1. Design

a. Describe your thoughts and overall approach, including the breakdown of your logic into sub components and how these sub components are related/connected (basically the structure of your program)

I wrote all the objects like snake, monster and foods in different functions.

I used ontimer() to allow the snake and the monster to move at the same time.

When the food is eaten, I cover the original food writing with the white writing in the same position, font and size, and delete its position data to make it unaccessible.

I used list to store many position data.

I used many flags to end some function.

b. Describe the data types that you use to develop your program for tracking the various game objects (snake, monster and food items)

Snake and monster are “turtle”. They keep moving.

g\_flag, g\_start\_flag, g\_monster\_flag and g\_title\_flag are “bool”. They are used to end the function.

g\_foods and g\_body are “list”. They are used to store position data of something hard to locate, like stamps and written foods.

g\_snake\_lenth, g\_snake\_extend, g\_cont and g\_time are “integer”. They are used to store integer data.

c. Describe the motion logic for both snake and monster.

In order to facilitate the game, the speed of the monster floats up and down around the speed when the snake decelerates while entending, and the speed is changed by the length of the single movement distance.

When the snake extends, its speed is halved, which is achieved by doubling the delay time.

d. Describe the expansion logic for the snake tail.

The tail is achieved by creating a static copy while deleting the oldest copy.

And the old copy is not deleted during the snake extension process.

Every time Snake creates a copy, it stores the location data in g\_body for title() to use.

e. Describe the body contact logic between the monster and the snake.

Check the distance between every position data in g\_body and the position of monster every second.

If the distance is to close, regard it as a contact.

2. Functions

a. Describe usage of all your newly defined functions, including details of parameter(s)

start(x,y): click, and start the game.

welcome\_words(): print the welcome words.

configure\_screen(): configure the screen.

title(): update the title.

configure\_snake(shape="square", color="red", x=0, y=0): configure the snake. “Shape” and “color” are the shape and color of the snake. “X” and “y” are the initial place of the snake.

configure\_monster(shape="square", color="purple"): configure the monster. “Shape” and “color” are the shape and color of the monster.

configure\_food(): configure the food.

move\_up(): change the direction of the snake.

move\_down(): change the direction of the snake.

move\_right(): change the direction of the snake.

move\_left(): change the direction of the snake.

flag\_change(): change the flag to determine whether the snake should move.

check\_move(): check whether the snake touch the boundary at previous direction and whether the snake should move now.

move(): move the snake.

configure\_key(s): match the keyboard and the function. “s” is a screen object.

chase(): move the monster.

copy(heading=0,update=500): make stamps of the snake. “heading” is the direction of the snake right now. “update” is the delay time interval.

extend(): extend the length of the snake.

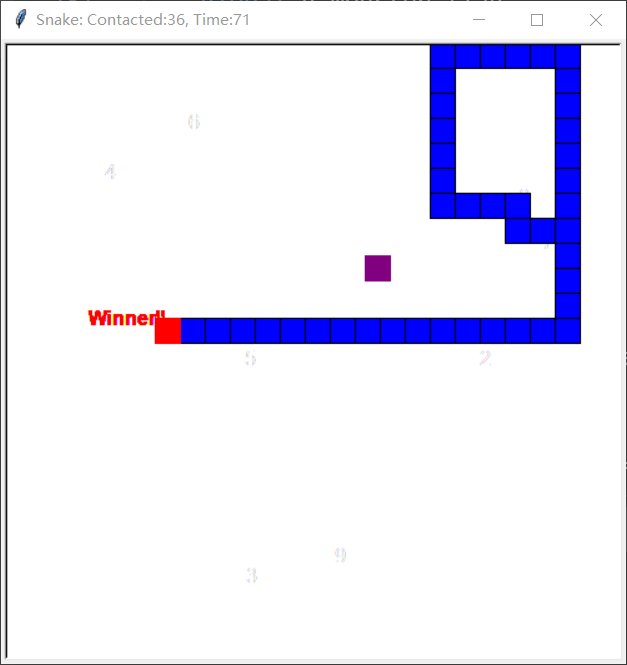
eat(): check eating.

check\_win(): check whether all the foods are consumed

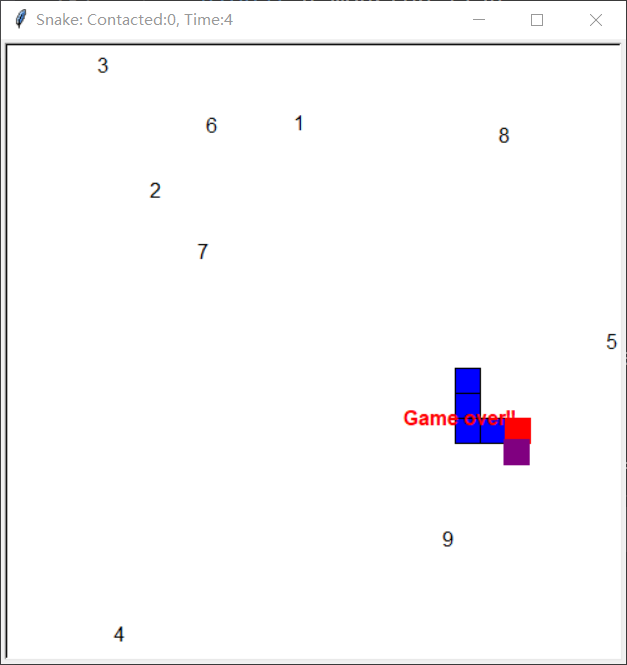
3. Output

a. Show samples of output (including status) from your program, including

i. Winner

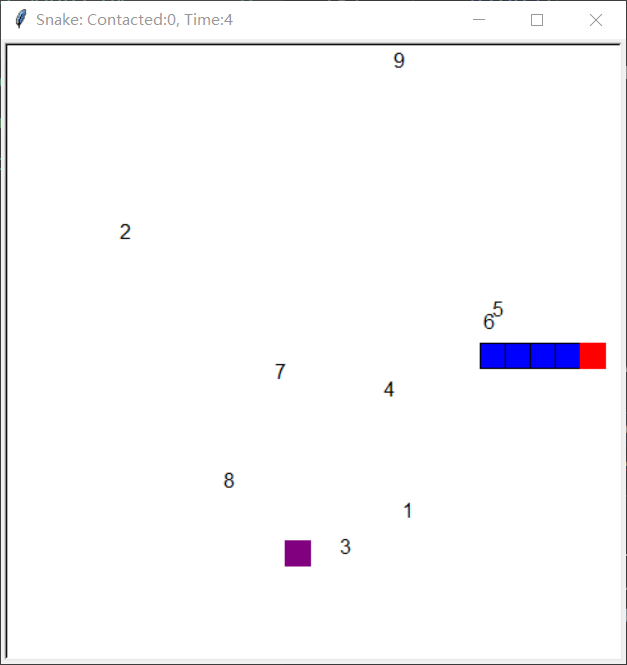


ii. Game over



iii. 2 others showing various stages of the game:

1. With 0 food item consumed



1. With 3 food items consumed

