Design Doc

Design

a. Overview

The game is designed to be 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 orange colors respectively, while the monster by a purple square. The numbers are food items to be consumed by the snake.

The goal of the game is to maneuver the snake within the game area in four directions (up, down, left and right), trying to consume all the food items while avoiding head-on collision with the monster. As each food item is consumed, the snake grows with its body lengthened in size equal to the value of the number being passed. While directing the movement of the snake you should avoid contact with the monster. Furthermore, the monster is also programmed to be motioned in the direction towards the head of the snake at a variable speed.

b. Data Model

a) Snake's head: a turtle object. (head)
b) Snake's tail(body): the "stamps" of the snake's head. (Their positions are stored in a list bodypos and their stampids are saved in another list segment.)
c) Food items: context drew by a turtle object. (Their positions are saved in a dictionary foodpos.

c. Program Structure

My program has a declaration - functions structure.

I declare all the global variables and create several turtle objects for the first part of my program.

The second part are functions. My functions are mainly relevant to the three items: snake(head), monster, and food items. I use ontimer method for the runMonster() and runSnake() to keep them moving at certain pace. For each step of the monster and snake, I check if certain conditions are met to pause or end the game (e.g., paused/blocked, win or fail). For the moving of the snake, I bind four direction arrow keys to four functions goUp() /goDown() /goLeft() /goRight() to set the directions.

For the food items, I use two functions setFood() and drawfood() to update their displays if they are eaten by the snake.

I also use drawBorder(), drawTitle0(), and drawTitle() to draw the game area, display the welcome page, and show the status during the gaming process.

I use initGame() to initiate the game at the end of my program.

d. Processing Logic

a) Logic for the motion of snake and monster

Overall, I use screen.ontimer(fun,t) for the movement of both snake and monster. Specifically, I add screen.ontimer(runSnake,t) and screen.ontimer(runMonster,t) inside the functions runSnake() and runMonster() themselves. So each time the snake or monster moves, it simultaneous set a timer to call the function later, i.e., they will move continuously until certain conditions are (not) satisfied.

For the snake's direction, I also use screen.onkey() in the main loop to bind four arrow keys and space bar to four direction-setting functions and one pause() functions. As a result, the player can change to direction of the snake independently. (Not relevant to its movement)

For the monster's movement. I compare the absolute differences of x and y coordinates respectively between the monster and the snake's head and choose the greater direction to move. (i.e., If they have a greater distance in y-axis, the monster will move in the y direction)

For each movement of snake and monster, the distance is 20 pixels. The delay in ontimer() for the snake is set to a constant 200 and changed to 400 when extending its tail. The delay for the monster is a random number between (150,500), making it slower or faster than the snake randomly.

b) Logic used to expand the snake tail

Before the movement of the snake head. I change it to the model of its body (change color & border), and use turtle.stamp() to make a copy of it at current position, storing its position & stampid coordinates into two list segement and bodypos, change it back to the head model, and let it go.

I modify its body length by comparing the length of segement and the global variable bodylen. I delete the extra body segement by using turtle.clearstamp()

