# Overview :

My program is used to conduct a game. The player can press up, down, right, left, space or click to control the main character. In the beginning, the player will see an introduction of my game. After that, the player can click anywhere to start the game. The main character is a snake, and it can eat food and become bigger. There will be a monster who follows it. If the monster touches the snake's head, the player will lose.

On the other hand, if the snake eats all the food and starch to the whole length, the player will win. If the snake eats the food, the snake will become longer as the step is taken. The details will be introduced afterwards.

# Data model:

In my program, I mainly use the turtle.Turtle() data model. Besides, I also use the Boolean and integer, 2D vector, and list.

# Program structure:

Find the random place of the fruit

Start game.

The snake is going anywhere, and the monster is chasing the snake. The snake should not move out of the map. I need to judge who is win.

# Processing logic:

1. snake moving introduction: the head of the snake is a turtle(). I use the stamp to copy the snake's head. If the snake length is the same as the stamp's number, it will first copy the stamp and then delete the last stamp.
2. As I mentioned above, if increasing, do not delete stamps.
3. Get the all stamps location, and compare them with the monster. If they are close enough, I will increase the contact number. Note that if the monster is on the body of the snake. The contact number will increase with the steps of the snake.

# Functional spec:

1. 'Drawmenu': I use this function to draw the margin and the map. The time, contact, and motion are also included in this method.
2. 'random': I use this model to get the random positions of the fruits.
3. 'Getclick': I use this method to get a click signal from the user. After getting that signal, the program will start the next move.
4. 'Drawtime': I use this method to count the time. Since the move of snake and monster are also related to time, I call a lot of other function.
5. 'Move’:I use this method to move my snake. To move my snake, I create a new stamp of my snake. Then, I also delete the last stamps of the snake if necessary. If it is not necessary which means the snake is growing. Then, I will not delete the last stamp.
6. 'Outofrange': To make sure the snake moves on the map, I create this method to return a boolean value.
7. 'GetDirection': To get the snake's direction, I use this method to get the input from the user. And return a string type value to the 'move' function.
8. 'Movemonster': To move the monster, I use this function. This method is almost as same as the 'move'. However, since my direction is pointing toward the snake, it will never get out of the map.
9. 'Finddirection': To find the direction of the monster, I use this method.

10.’Drawcontect’:To draw the times of contact. I use this method. First, I remember the location list of the snake. If the location is almost as same as that of the monster, the contact number will increase for one.

11.’Drawdirection': Draw the direction of the snake.

12.’colid': It will return the location of the colliding place of monster or fruit to 'judge' function

13.’Judge': To judge if a player is winning or lose and how long it should increase. If the return value is the location of fruit, then it will return grow value. Else, snake lose.

