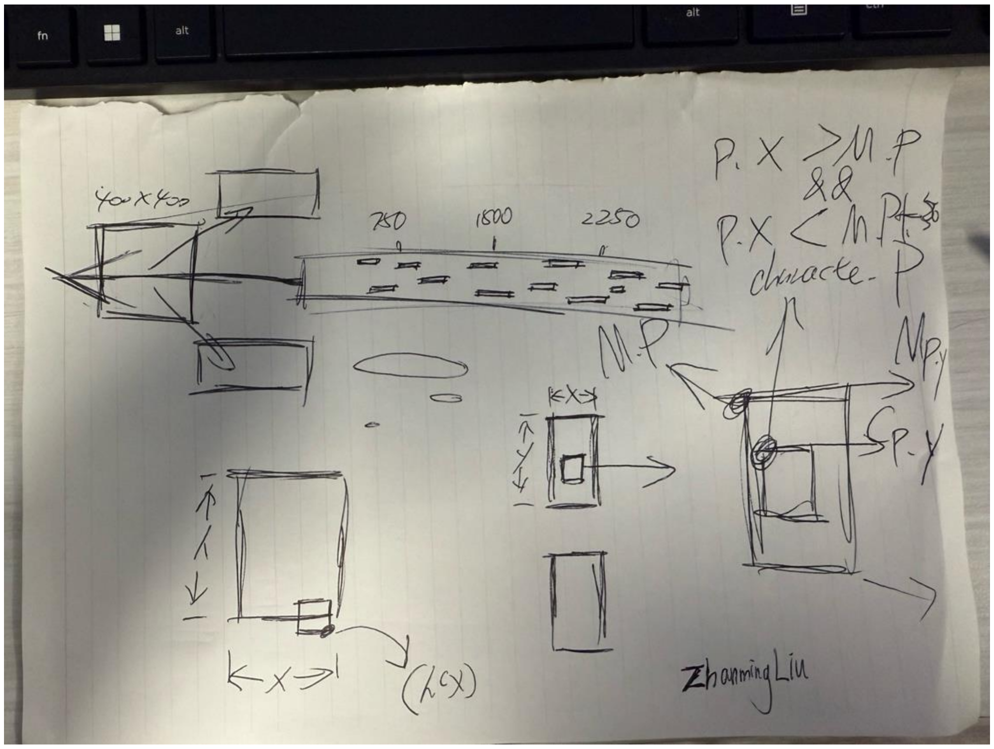
1: The Touch Determination(or judgement) Between Character and Machines.



First, when the two touch, the player will lose a certain amount of health and accelerate towards gameOver. How to determine the contact range of the two graphics?

The biggest difficulty lies in the fact that both are different classes, and I need to determine it in the Main Tab. To indicate the value in other tabs, I need to use the "." to connect codes and claim it.

The range of damage to the player is a central rectangle, so the determination thought is that if the machine's large rectangle covers the character's rectangle, it will lead to a successful determination.

Therefore, the values I need to consider are: the position of both PVector. First, in my class's function, I declared that the x and y values of position are the coordinates of the graphics. Because velocity and acceleration all act on position, the value of position is the coordinate position of the moving graphics, which is the determination value I need.

Next, I will compare them in the main page. In the main page, if I need to use the class codes of other tabs, I need to use very specific codes such as "myCharacter.position.x". Then I can use myCharacter.position.x and y, as well as badMachine.position.x and y for comparison. Because the large rectangle needs to wrap or cover the positions of the small rectangle, both x and y need to be compared. For example, the x value of the character is greater than the left value of the machine and less than the right value of the machine.

When both x and y are determined to be successful, I have achieved the comparison of determination range.

Next is a related problem. Because badMachine will move faster and faster as the game progresses, the hit time of the player will become shorter. My codes implement that the player receives damage within the range of the machine rectangle, so the longer the time, the higher the damage received. Now that the time is shortened and it is the latter stage of the game, I also need to increase the difficulty of the game. So I achieve this by increasing the loss of health according to a certain score.

I use a simple method, using an if statement and scores to determine the damage received. So if a certain score is reached, higher damage is received. I have set up three different types of damage and 18 scores after one touch die!

2: The challenge of creating a background with strong airflow.

This was my first PVector setup.

I wanted to achieve the effect of a background that was a long rectangle both inside and outside the screen. I achieved the feeling of airflow by having the rectangle move horizontally and put it back with the "%" code.

Although the screen is 400*400 pixels, the moving background still runs and is displayed on the screen.

My parameters: acceleration adds an x value to velocity, and velocity adds an x value to position. So, as the game progresses, not only will the machines move faster, but the background (sky and clouds) will move faster as well.

At first, I drew a PNG cloud in Procreate. But when I tried to put it into the code, the result was poor and I abandoned the image. Next, I used long, thin white ellipses to represent the clouds.

Now, let's talk about PVector. I originally wanted to use a single, long enough rectangle to represent the beginning and end of the game. If the rectangle moves to the far left, it represents a game victory, so I didn't use the "%" code. However, the problem arose because of the existence of acceleration and velocity, which made the very long rectangle disappear in an instant. So, I still needed to use scores and the "%" code applied to the rectangle to achieve the background's constant change.

Next, I set up the PVectors and drew the rectangle and clouds. I brought the update and display functions to the main tab. And finally, I achieved the ideal background movement! I could then add other objects and elements that I wanted.

3: The game is reset (reset game).

This is the last problem I encountered and the most challenging one.

According to the brief, I need to implement the reset game mechanic when the game is over or won. How can I achieve this?

I tried using the new function of a class, but it didn't work. I also tried whether I could completely replace the running code with all the codes, which sounds crazy! It is also impossible to achieve and may cause a burden on the code execution.

In the end, I found that my class functions, such as display and update, are the main codes that run. So I need to reset all the important variables and codes (or functions)!

In my code, scores determine the game's victory. Because scores keep increasing (even after the game is over), I set the scores to zero when the game is over. Therefore, scores must be reset to 0, otherwise the following codes cannot run. This also includes the health bar and other variable values that need to be reset. After I reset all the necessary functions and variables, I found that the game was frozen. This made me very frustrated and I thought about it for a long time.

Since this is an if statement that determines the "gameRestart" boolean, if it is true, the if statement will keep running. This means that scores are always 0, health is always at its maximum value, position is constantly reset, and all values are reset!

Therefore, my solution is to make "gameRestart" false at the end of the if statement (after all the resetting is completed), so that the condition is over, the game is reset and restarted.

In this way, I provide a method to activate game reset for both gameWin and gameOver, so that players can play my game infinitely!