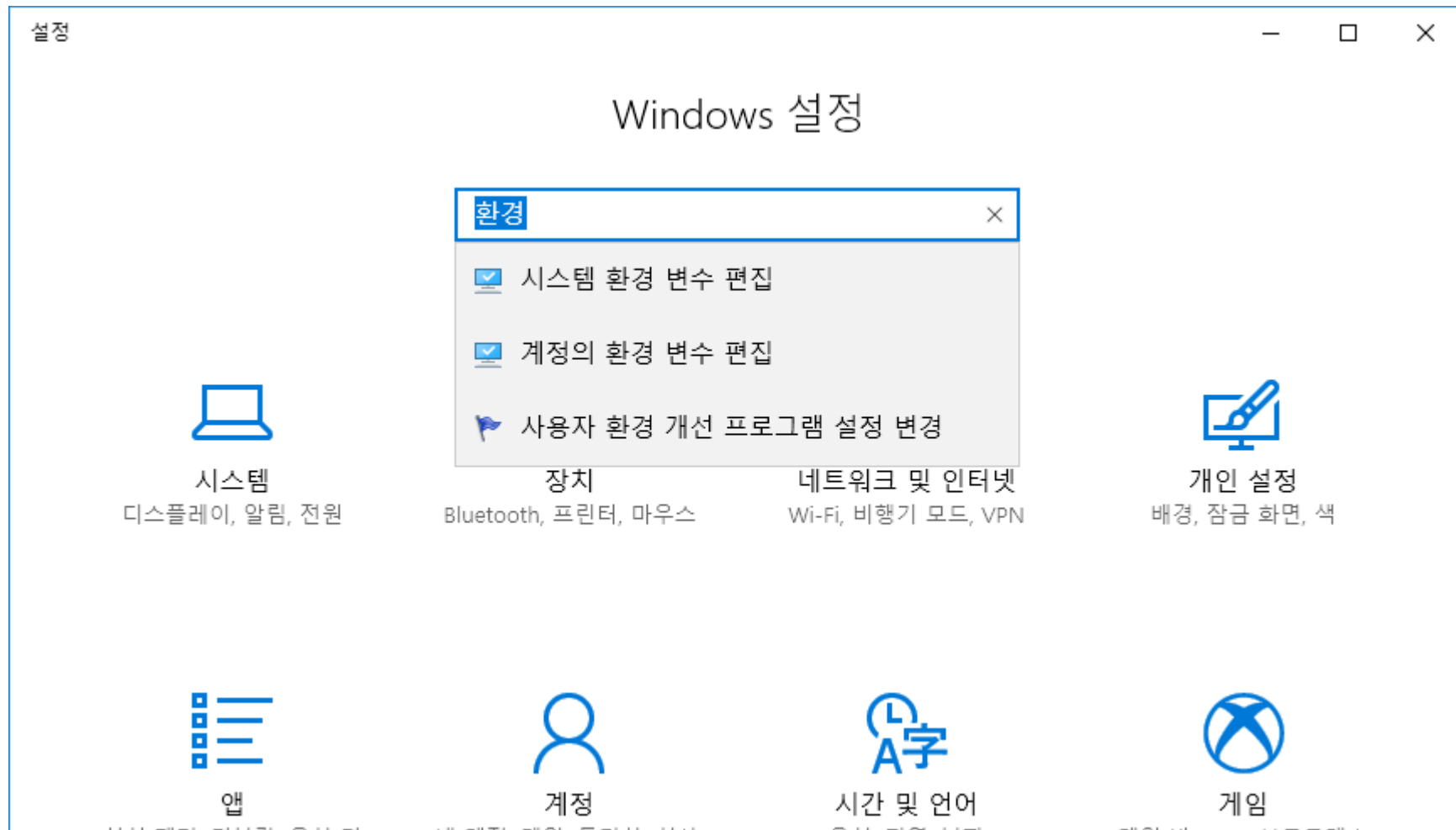


64비트의 경우: 환경 변수 PYSDL2_DLL_PATH 의 설정

관리자 권한으로 cmd 실행, setx PYSDL2_DLL_PATH "C:\Windows\SysWOW64"