c) Logic used to detect body contact between the snake and the monster

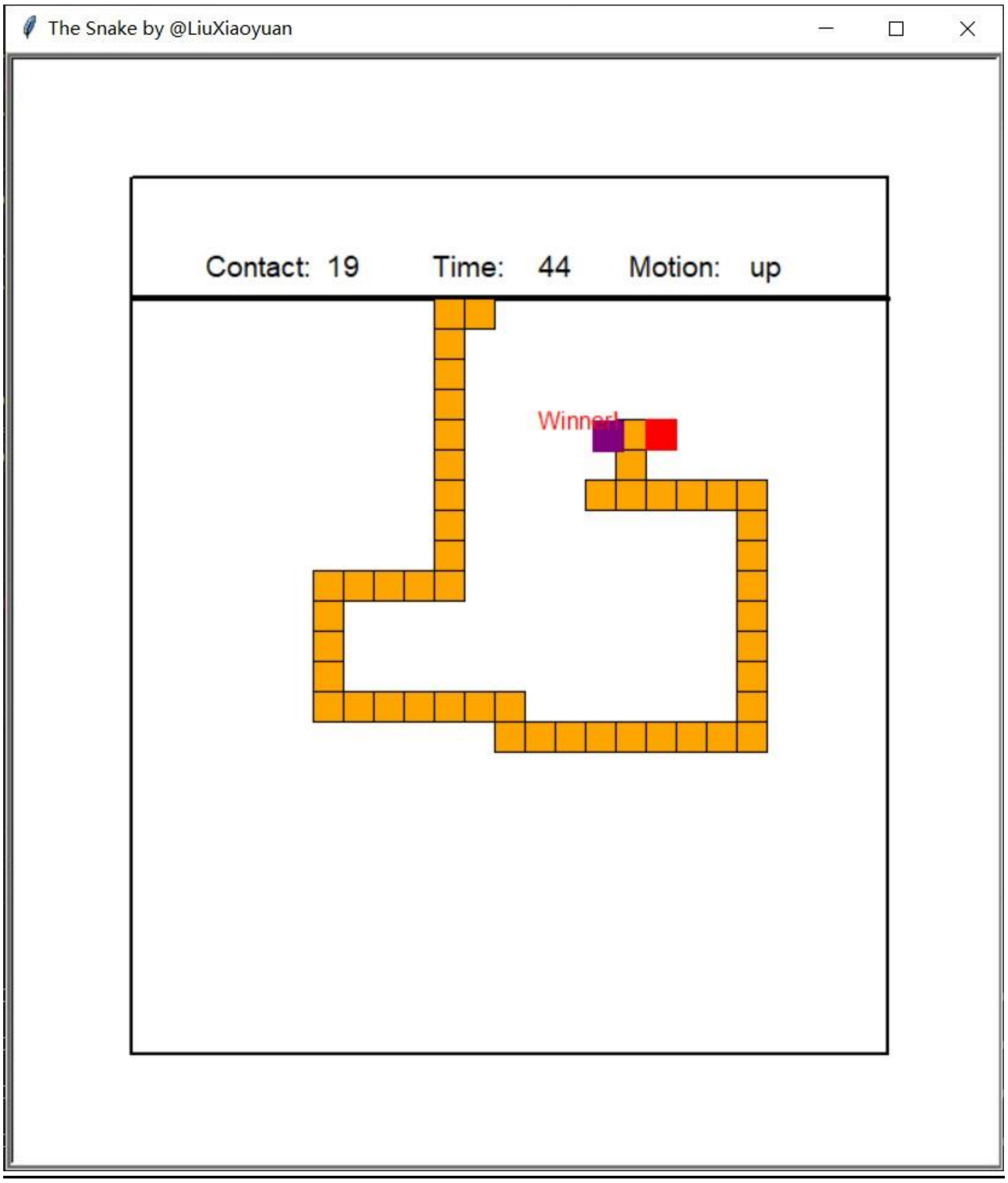
Inside the runMonster() function and before the movement of monster, I use a for loop to check the distance between the monster and each (x,y) of snake tail stored in the list bodypos. The global variable contact will plus one if one distance is <10 and break the loop.

