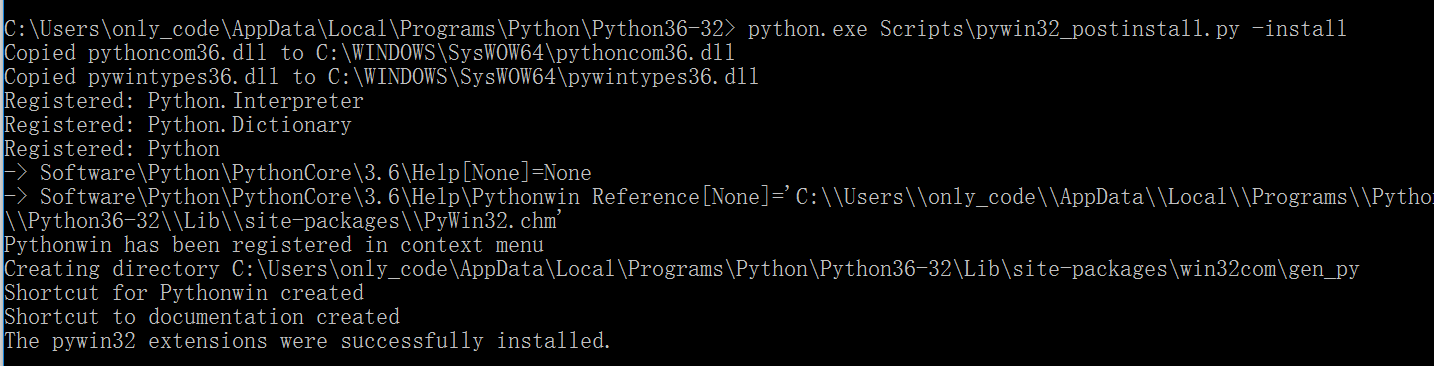
**Step 1: Environment configuration for keyboard simulation**

After searching how to simulate keyboard input by running a program, I decide to use a package called PyUserInput for its cross-platform property and comprehensive functions.

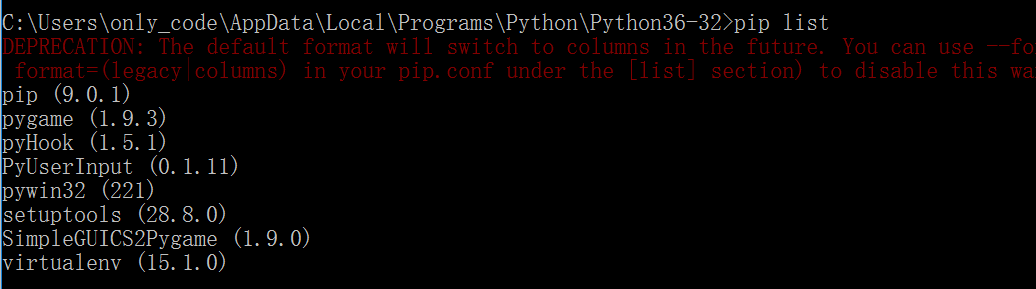
I firstly tried “pip install PyUserInput”, a commonly used tool to install packages, but failed for “Could not find a version that satisfies the requirement pyHook/pywin32”. (PyUserInput requires pyHook and pywin32 and there isn’t any available version for me). So I download the .whl file from <http://www.lfd.uci.edu/~gohlke/pythonlibs/> and pip install pyHook locally.



Further I followed the instructions on that website and install pywin32 using elevated command prompt.



Then I finished the first step. Now I have all environments that I need.



The relationship among this three packages.

