

#13: Walkthrough - Snake

SAMS SENIOR CS TRACK

A solid orange horizontal bar at the bottom of the slide.

This Week's Learning Goals

Use a **time loop** to create time-based animations.

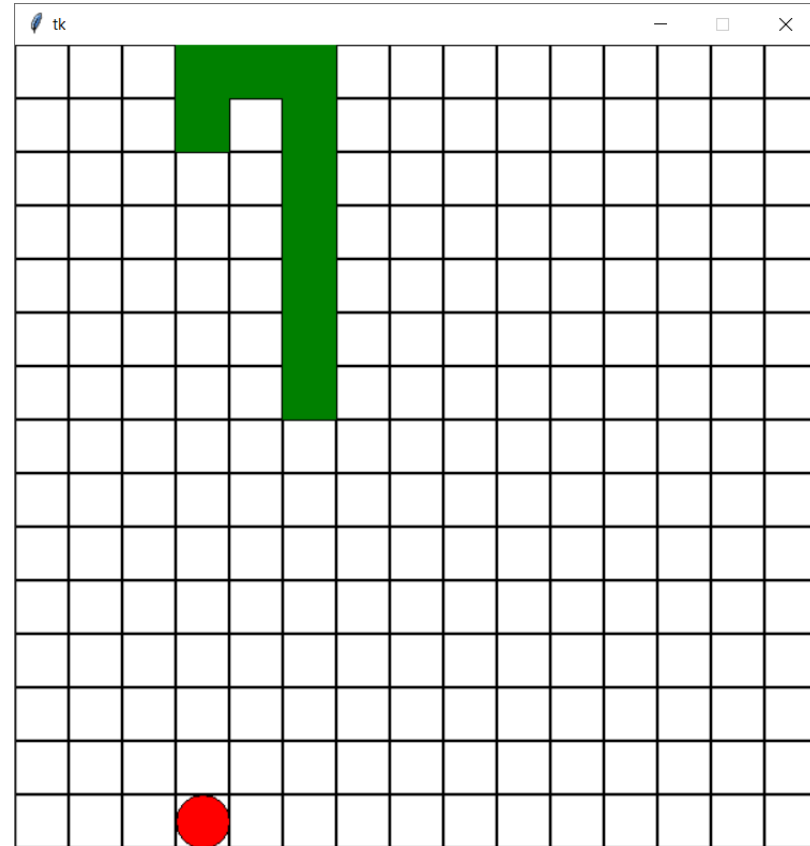
Use **randomness** to simulate events.

Today's Goal

Program the classic arcade game Snake!

Example:

<https://www.google.com/search?q=snake+game>

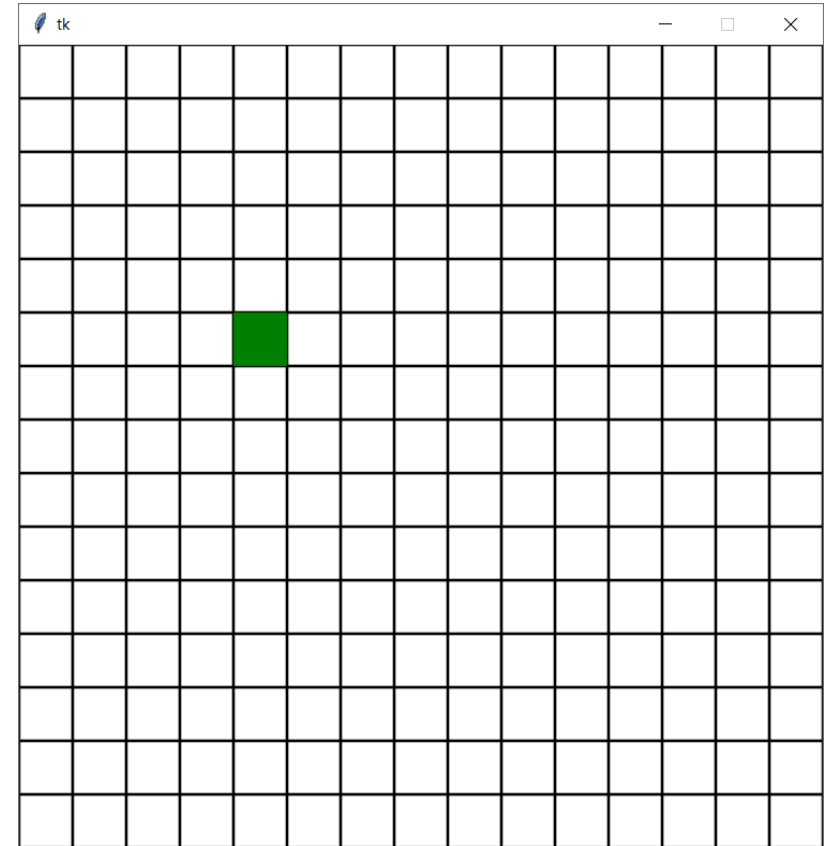


Step 1: Grid + Snake

```
def init(data):
    data.size = 15
    data.cellSize = data.width // data.size
    data.snake = [ [ random.randint(0, data.size-1),
                    random.randint(0, data.size-1) ] ]

def redrawAll(canvas, data):
    for row in range(data.size):
        for col in range(data.size):
            canvas.create_rectangle(col * data.cellSize,
                                   row * data.cellSize,
                                   (col + 1) * data.cellSize,
                                   (row + 1) * data.cellSize, width=2)

    for part in data.snake:
        [row, col] = part
        canvas.create_rectangle(col * data.cellSize,
                                row * data.cellSize,
                                (col + 1) * data.cellSize,
                                (row + 1) * data.cellSize,
                                fill="green", width=0)
```

