**Snake**

**Game method：**

**A small snake eats food in the scene. The length of the snake increases by one for each meal it eats. Until the snake bumps into a wall or itself.**

**1. How do you mean snake?**

**We can divide the whole game area into small grids, and a group of interconnected grids will form a "snake". We can represent each of the little squares in terms of coordinates, and we can set the range of the X and Y axes. Use a list to store the coordinates of the snake body, and a snake emerges. Finally, we just need to display it in a different color.**

**2. How do snakes move?**

**Each time the snake moves, it feels like it moves one space forward as a whole. In fact, apart from the head and tail, nothing has changed at all. Adding the coordinates of the next grid to the beginning of the list and removing the last element of the list is equivalent to the snake moving forward one grid.**

**3. How do you know the game is over?**

**The "snake" moves out of the game zone or runs into itself and loses. The range of axis coordinates is set in advance, and it is easy to judge if it is out of range. Just determine if the coordinates of the next grid are already included in the list of snakes or beyond the scene range.**

**Design steps:**

1. **Create Windows**

**Create Windows with the tkinter package**

1. **Basic parameter setting**

**The snake's body and walls are represented by a square grid, so set the length of the sides of the square, as well as the height and width of the form.**

1. **Initialization of walls and snakes**
2. **Random cookie(food) generation**
3. **The movement and turning of snakes**

**In fact, this is to update the snake coordinates in the list, and then update the image in the window. The snake's movement can be interpreted as removing the coordinates of the snake's tail and adding the coordinates of the previous snake's head plus the direction of movement.**

1. **Operation related to snake eating cookie**
2. **Score Function and Game Over**