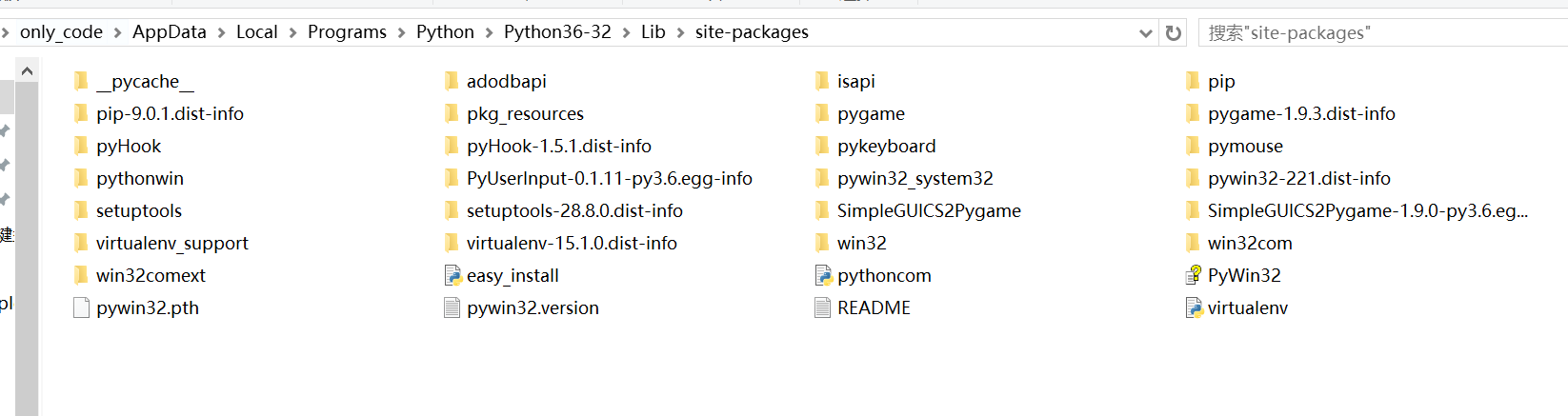
pywin32

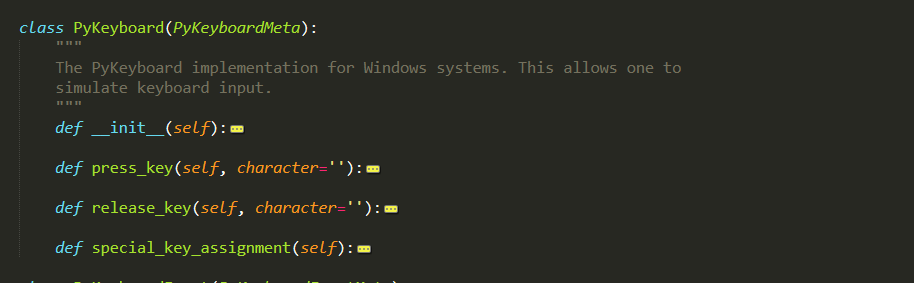
pyHook

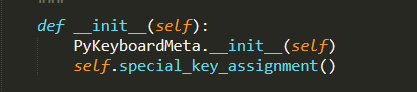
PyUserInput

**Step 2: Learning PyUserInput**

By checking the files under the install directory of python, I found that two classes, or objects, which are pymouse and pykeyboard, make up the core of PyUserInput.



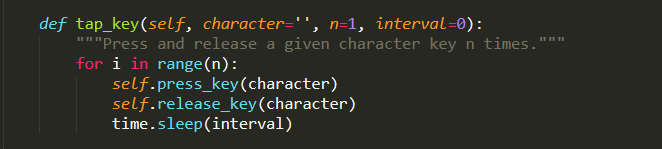
To play pokermon, I will mainly use pykerboard. Thus I pay the most attention on pykeyboard.



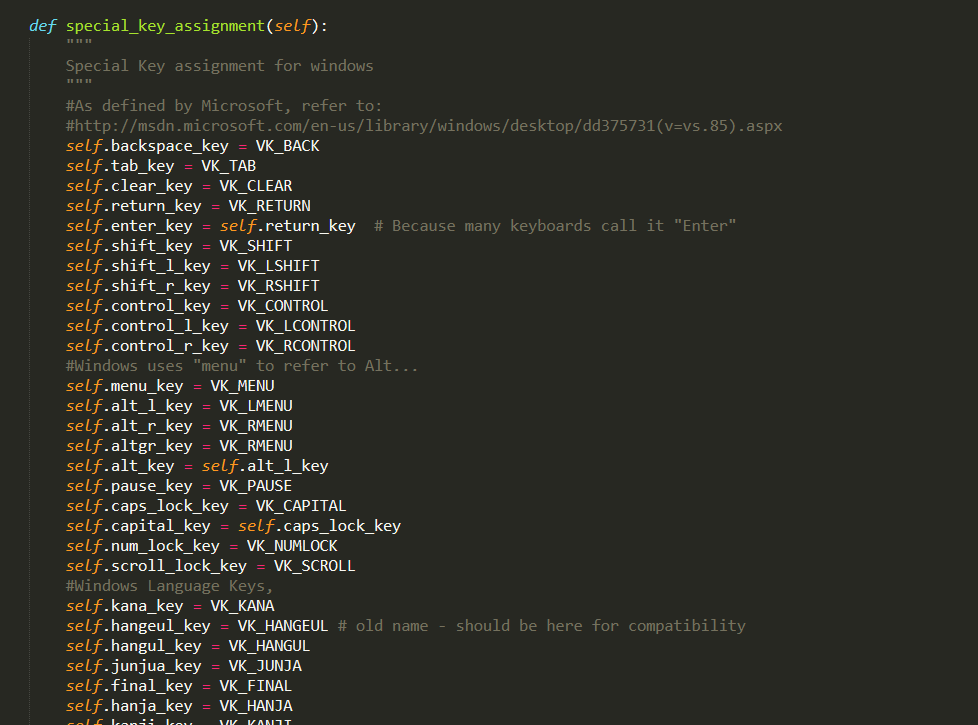
I found that the core of PyKeyboard is another object called PyKeyboardMeta and pre-treatment to special keys.

