Straightforward Python solution using deque

ninjaVic ★ 456 Last Edit: October 27, 2018 5:34 AM 3.1K VIEWS

39

class SnakeGame(object):

```
def __init__(self, width,height,food):
    """
    Initialize your data structure here.
    @param width - screen width
    @param height - screen height
    @param food - A list of food positions
    E.g food = [[1,1], [1,0]] means the first food is positioned at [1,1], the second is at [1,0].
    :type width: int
    :type height: int
    :type food: List[List[int]]
    """
    self.snake = collections.deque([[0,0]])    # snake head is at the front
    self.width = width
    self.height = height
    self.food = collections.deque(food)
    self.direct = {'U': [-1, 0], 'L': [0, -1], 'R': [0, 1], 'D': [1, 0]}


def move(self, direction):
    """
    Moves the snake.
    @param direction - 'U' = Up, 'L' = Left, 'R' = Right, 'D' = Down
    @return The game's score after the move. Return -1 if game over.
    Game over when snake crosses the screen boundary or bites its body.
    :type direction: str
    :rtype: int
    """
    newHead = [self.snake[0][0]+self.direct[direction][0], self.snake[0][1]+self.direct[direction][1]]

    # notice that the newHead can be equal to self.snake[-1]
    if (newHead[0] < 0 or newHead[0] >= self.height) or (newHead[1] < 0 or newHead[1] >= self.width)\
    or (newHead in self.snake and newHead != self.snake[-1]): return -1

    if self.food and self.food[0] == newHead:  # eat food
        self.snake.appendleft(newHead)   # just make the food be part of snake
        self.food.popleft()   # delete the food that's already eaten
    else:    # not eating food: append head and delete tail
        self.snake.appendleft(newHead)
        self.snake.pop()

    return len(self.snake)-1
```