Design Doc

OVERVIEW

• This is a Snake game. The game is composed of 3 objects: a snake, a monster and food items represented by a set of numbers from 1 to 9. The snake is represented by a sequence of squares where its head and its body are displayed in red and black colors respectively, while the monster by a purple square. The numbers are food items to be consumed by the snake.

• Users can move the snake by four directions and stop the snake by space. Margin will show the motion of the snake.

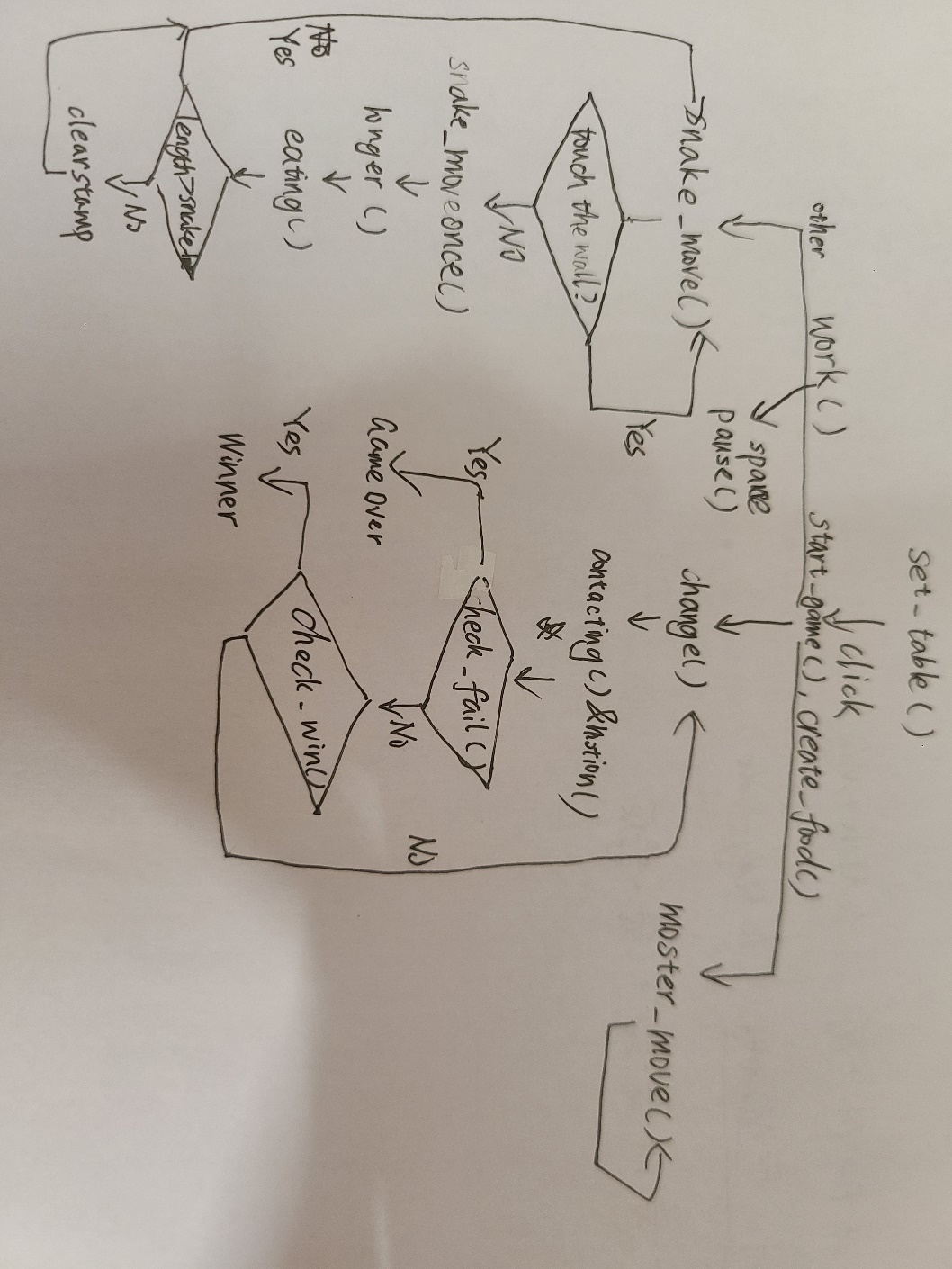
DATA MODEL

• snake: s, snake\_pos, stamp\_list, pause\_list, direction, direction\_list,

1. s: turtle() to represent the snake
2. snake\_pos: list, to record the xcor and ycor of the snake’s body
3. stamp\_list: list, to record the stamp of every part of the snake’s body
4. pause\_list: list, to record the xcor and ycor of the stop snake.
5. direction: int, to record the direction of the head of the snake
6. direction\_list: list, to record all the direction the snake has ran.

• food items: food\_list, food

1. food\_list: list, record the id of every food
2. food: list, record the food which haven’t been eaten by snake.

PROGRAM STRUCTURE

• set\_table: set the game table

• start\_game: make the welcome words disappear, do creat\_food(), record the start\_time

• work: Read the instructions and work

• snake\_move: control the snake to move

• snake\_moveonce: make the snake move one step

• pause, right, up, left, down: do the basic control

• change: change the information of contact and motion and stop the game when it is done

• monster\_move: control the monster to move and chasing the snake