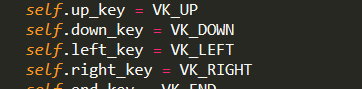


My need is simulating the keyboard input, so I focused on the function “tap\_key()”

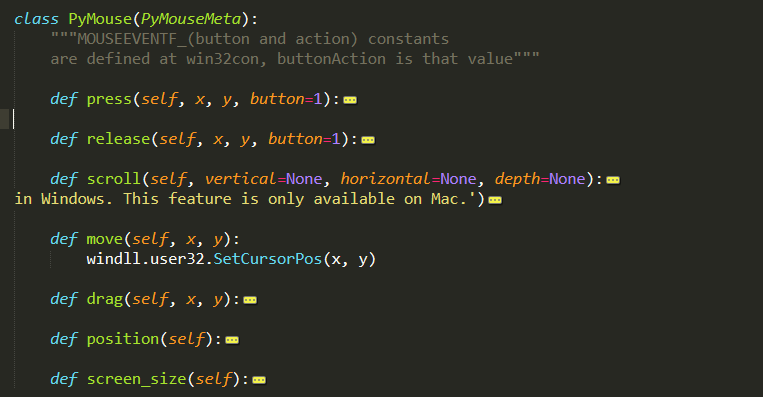


k.tap\_key(“H”) #tap H



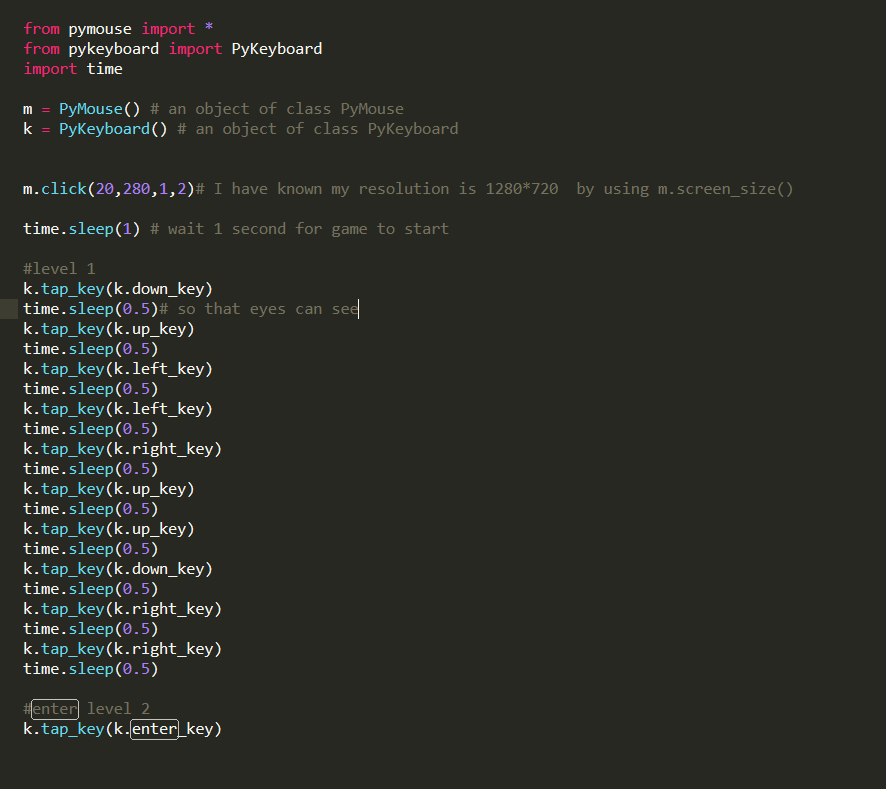


k.tap\_key(k.down\_key) #tap down

pyMouse also defines many methods, I don’t need to know how these methods works because of capsulation, a concept of not showing every details to the users.

