L0 — Important Issues in the Experiment

181240020 Hu Junhao

February 27, 2020

1 Encounter AM Again and Embrace it

It is really great to reread codes in AM because when I was doing PA, I did not fully understand what it is. Now, except some technical details in the code itself, I have totally figured out what AM is and what it is capable of thanks to my previous experience in writing a game "alien_invation" using Python.

Here, except the function we called, other details are almost the same — When we want to read keycode from the keyboard, we simply call the function read_key() or pygame.event.get() in Python; When we want to draw what ever on the screen, we call draw_sync() or screen.blit() in Py; When we want to check the time and use it to make decisions, we call update()......

Although AM does not provide so many "powerful" APIs as those in Python, however, the basic idea is the same and actually, all those seemingly powerful APIs in Python is in the end realized by calling the basic APIs similar to those in AM.

2 Arrange My Programm

This small program I wrote is not that demanding in the style we arrange our functions and files. However, I was deeply impressed by the difficulty of debugging and upgrading my poorly written "alien_invasion" in Python.

That is why, although this program is small, I still seperate each functions and even store some of them in different .c files.

In this way, the structure of our program is much clearer and it is possible for future upgrading in an easy and smooth fashion.

2.1 update_screen()

Although in this small program, the only thing we are going to draw on the screen is a ship with a shape of square, we did not only write a single draw_ship() function. Instead, we wrap it up with update_screen() anyway. In this way, later, when we wang to add in some bullets or aliens falling from the top, we can write a seperate function to blit them and also wrap the function up in update_screen().

2.2 check_events()

Again, although we only need to check keyboard events for this experiment, we wrap it up in check_events() anyway. In the future, we might check if the mouse is clicking. Then, we can write a seperate checking functions for mouse clicking and wrap it up within check_events().

2.3 Settings

Initially, we need to specify the initial position of the ship, its shape, size etc.. We avoid using numbers directly and instead use a structure called settings to specify those parameters. In the future, when we want to add in functions like “ship upgrading!!”, we might increase the speed or the size of the ship. And all we need to do is to change the corresponding settings.

3 Complaints

It is a little bit painful to write games using C after I have been so used to Python. Everything is just so rudimentary!! And also that is why I just stop here, only aiming to finish the experiment instead of really writing a big interesting game using C.Therefore the C-version of “alien_invasion” is rather of poor quality.

4 Bugs

Luckily, in this small program, I did not encounter any bugs and gained the one-time-pass achievment.