PROCESSING LOGIC (SPECIFIC)

• motion the snake:

1. the snake’s head haven’t touched the wall
   1. check whether the snake eat food
   2. if the snake have to be longer, do “longer”
   3. else, clear the first stamp
2. the snake’s head touch the wall, check whether it will turn around
   1. if do, make it turn around
   2. else, make it stop

• motion the monster:

let absx become the difference of x between snake and monster and absy become the difference of y.

Priority to reduce the larger of absx and absy

• expand the tail

1. the tail has to expand: make timer become 400 and add a stamp in this direction.
2. The tail has expanded all: make the timer become 200 and add a stamp in the direction, delete the first stamp

• detect body contact between the snake and the monster:

As the stamp of each body of snake and monster are both square with edge of 20. If the difference between them of x and y both smaller than 20, they are contacting.

FUNCTIONAL SPEC

• set\_table: set the game table

• start\_game: make the welcome words disappear, do creat\_food, record the start\_time

• longer: make the snake expand with the direction and record the stamp of the new head.

• get\_pos: return a random position on the table, to create the food.

• eating: check whether the head touch any food, if it does, clear the stamp of the food and return the length it should add

• check\_in: check whether the head touch the wall, if it doesn’t, return True.

• check\_fail: check whether the monster touch the head of the snake, if it does, return True.

• check\_win: check whether the snake eating all the food, if it does, return True.