I focus on the function “click” to manipulate the click of mouse. (not shown in the screenshot)

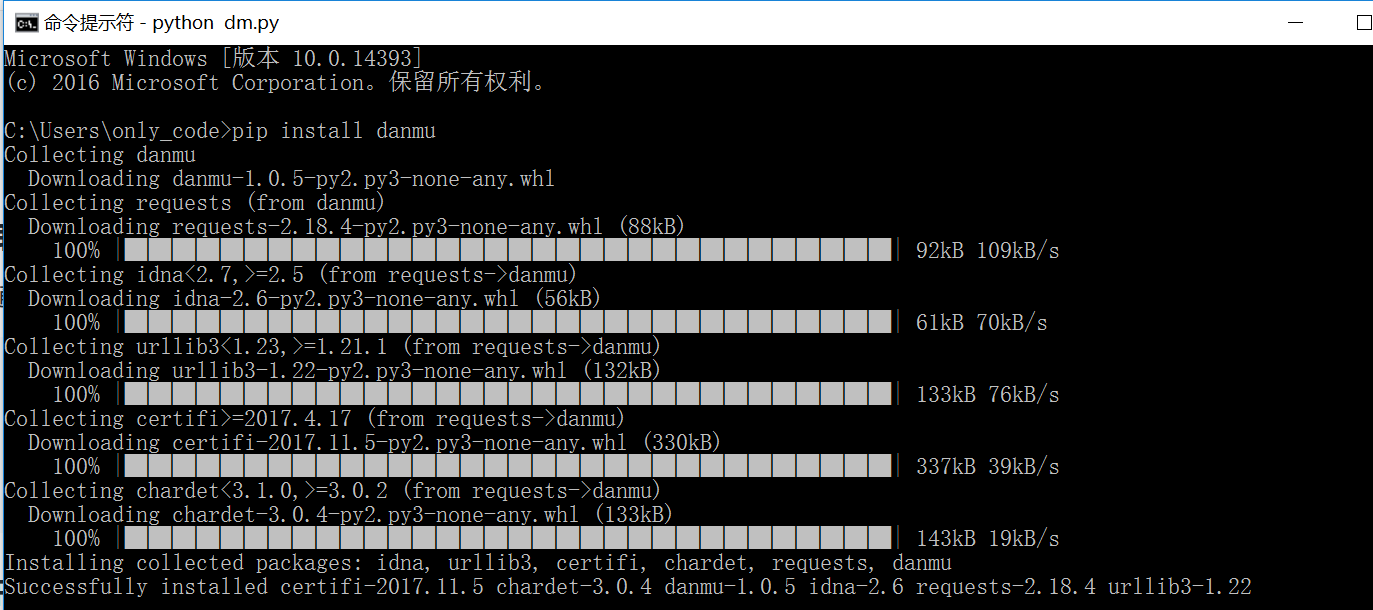
m.click(x,y,n,z) #click button n (1 for left and 2 for right) for z times at point (x,y)

For practice, I wrote a program to open Sokoban on my computer and pass level 1. 