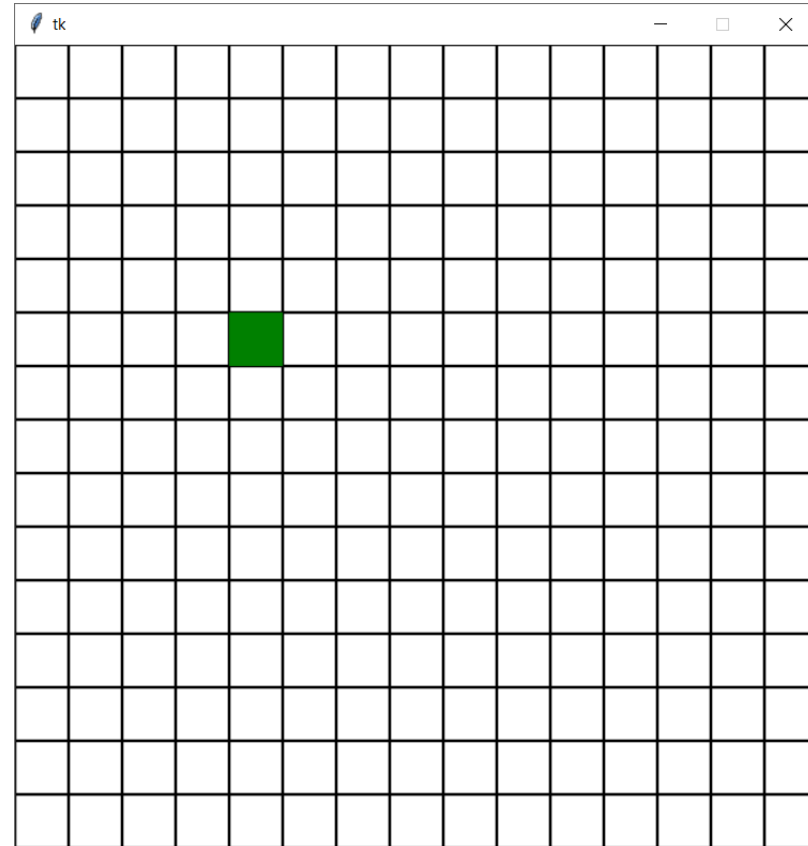


Step 2: Snake Movement

```
def init(data):
    ...
    data.timerDelay = 500
    data.dir = random.choice([ [-1, 0], [1, 0],
                               [0, -1], [0, 1] ])

def keyPressed(event, data):
    if event.keysym == "Up":
        data.dir = [-1, 0]
    elif event.keysym == "Down":
        data.dir = [1, 0]
    elif event.keysym == "Left":
        data.dir = [0, -1]
    elif event.keysym == "Right":
        data.dir = [0, 1]

def timerFired(data):
    for i in range(len(data.snake)):
        data.snake[i][0] += data.dir[0]
        data.snake[i][1] += data.dir[1]
```