• work: Read the instructions and work

• snake\_move: control the snake to move

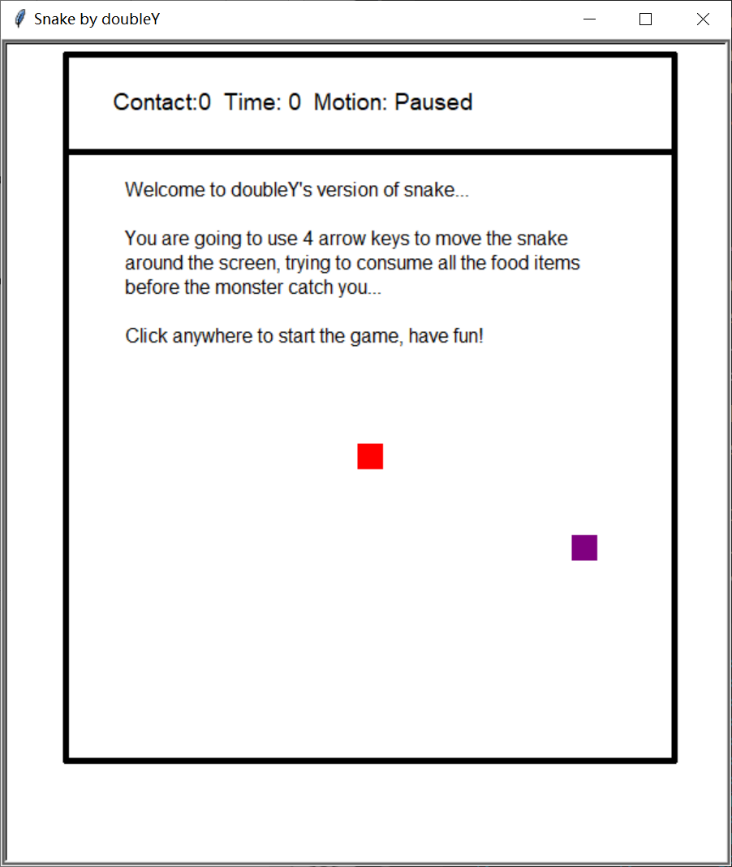
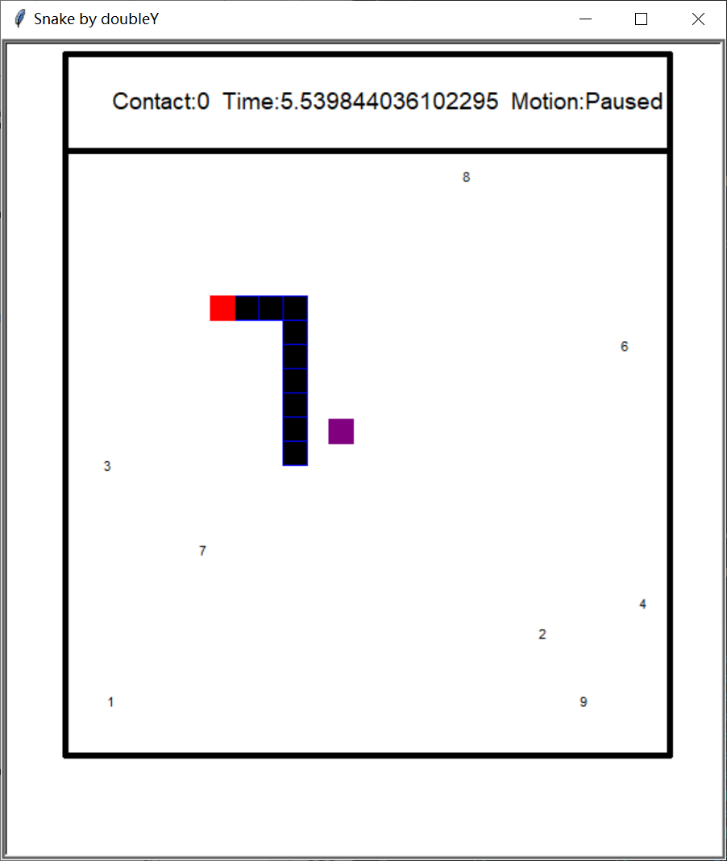
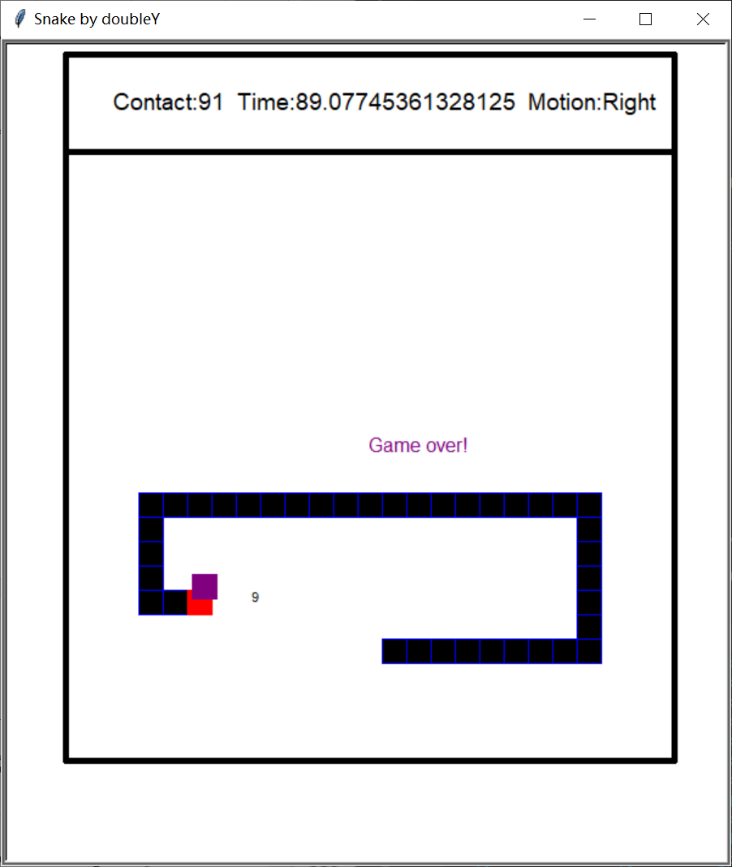
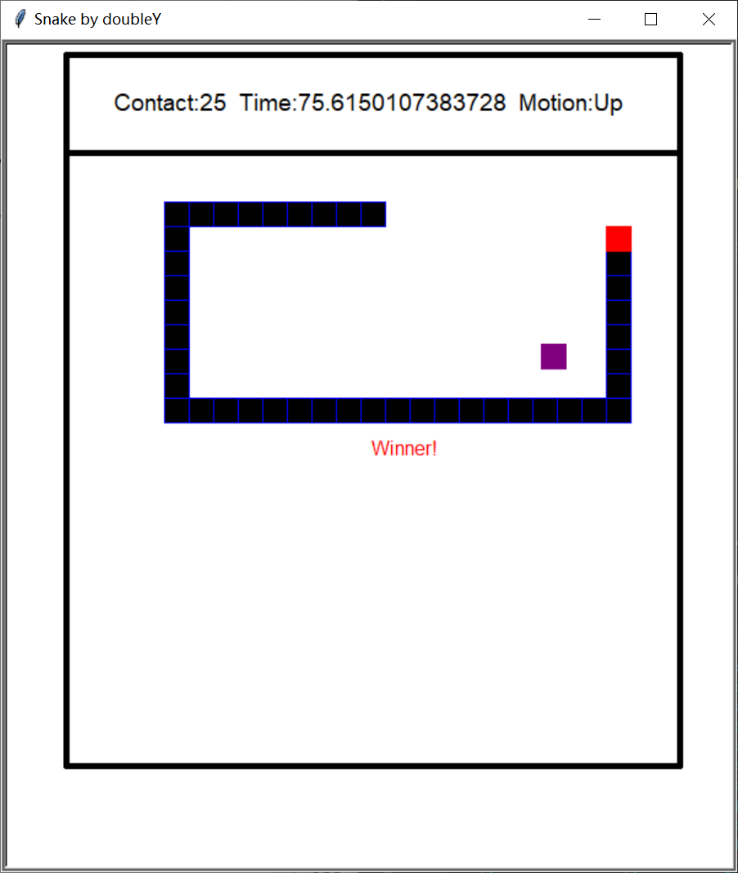
• snake\_moveonce: make the snake move one step

• pause, right, up, left, down: do the basic control

• change: change the information of contact and motion and stop the game when it is done

• monster\_move: control the monster to move and chasing the snake

• motion: return the motion of the snake as str.

SAMPLE OUTPUT