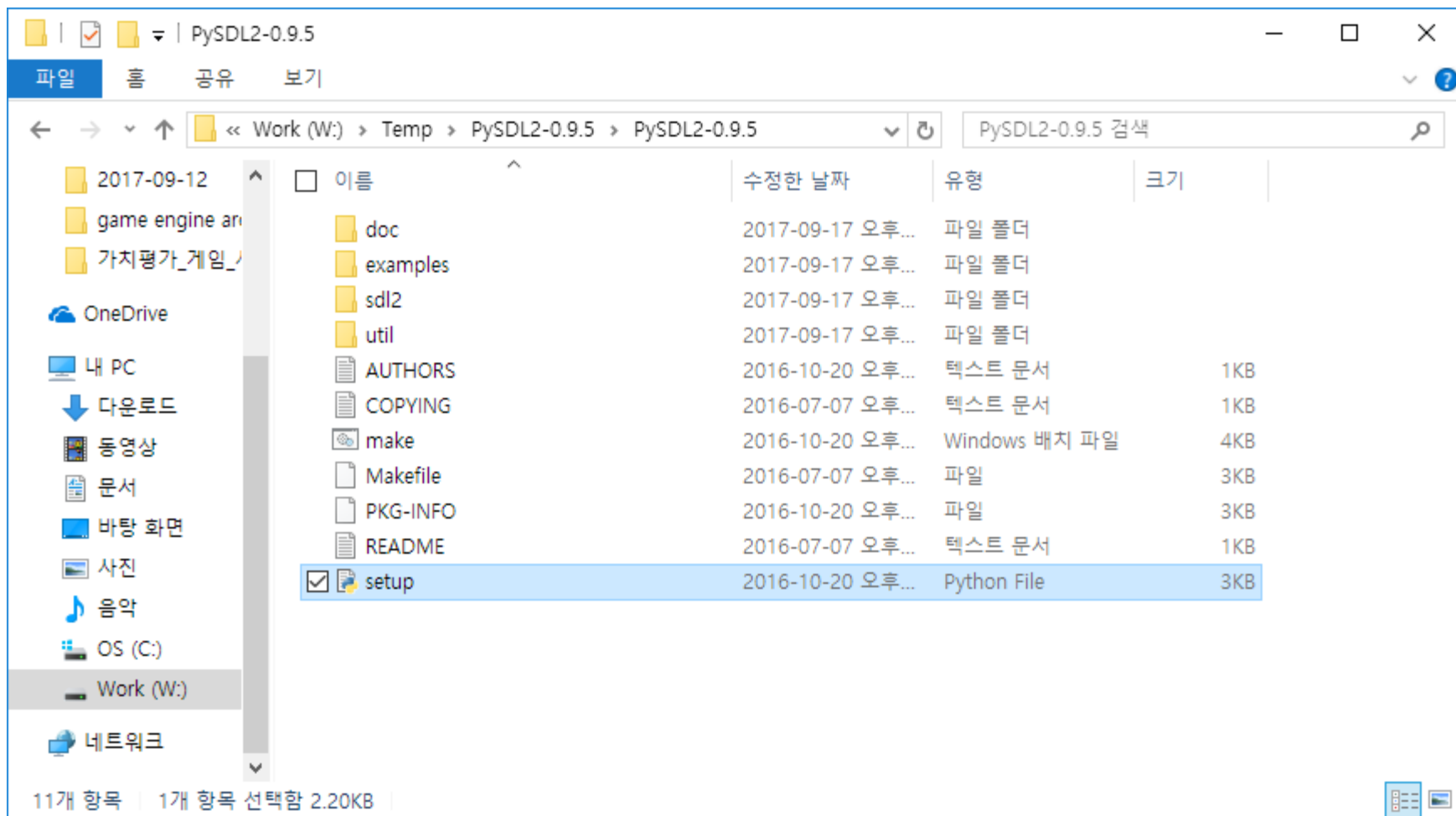


환경 변수 찾기

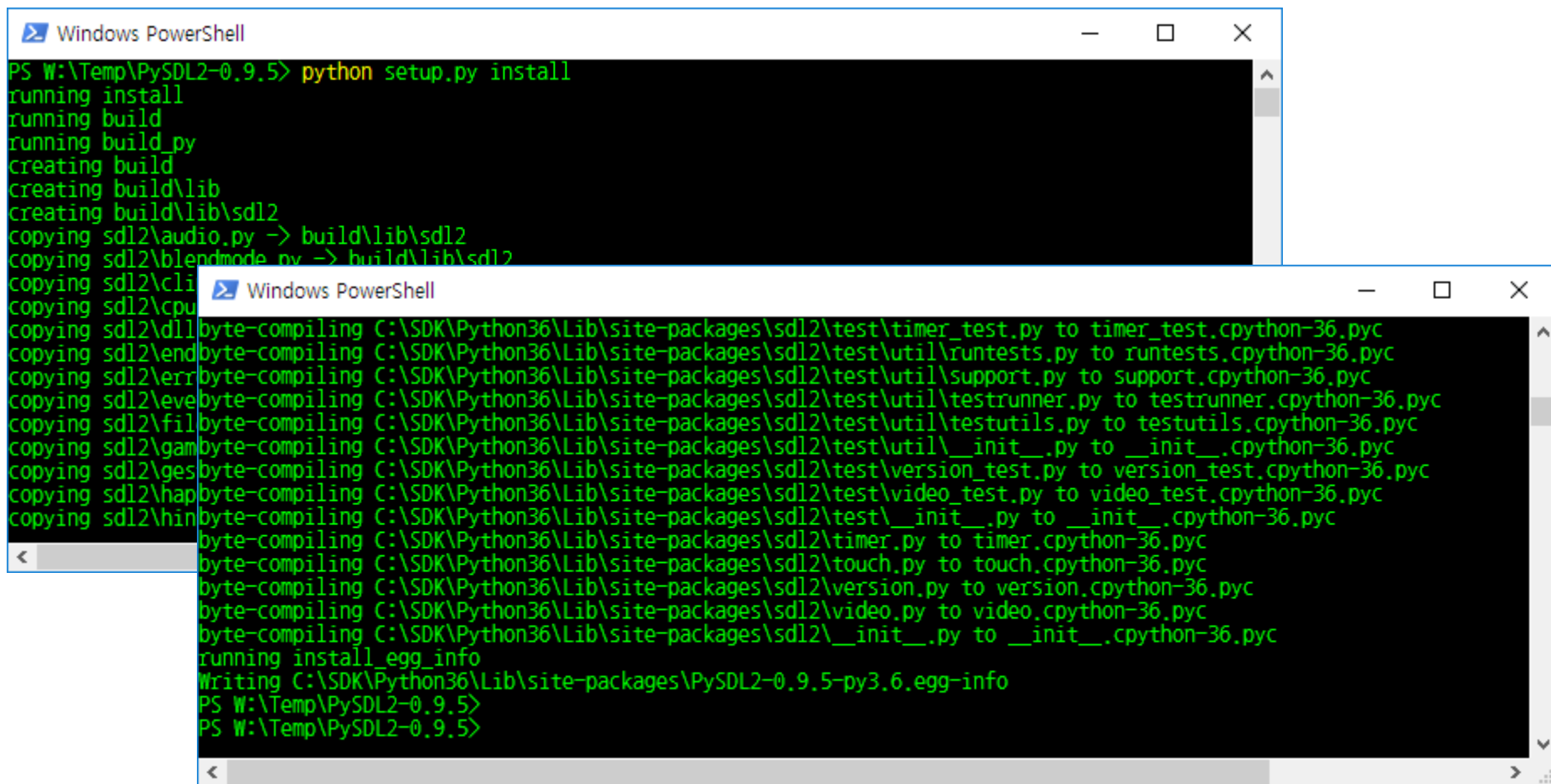


PySDL 설치

- 적당한 곳에 압축 해제
 - C:\temp

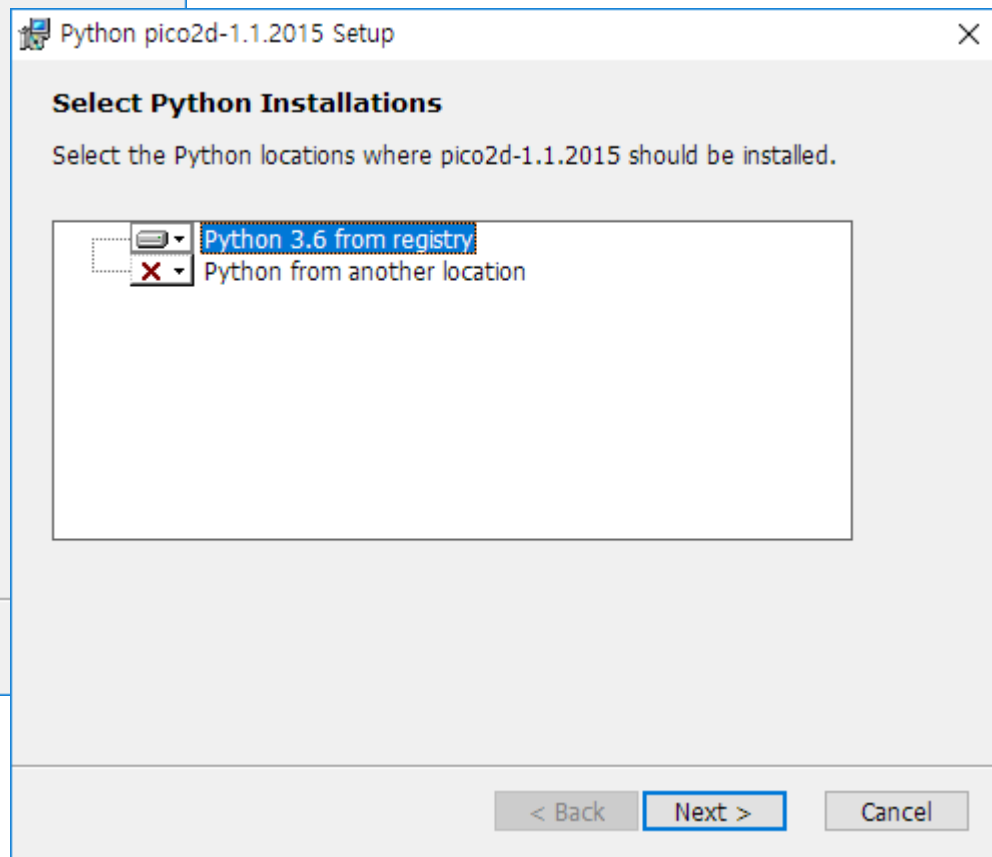