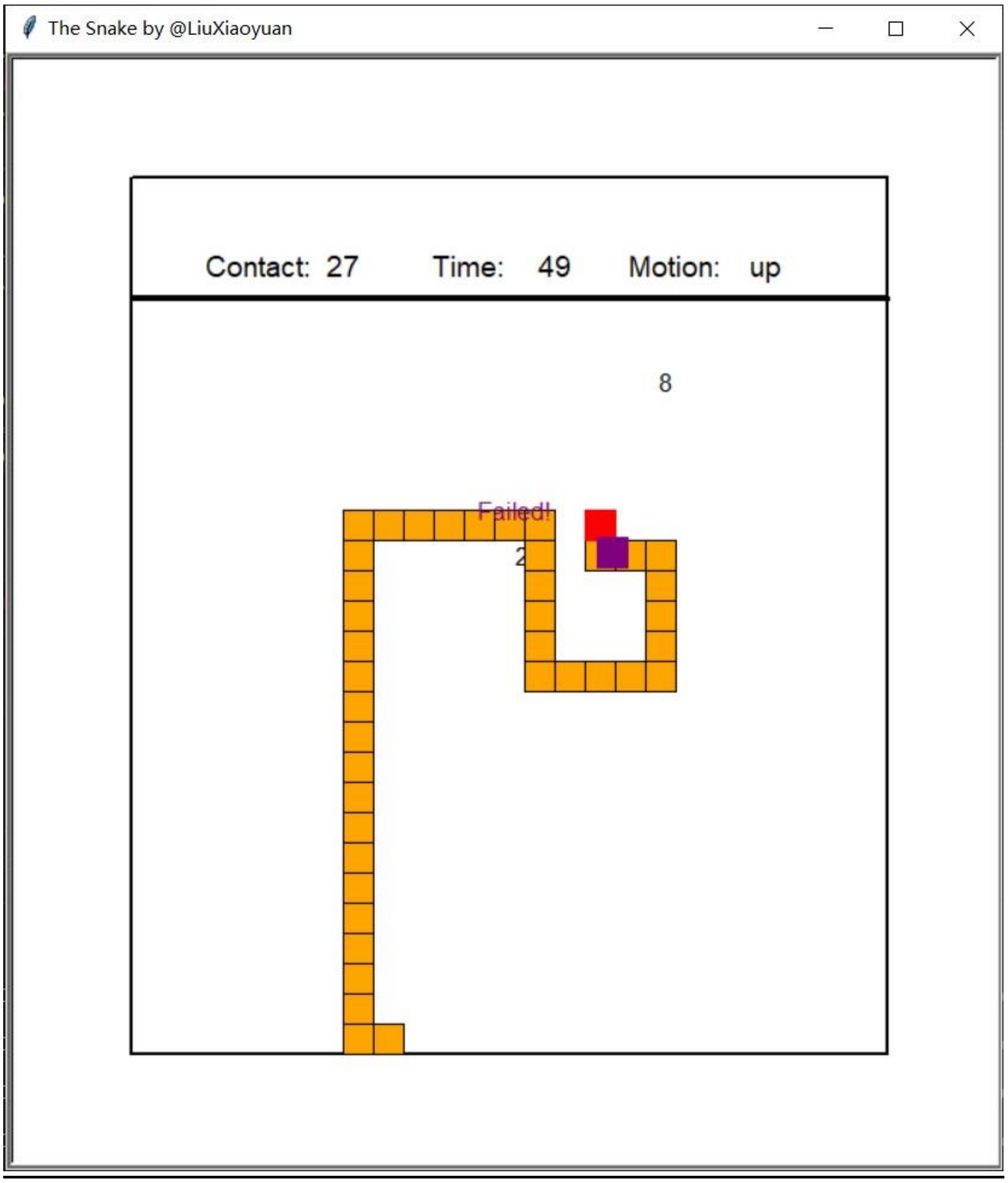
Function Specifications

1. initMonster(): change the color & shape of the monster object and place it at certain position inside the gaming area, except those occupied by snake body. The distance between the monster and snake head is no smaller than 200.
2. drawBorder(): use the turtle to draw the borders and the static caption.
3. setFood(): generate nine position coordinates in the gaming area and save them into a dictionary foodpos with key 1-9.
4. drawFood(somedic): use a turtle to write the key at its corresponding position (its value), given a dictionary.
5. goUp/Down/Left/Right(): set the snake's head into corresponding direction.
6. move(): for the snake head, forward it by 20 pixels at the current direction.
7. pause(): pause/unpause the snake's current motion. Its initial direction will not change after the pause/unpause.
8. eatFood(): check whether the snake's head is within a small distance of the food's positions. If so, change the bodylen and pop that food position in the foodpos and call drawFood() to redraw the rest food items. If the length of foodpos is 0, change the global winning to 1.
9. runMonster(): check head-on collision. Else, check body contact, move the monster at the direction with larger distance with the snake head. randomly generate a pace, and set the timer to call itself at this pace.
10. runSnake(): check winning condition. Check direction status. Check if touches border. Else, make a copy of the head and forward in current direction. Set a timer to call itself with a constant delay 200. The delay will be set to 400 when the bodylen is greater than len(segement), i.e., the snake is extending its tail.