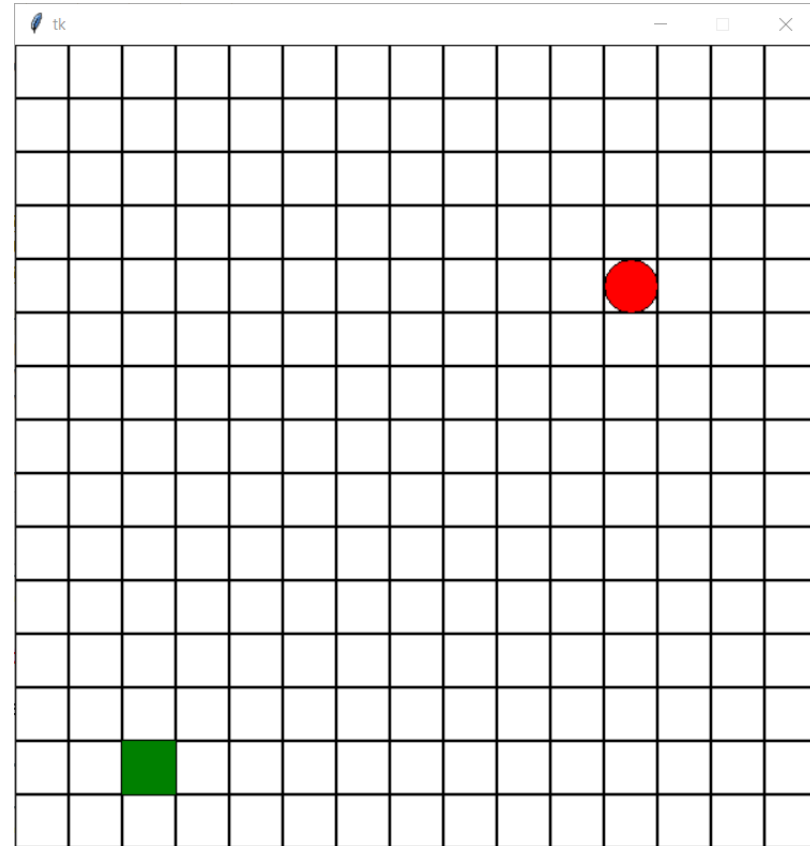


Step 3: Adding Food

```
def init(data):
    ...
    makeFood(data)

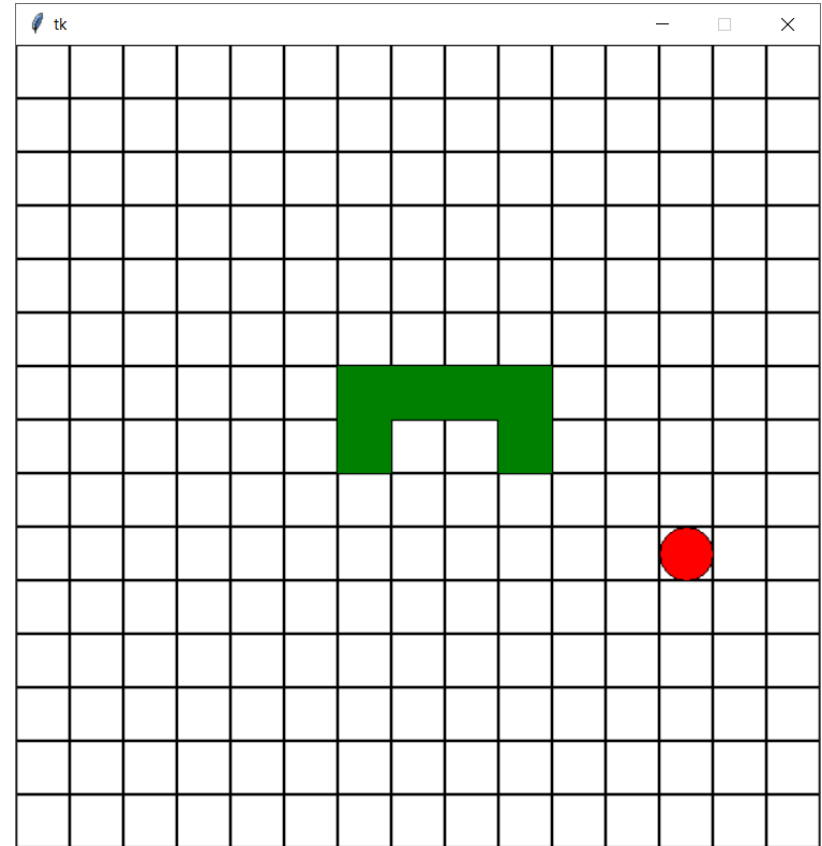
def makeFood(data):
    row, col = random.randint(0, data.size-1),
               random.randint(0, data.size-1)
    while [row, col] in data.snake:
        row, col = random.randint(0, data.size-1),
                   random.randint(0, data.size-1)
    data.food = [row, col]

def redrawAll(canvas, data):
    ...
    canvas.create_oval(data.food[1] * data.cellSize,
                       data.food[0] * data.cellSize,
                       (data.food[1] + 1) * data.cellSize,
                       (data.food[0] + 1) * data.cellSize,
                       fill="red")
```



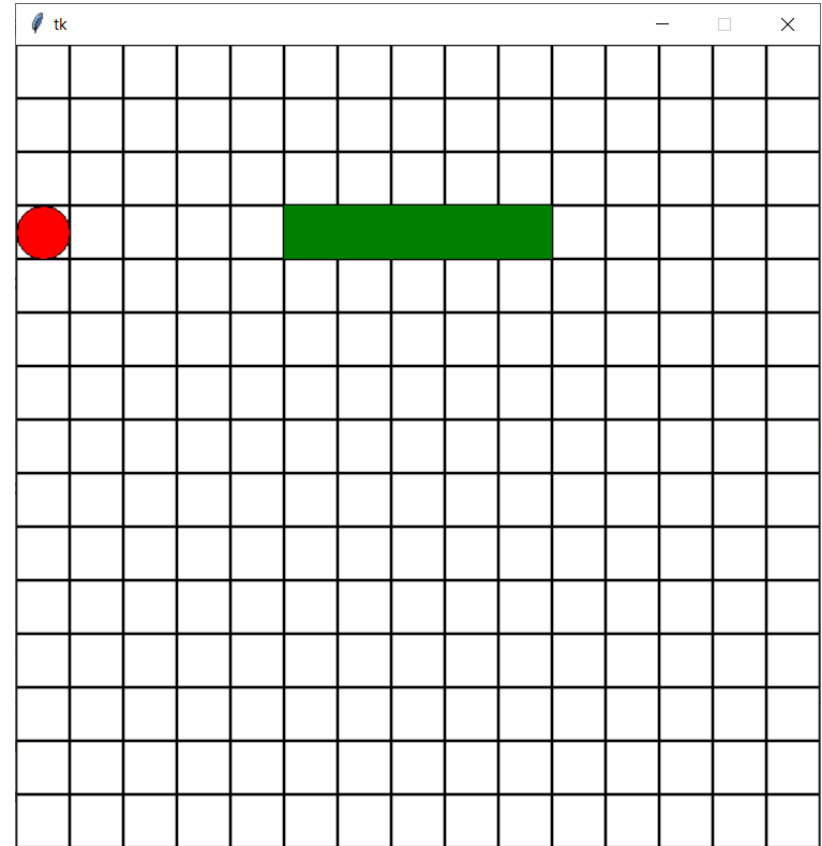
Step 4: Eating Food + Growing - Bad

```
def timerFired(data):  
    lastPart = data.snake[-1] + []  
  
    for i in range(len(data.snake)):  
        data.snake[i][0] += data.dir[0]  
        data.snake[i][1] += data.dir[1]  
  
    if data.snake[0] == data.food:  
        data.snake.append(lastPart)  
        makeFood(data)
```