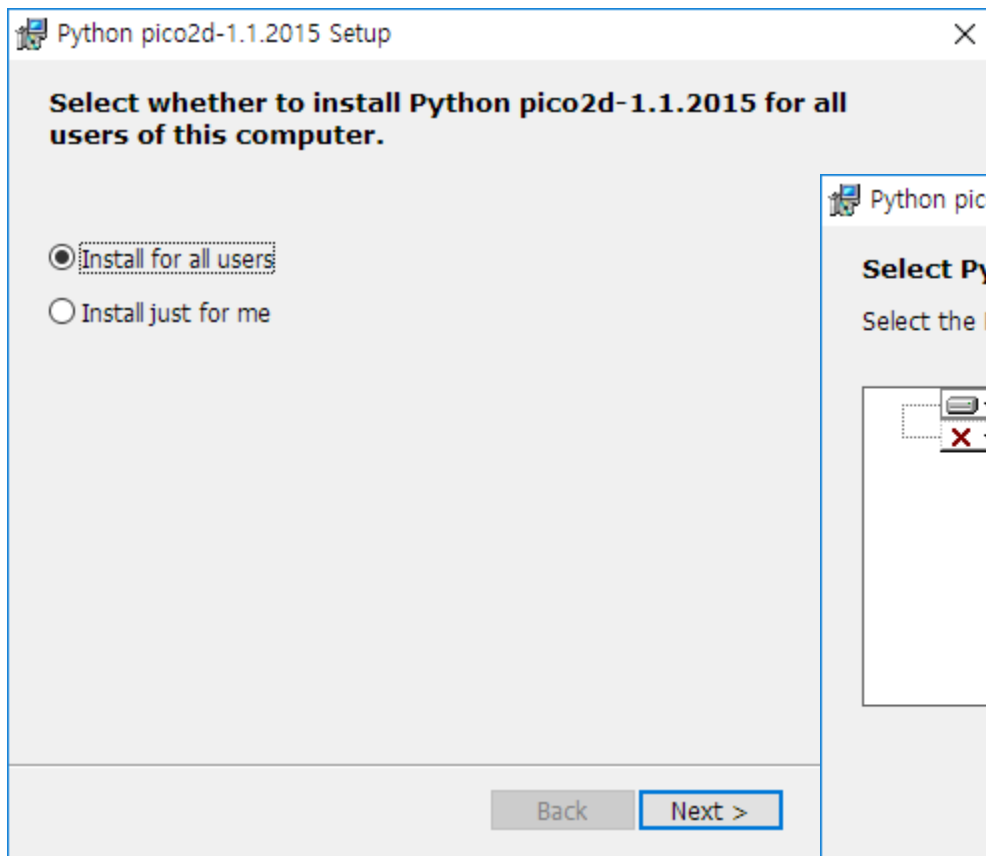


- cmd (코맨드) 창(또는 Windows PowerShell)을 열고
- 폴더를 c:\Temp\PySDL2-0.9.5 으로 변경한 후,
- python setup.py install