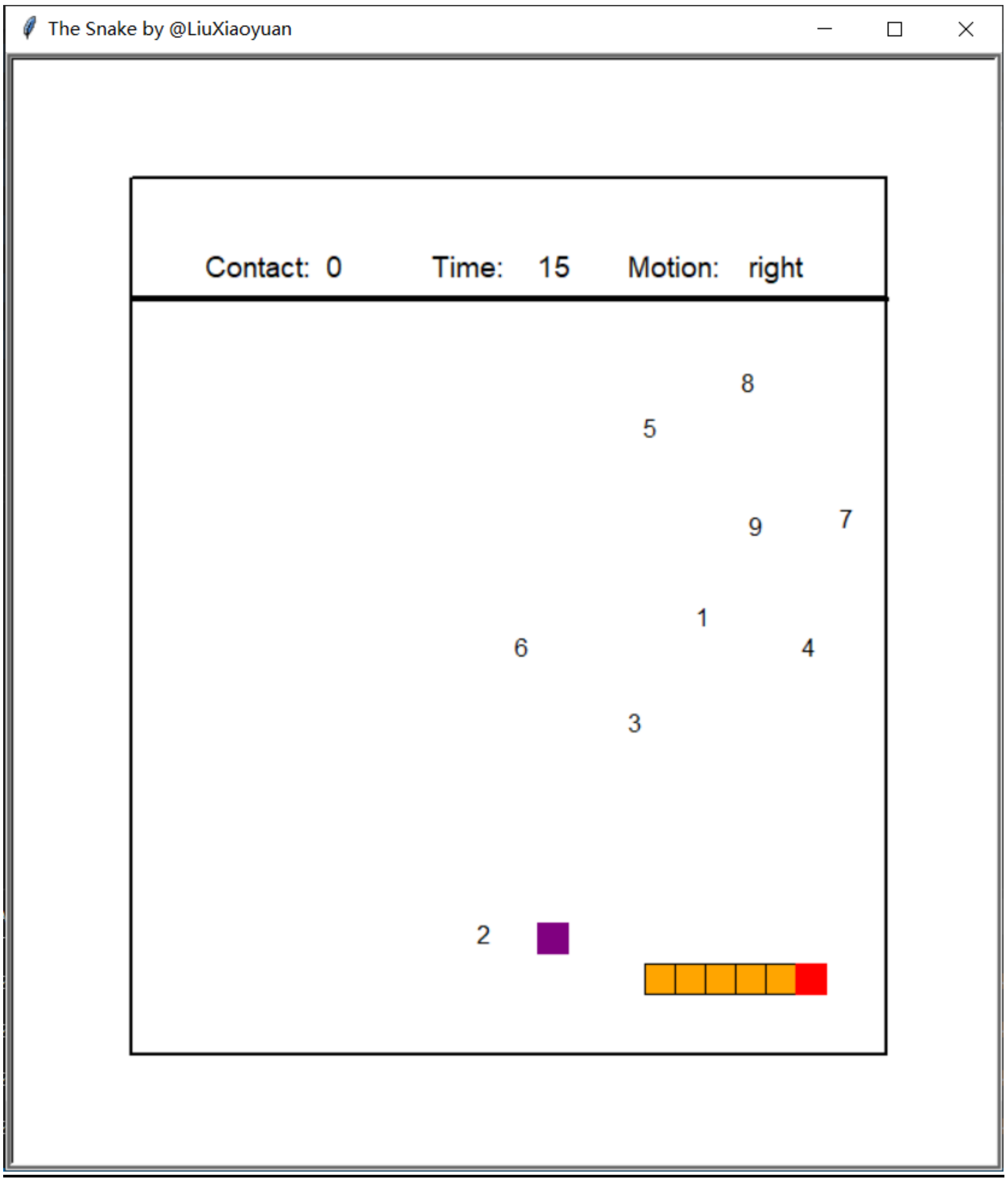
11. drawTitle(): use a turtle to draw the gaming info on the screen. (win/fail; contact, time, motion)
12. drawTitle0(): use a turtle to draw the welcome words and default game settings on the screen.
13. main(x,y): mainloop, sequence the functions above by logic. X, y are only create to fit screen.onclick(), they have no actual meanings.
14. initGame(): set the game for ready and display the welcome info. The game will start after a mouse click.

