**import** turtle

**import** random

w **=** 500

h **=** 500

food\_size **=** 10

delay **=** 100

offsets **=** {

    "up": (0, 20),

    "down": (0, **-**20),

    "left": (**-**20, 0),

    "right": (20, 0)

}

**def** reset():

**global** snake, snake\_dir, food\_position, pen

    snake **=** [[0, 0], [0, 20], [0, 40], [0, 60], [0, 80]]

    snake\_dir **=** "up"

    food\_position **=** get\_random\_food\_position()

    food.goto(food\_position)

    move\_snake()

**def** move\_snake():

**global** snake\_dir

    new\_head **=** snake[**-**1].copy()

    new\_head[0] **=** snake[**-**1][0] **+** offsets[snake\_dir][0]

    new\_head[1] **=** snake[**-**1][1] **+** offsets[snake\_dir][1]

**if** new\_head **in** snake[:**-**1]:

        reset()

**else**:

        snake.append(new\_head)

**if** **not** food\_collision():

            snake.pop(0)

**if** snake[**-**1][0] > w **/** 2:

            snake[**-**1][0] **-=** w

**elif** snake[**-**1][0] < **-** w **/** 2:

            snake[**-**1][0] **+=** w

**elif** snake[**-**1][1] > h **/** 2:

            snake[**-**1][1] **-=** h

**elif** snake[**-**1][1] < **-**h **/** 2:

            snake[**-**1][1] **+=** h

        pen.clearstamps()

**for** segment **in** snake:

            pen.goto(segment[0], segment[1])

            pen.stamp()

        screen.update()

        turtle.ontimer(move\_snake, delay)

**def** food\_collision():

**global** food\_position

**if** get\_distance(snake[**-**1], food\_position) < 20:

        food\_position **=** get\_random\_food\_position()

        food.goto(food\_position)

**return** True

**return** False

**def** get\_random\_food\_position():

    x **=** random.randint(**-** w **/** 2 **+** food\_size, w **/** 2 **-** food\_size)

    y **=** random.randint(**-** h **/** 2 **+** food\_size, h **/** 2 **-** food\_size)

**return** (x, y)

**def** get\_distance(pos1, pos2):

    x1, y1 **=** pos1

    x2, y2 **=** pos2

    distance **=** ((y2 **-** y1) **\*\*** 2 **+** (x2 **-** x1) **\*\*** 2) **\*\*** 0.5

**return** distance

**def** go\_up():

**global** snake\_dir

**if** snake\_dir !**=** "down":

        snake\_dir **=** "up"

**def** go\_right():

**global** snake\_dir

**if** snake\_dir !**=** "left":

        snake\_dir **=** "right"

**def** go\_down():

**global** snake\_dir

**if** snake\_dir!**=** "up":

        snake\_dir **=** "down"

**def** go\_left():

**global** snake\_dir

**if** snake\_dir !**=** "right":

        snake\_dir **=** "left"

screen **=** turtle.Screen()

screen.setup(w, h)

screen.title("Snake")

screen.bgcolor("blue")

screen.setup(500, 500)

screen.tracer(0)

pen **=** turtle.Turtle("square")

pen.penup()

food **=** turtle.Turtle()

food.shape("square")

food.color("yellow")

food.shapesize(food\_size **/** 20)

food.penup()

screen.listen()

screen.onkey(go\_up, "Up")

screen.onkey(go\_right, "Right")

screen.onkey(go\_down, "Down")

screen.onkey(go\_left, "Left")

reset()

turtle.done()