Step 4: Eating Food + Growing - Good

```
def timerFired(data):
    newHead = [ data.snake[0][0] + \
                data.dir[0],
                data.snake[0][1] + \
                data.dir[1] ]
    data.snake.insert(0, newHead)
    if newHead == data.food:
        makeFood(data)
    else:
        data.snake.pop()
```



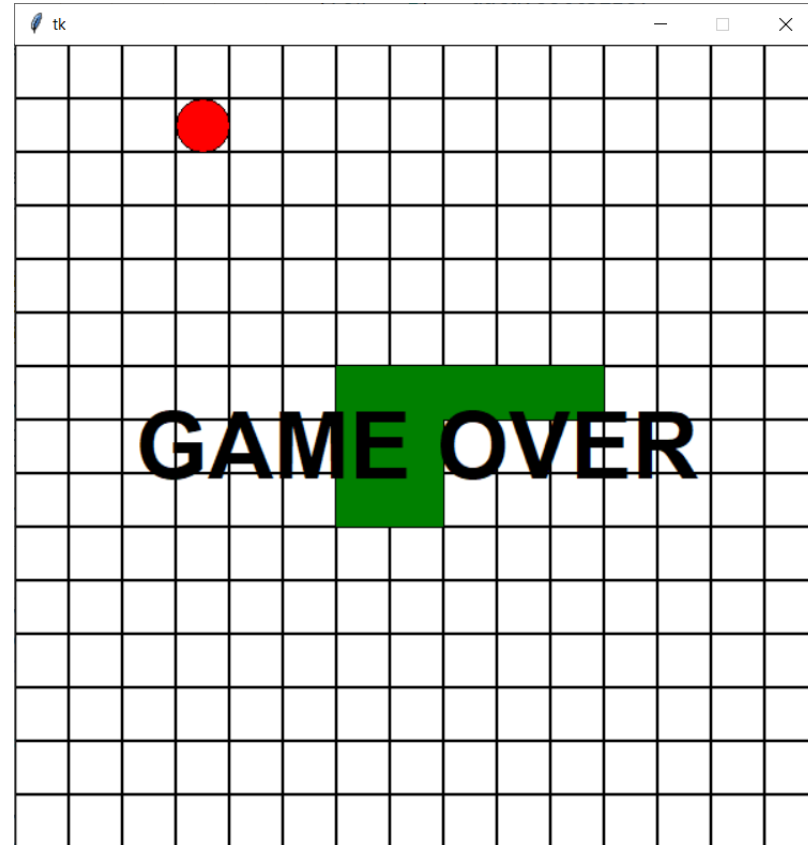
Step 5: Detect Game Over

```
def init(data):
    ...
    data.gameOver = False

def keyPressed(event, data):
    if data.gameOver: return
    ...

def timerFired(data):
    if data.gameOver: return
    ...
    if newHead[0] < 0 or newHead[0] >= data.size or \
        newHead[1] < 0 or newHead[1] >= data.size:
        data.gameOver = True
    elif newHead in data.snake[1:]:
        data.gameOver = True

def redrawAll(canvas, data):
    ...
    if data.gameOver:
        canvas.create_text(data.width/2, data.height/2,
                           text="GAME OVER", font="Arial 52 bold")
```



Homework Q&A Time