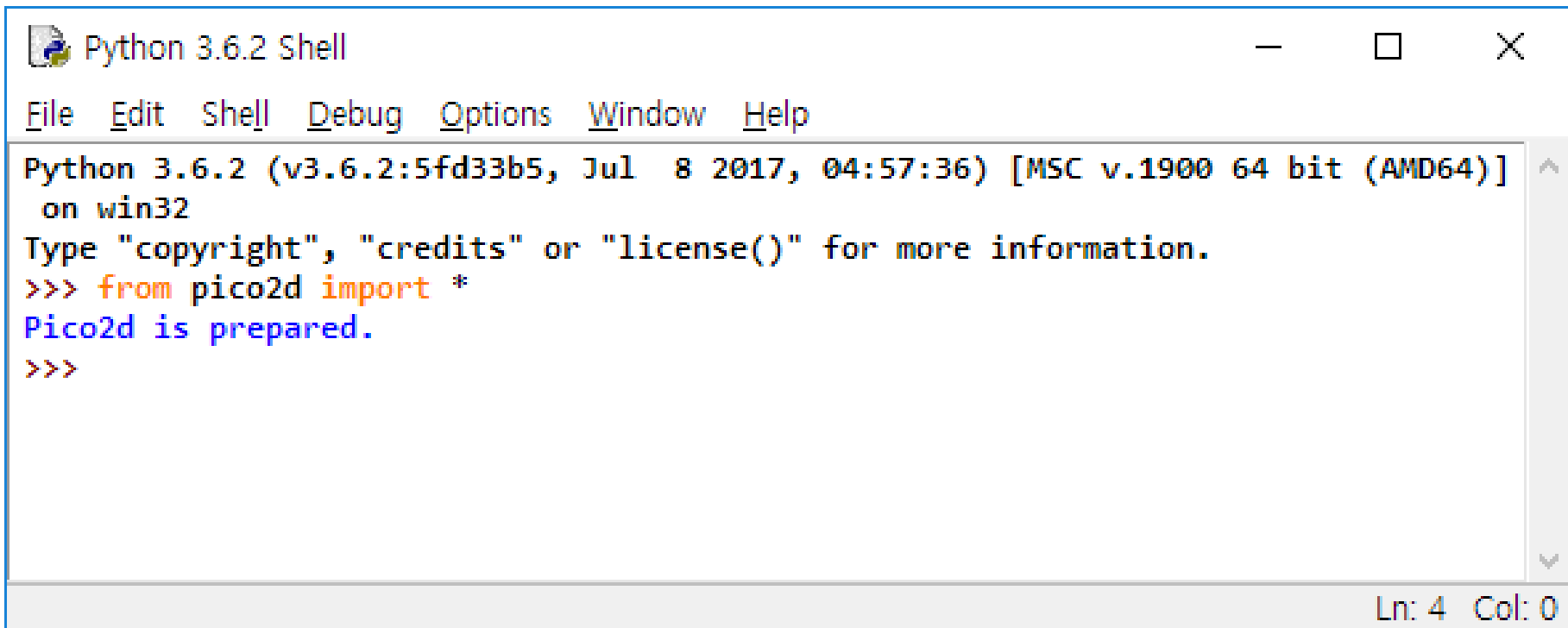


```
Windows PowerShell
PS W:\Temp\PySDL2-0.9.5> python setup.py install
running install
running build
running build_py
creating build
creating build\lib
creating build\lib\SDL2
copying SDL2\audio.py -> build\lib\SDL2
copying SDL2\blendmode.py -> build\lib\SDL2
copying SDL2\cli
copying SDL2\cpu
copying SDL2\dl
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\test\timer_test.py to timer_test.cpython-36.pyc
copying SDL2\end
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\test\util\runtests.py to runtests.cpython-36.pyc
copying SDL2\err
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\test\util\support.py to support.cpython-36.pyc
copying SDL2\eve
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\test\util\testrunner.py to testrunner.cpython-36.pyc
copying SDL2\fil
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\test\util\testutils.py to testutils.cpython-36.pyc
copying SDL2\gam
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\test\util\__init__.py to __init__.cpython-36.pyc
copying SDL2\ges
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\test\version_test.py to version_test.cpython-36.pyc
copying SDL2\hap
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\test\video_test.py to video_test.cpython-36.pyc
copying SDL2\hin
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\test\__init__.py to __init__.cpython-36.pyc
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\timer.py to timer.cpython-36.pyc
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\touch.py to touch.cpython-36.pyc
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\version.py to version.cpython-36.pyc
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\video.py to video.cpython-36.pyc
byte-compiling C:\SDK\Python36\Lib\site-packages\SDL2\__init__.py to __init__.cpython-36.pyc
running install_egg_info
Writing C:\SDK\Python36\Lib\site-packages\PySDL2-0.9.5-py3.6.egg-info
PS W:\Temp\PySDL2-0.9.5>
PS W:\Temp\PySDL2-0.9.5>
```

pico2d-1.1.2015.win-adm64.msi 설치



Pico2d 라이브러리 설치 완료 확인

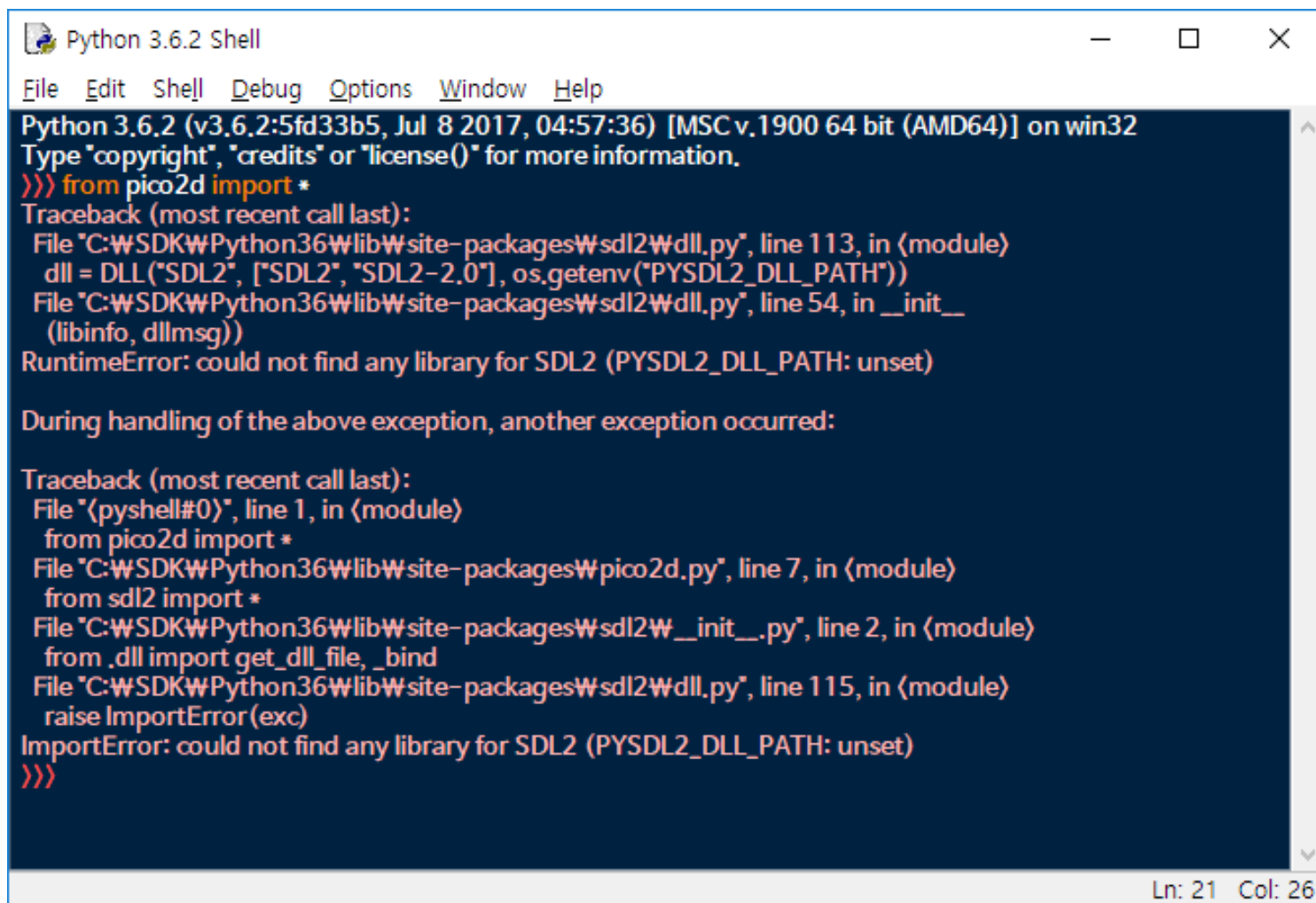


```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> from pico2d import *
Pico2d is prepared.
>>>
```

Ln: 4 Col: 0

Error?