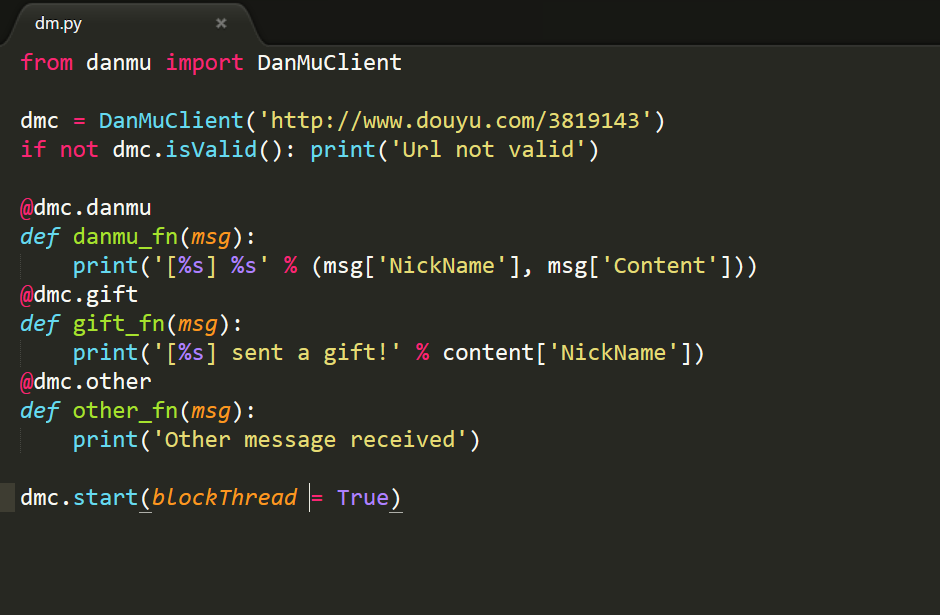
**Step 3: Collect Danmu from Douyu**

I choose to use package “danmu” ( it’s a pinyin and means the messages that are sent to a live channel)

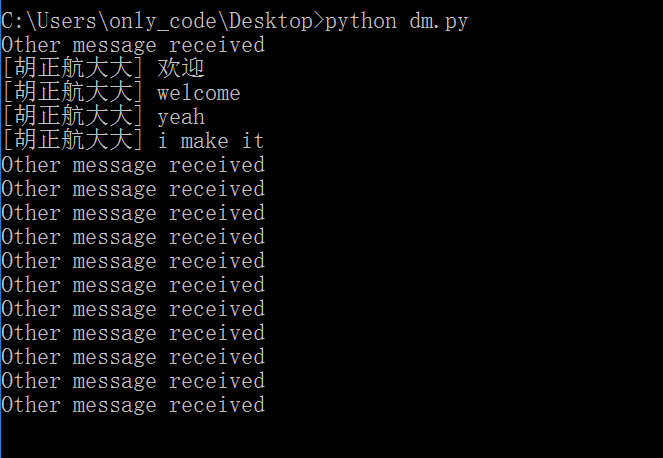
It’s open source: <https://github.com/littlecodersh/danmu>



A program to get danmu :



How it shows on my terminal:



I revised the program to to collect danmu from Douyu and use these commands to play Sokoban.