Sample Output

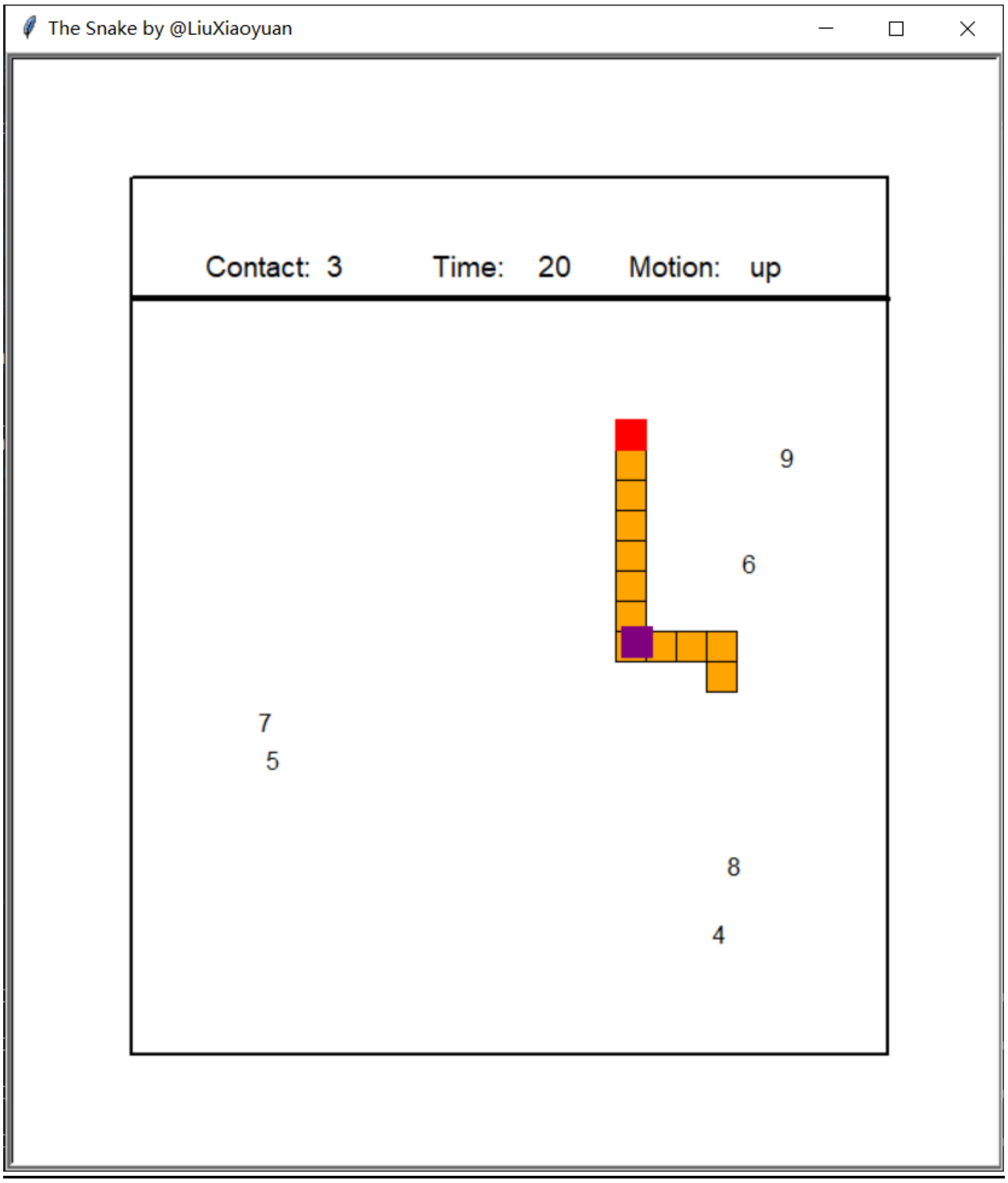



The Snake by @LiuXiaoyuan

Contact: 27 Time: 49 Motion: up

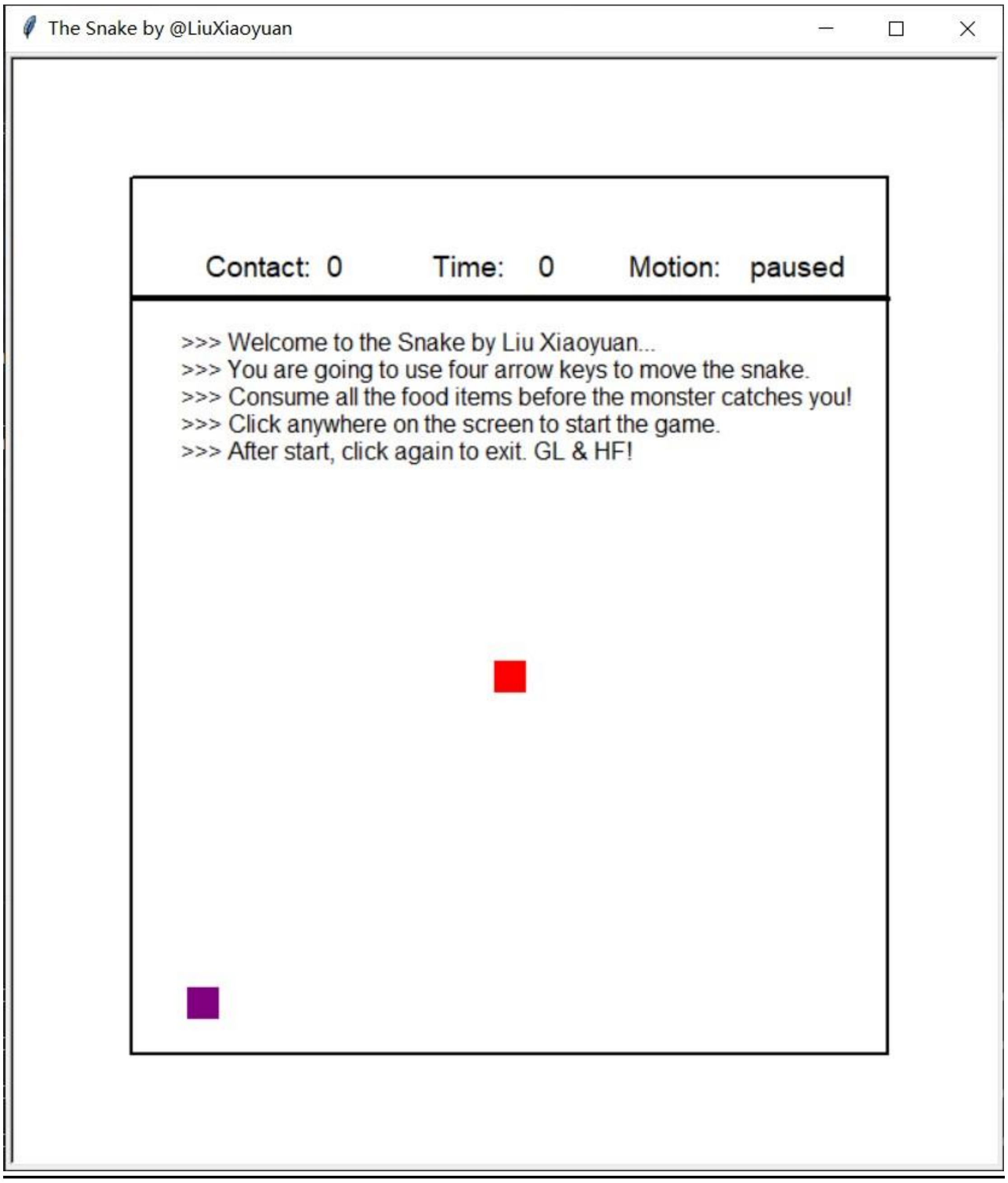

The Snake by @LiuXiaoyuan

Contact: 0 Time: 15 Motion: right



The Snake by @LiuXiaoyuan

Contact: 3 Time: 20 Motion: up



The Snake by @LiuXiaoyuan

Contact: 0 Time: 0 Motion: paused

```
>>> Welcome to the Snake by Liu Xiaoyuan...
>>> You are going to use four arrow keys to move the snake.
>>> Consume all the food items before the monster catches you!
>>> Click anywhere on the screen to start the game.
>>> After start, click again to exit. GL & HF!
```