- 환경변수 PYSDL2_DLL_PATH 지정 확인 필수.



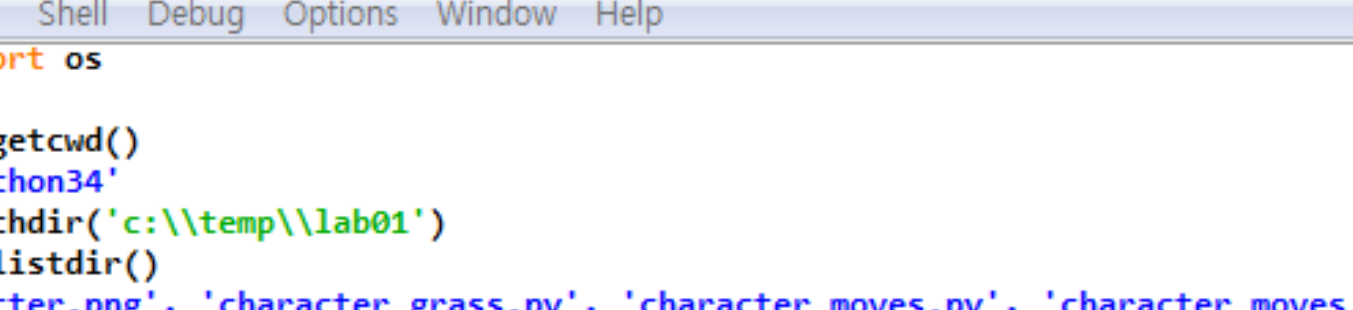
```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> from pico2d import *
Traceback (most recent call last):
  File "C:\WSDK\Python36\lib\site-packages\sdl2\dll.py", line 113, in <module>
    dll = DLL("SDL2", ["SDL2", "SDL2-2.0"], os.getenv("PYSDL2_DLL_PATH"))
  File "C:\WSDK\Python36\lib\site-packages\sdl2\dll.py", line 54, in __init__
    (libinfo, dllmsg))
RuntimeError: could not find any library for SDL2 (PYSDL2_DLL_PATH: unset)

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    from pico2d import *
  File "C:\WSDK\Python36\lib\site-packages\pico2d.py", line 7, in <module>
    from sdl2 import *
  File "C:\WSDK\Python36\lib\site-packages\sdl2\__init__.py", line 2, in <module>
    from .dll import get_dll_file, _bind
  File "C:\WSDK\Python36\lib\site-packages\sdl2\dll.py", line 115, in <module>
    raise ImportError(exc)
ImportError: could not find any library for SDL2 (PYSDL2_DLL_PATH: unset)
>>>
```

Ln: 21 Col: 26

OS 모듈을 이용한 Working Directory 설정

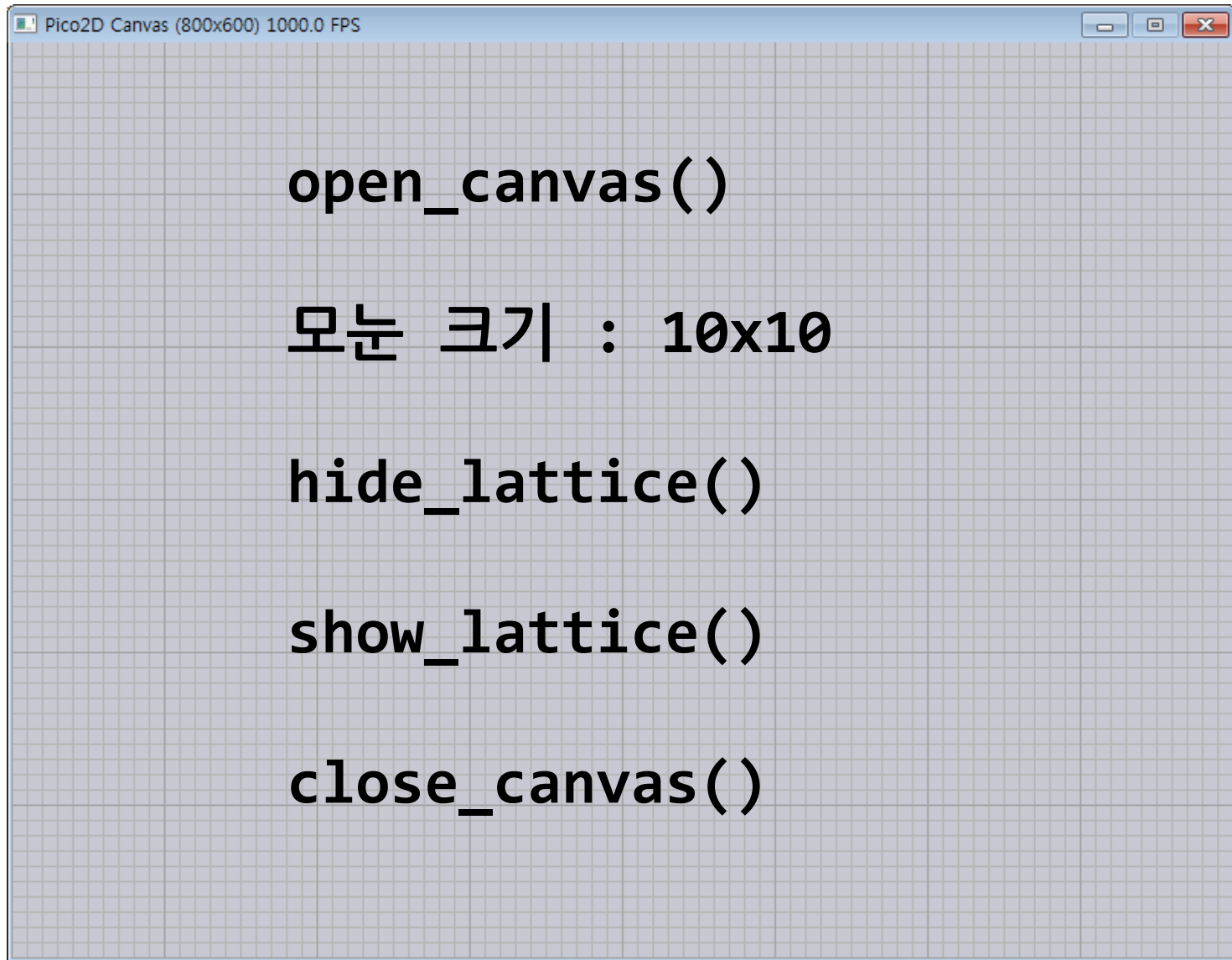


The screenshot shows a Python 3.4.3 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and standard window controls. The command prompt shows the following sequence of commands and outputs:

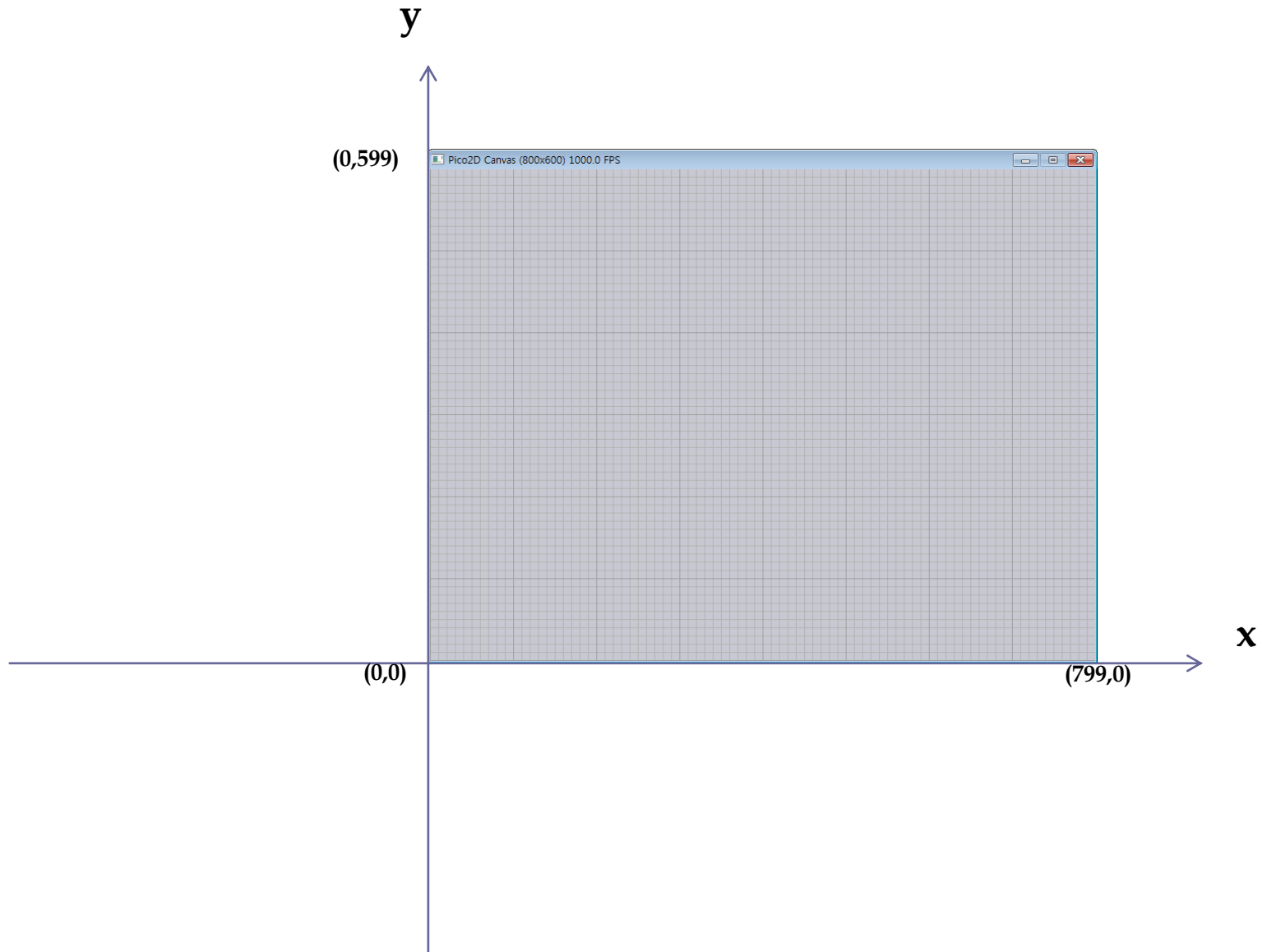
```
>>> import os
>>>
>>> os.getcwd()
'C:\\Python34'
>>> os.chdir('c:\\temp\\lab01')
>>> os.listdir()
['character.png', 'character_grass.py', 'character_moves.py', 'character_moves_recta
ngularly.py', 'grass.png', 'pico2d.py', '__pycache__']
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```

The status bar at the bottom right indicates the cursor is at line 32, column 4.

캔버스 열기 - open_canvas(800,600)