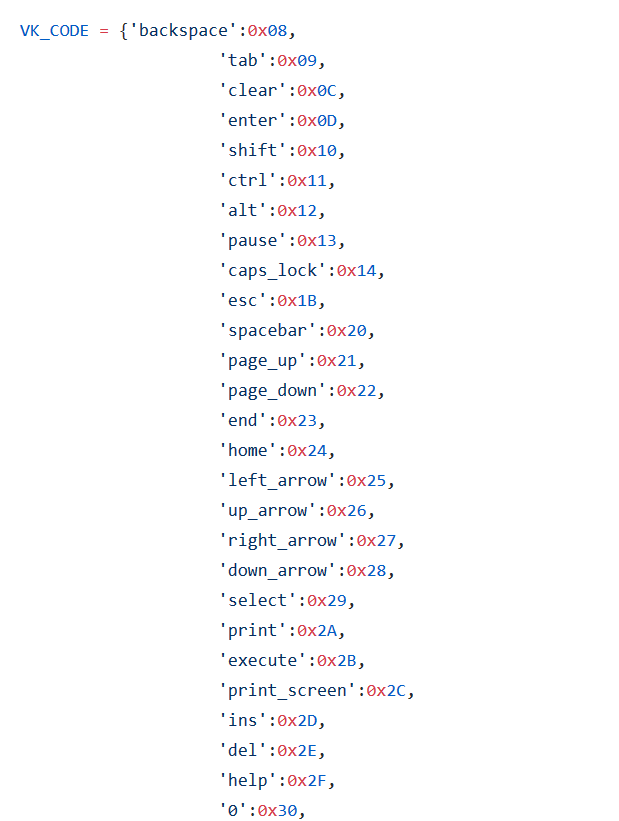
**Step 4: solve an unexpected problem**

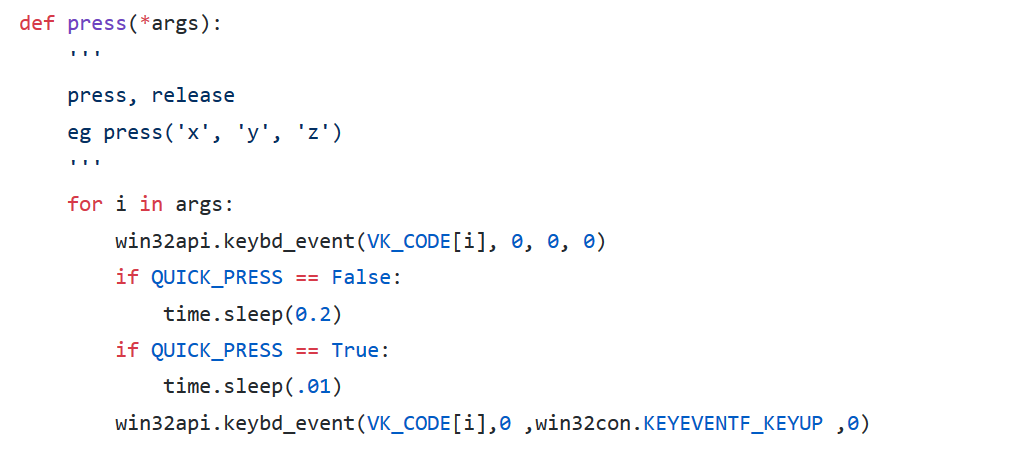
Although I successfully play Sokoban, the program fails to simulate keyboard input when I tried to play pokermon.

I think about two reasons:

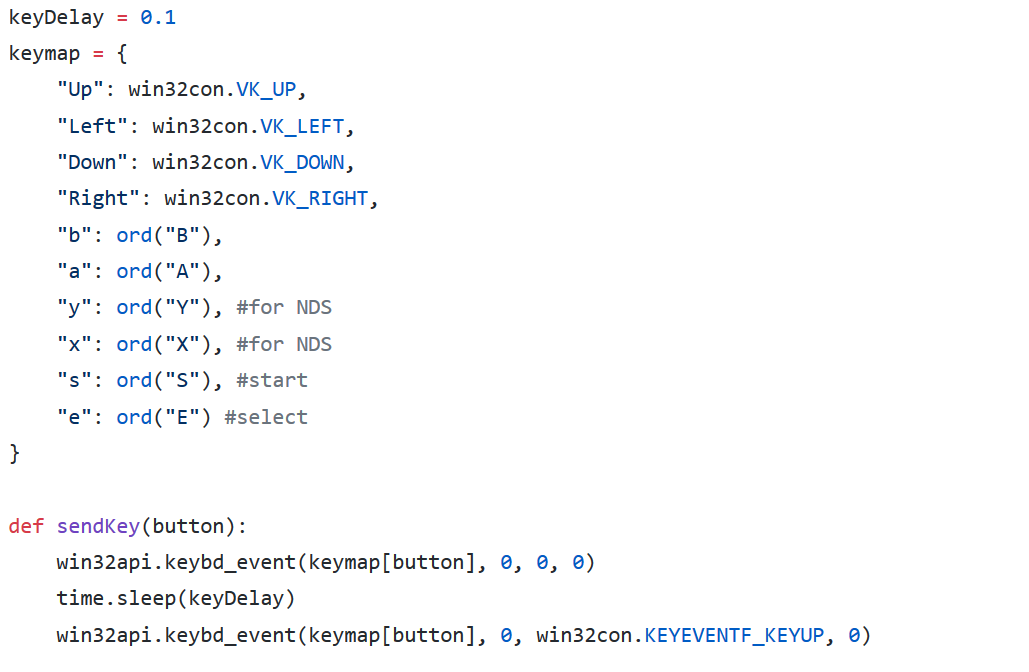
1. Something went wrong with the simulator, VisualBoyAdvance , to be specific, that I use to play pokermon. So I tried another game, 2048, which doesn’t need simulator to play. The program still doesn’t work. So I eliminate the cause of simulator.
2. I think about that PyUserInput requires pyHook and pywin32. In another word, PyUserInput is the extended development from pywin32. Something may go wrong during this secondary development. In addition, I remember that pywin32 is a commonly used package in Github to solve virtual keyboard input problem. Therefore, I decide to use pywin32 directly.

First example:





Second example:



Another example shows how PyUserInput uses pywin32:



I found the pattern that firstly use a dictionary, a data structure in python, to store all the pairs of keys and danmu, then use function “win32api.keybd\_event” to simulate keyboard input.

After revising the code, my program works to play pokermon.



**Step 5: optimization**

Consider the fact that my channel may become popular and the program can’t deal with all the danmu(It would be too many) , I revise my program to consider one valid danmu command for one second.

