

E:\snake.py

```
1  # import required modules
2  import turtle
3  import time
4  import random
5
6  delay = 0.1
7  score = 0
8  high_score = 0
9
10
11 # Creating a window screen
12 wn = turtle.Screen()
13 wn.title("Snake Game")
14 wn.bgcolor("blue")
15 # the width and height can be put as user's choice
16 wn.setup(width=600, height=600)
17 wn.tracer(0)
18
19 # head of the snake
20 head = turtle.Turtle()
21 head.shape("square")
22 head.color("white")
23 head.penup()
24 head.goto(0, 0)
25 head.direction = "Stop"
26
27 # food in the game
28 food = turtle.Turtle()
29 colors = random.choice(['red', 'green', 'black'])
30 shapes = random.choice(['square', 'triangle', 'circle'])
31 food.speed(0)
32 food.shape(shapes)
33 food.color(colors)
34 food.penup()
35 food.goto(0, 100)
36
37 pen = turtle.Turtle()
38 pen.speed(0)
39 pen.shape("square")
40 pen.color("white")
41 pen.penup()
42 pen.hideturtle()
43 pen.goto(0, 250)
44 pen.write("Score : 0 High Score : 0", align="center",
45         font=("candara", 24, "bold"))
46
47
48 # assigning key directions
```

```
49 def group():
50     if head.direction != "down":
51         head.direction = "up"
52
53
54 def godown():
55     if head.direction != "up":
56         head.direction = "down"
57
58
59 def goleft():
60     if head.direction != "right":
61         head.direction = "left"
62
63
64 def goright():
65     if head.direction != "left":
66         head.direction = "right"
67
68
69 def move():
70     if head.direction == "up":
71         y = head.ycor()
72         head.sety(y+20)
73     if head.direction == "down":
74         y = head.ycor()
75         head.sety(y-20)
76     if head.direction == "left":
77         x = head.xcor()
78         head.setx(x-20)
79     if head.direction == "right":
80         x = head.xcor()
81         head.setx(x+20)
82
83
84 wn.listen()
85 wn.onkeypress(group, "w")
86 wn.onkeypress(godown, "s")
87 wn.onkeypress(goleft, "a")
88 wn.onkeypress(goright, "d")
89
90 segments = []
91
92
93 # Main Gameplay
94 while True:
95     wn.update()
96     if head.xcor() > 290 or head.xcor() < -290 or head.ycor() > 290 or head.ycor() < -290:
97         time.sleep(1)
98         head.goto(0, 0)
```

```
99     head.direction = "Stop"
100     colors = random.choice(['red', 'blue', 'green'])
101     shapes = random.choice(['square', 'circle'])
102     for segment in segments:
103         segment.goto(1000, 1000)
104     segments.clear()
105     score = 0
106     delay = 0.1
107     pen.clear()
108     pen.write("Score : {} High Score : {}".format(
109         score, high_score), align="center", font=("candara", 24, "bold"))
110 if head.distance(food) < 20:
111     x = random.randint(-270, 270)
112     y = random.randint(-270, 270)
113     food.goto(x, y)
114
115     # Adding segment
116     new_segment = turtle.Turtle()
117     new_segment.speed(0)
118     new_segment.shape("square")
119     new_segment.color("orange") # tail colour
120     new_segment.penup()
121     segments.append(new_segment)
122     delay -= 0.001
123     score += 10
124     if score > high_score:
125         high_score = score
126     pen.clear()
127     pen.write("Score : {} High Score : {}".format(
128         score, high_score), align="center", font=("candara", 24, "bold"))
129 # Checking for head collisions with body segments
130 for index in range(len(segments)-1, 0, -1):
131     x = segments[index-1].xcor()
132     y = segments[index-1].ycor()
133     segments[index].goto(x, y)
134 if len(segments) > 0:
135     x = head.xcor()
136     y = head.ycor()
137     segments[0].goto(x, y)
138 move()
139 for segment in segments:
140     if segment.distance(head) < 20:
141         time.sleep(1)
142         head.goto(0, 0)
143         head.direction = "stop"
144         colors = random.choice(['red', 'blue', 'green'])
145         shapes = random.choice(['square', 'circle'])
146         for segment in segments:
147             segment.goto(1000, 1000)
148         segments.clear()
```

```
149
150         score = 0
151         delay = 0.1
152         pen.clear()
153         pen.write("Score : {} High Score : {}".format(
154             score, high_score), align="center", font=("candara", 24, "bold"))
155         time.sleep(delay)
156
157 wn.mainloop()
158
```