캔버스의 좌표계



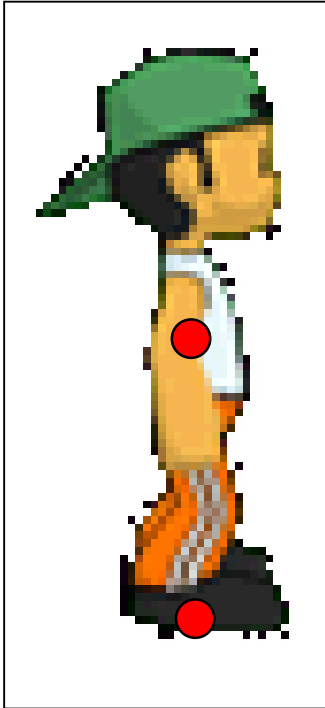
JPG vs PNG

우리의 주인공



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

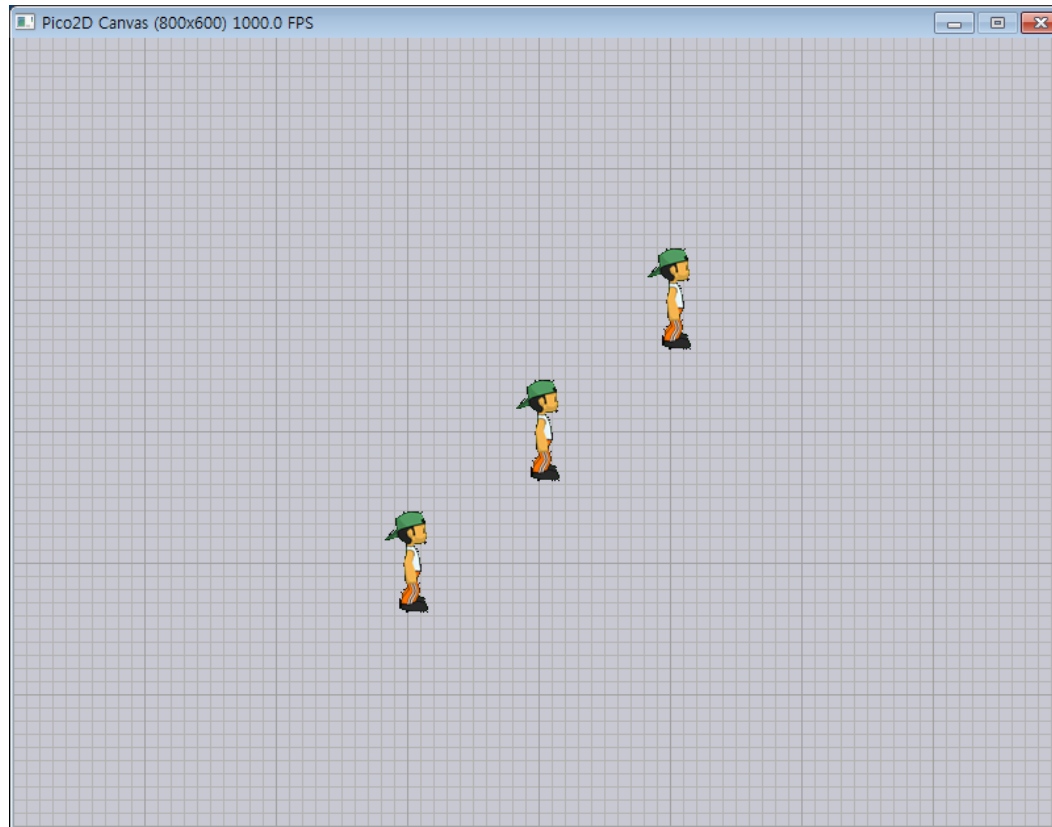
피봇(Pivot)



여기가 피봇입니다.

이 점을 피봇으로 삼기도 합니다

몇 명 더 그려 봅시다~

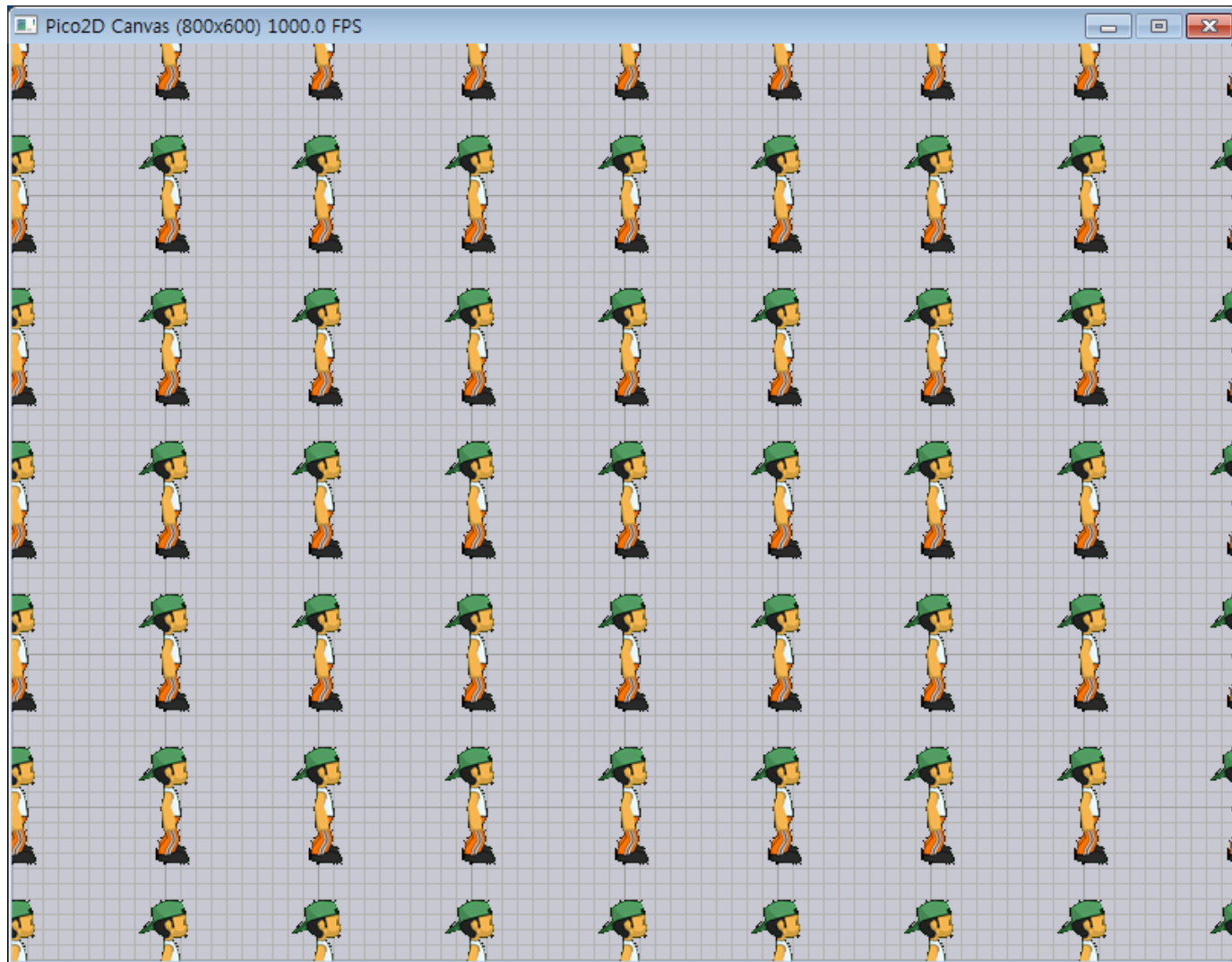


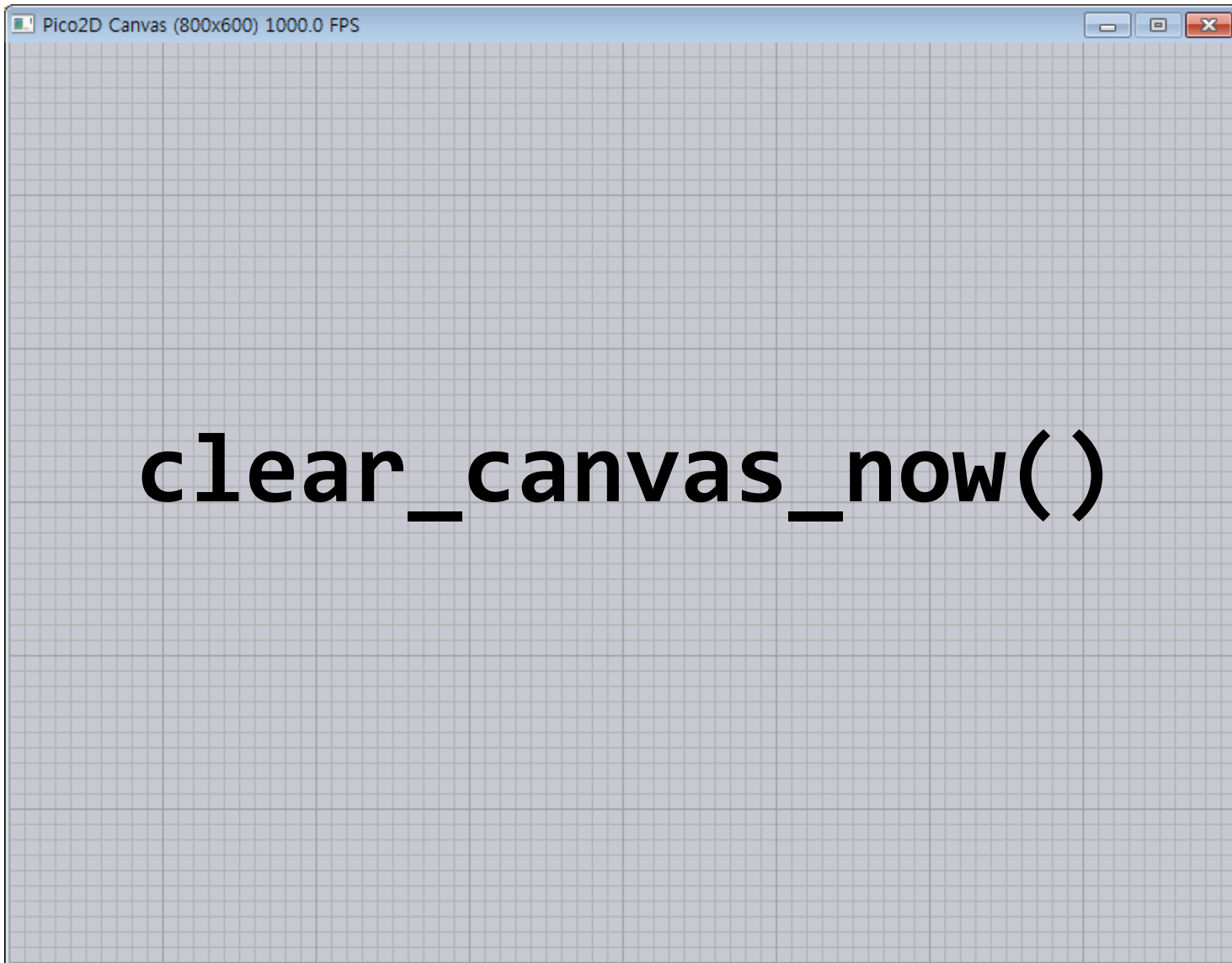
```
>>> image.draw_now(300,200)  
>>> image.draw_now(500,400)
```

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')
```

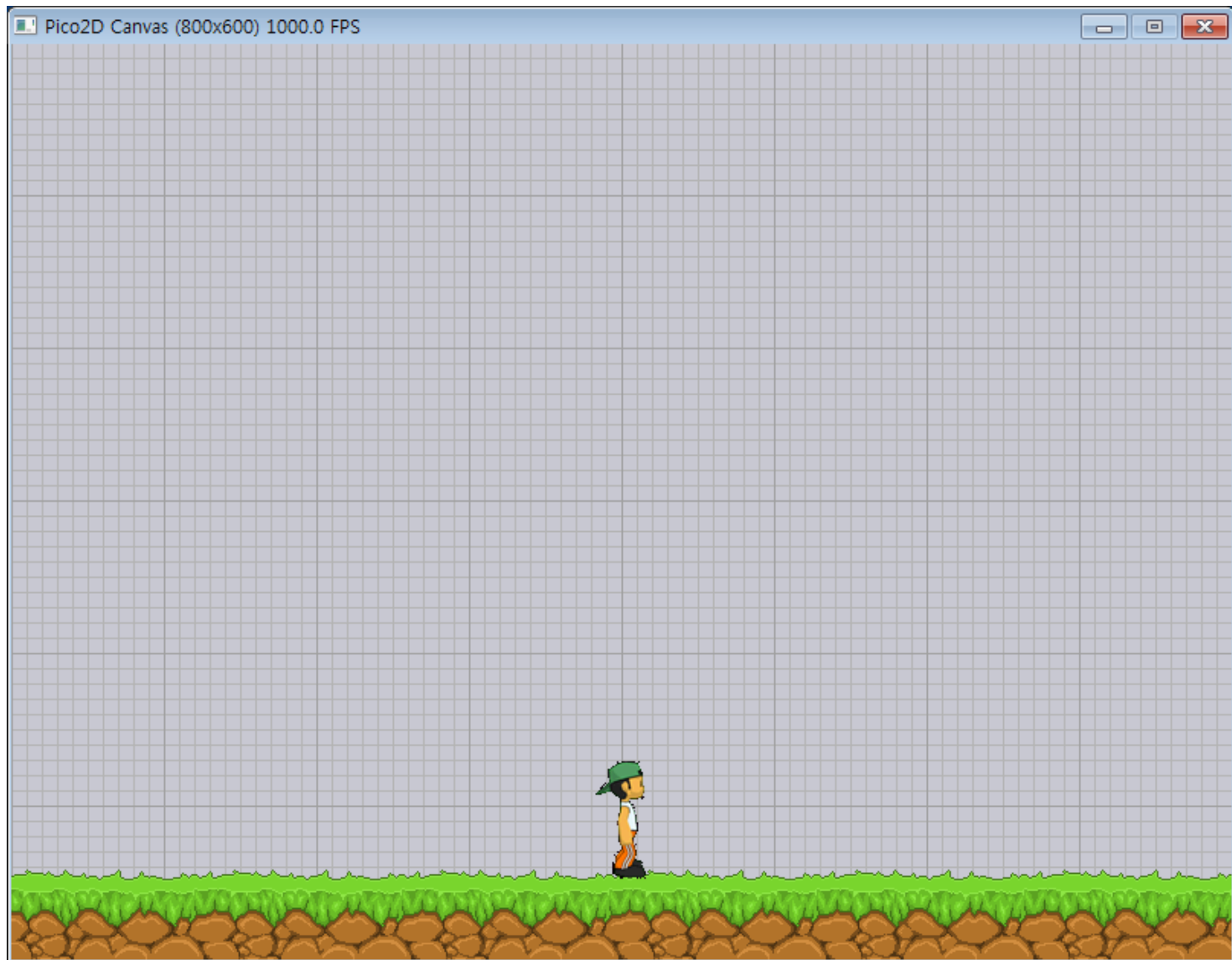
```
grass.draw_now(400, 30)
```

```
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)
```