

In []:

```
#imports
import turtle
import time
import random

delay = 0.1

#scores
score = 0
high_score = 0

#set up screen
wn = turtle.Screen()
wn.title("Snake Game")
wn.bgcolor('black')
wn.setup(width=1000, height=1000)
wn.tracer(0)

#snake head
head = turtle.Turtle()
head.speed(0)
head.shape("circle")
head.color("white")
head.penup()
head.goto(0,0)
head.direction = "stop"

#snake food
food= turtle.Turtle()
food.speed(0)
food.shape("circle")
food.color("yellow")
food.penup()
food.goto(0,100)

segments = []

#scoreboards
sc = turtle.Turtle()
sc.speed(0)
sc.shape("circle")
sc.color("white")
sc.penup()
sc.hideturtle()
sc.goto(0,260)
sc.write("Score: 0  High score: 0", align = "center", font=("ds-digital", 24, "normal"))

#Functions
def go_up():
    if head.direction != "down":
        head.direction = "up"
def go_down():
    if head.direction != "up":
        head.direction = "down"
def go_left():
    if head.direction != "right":
        head.direction = "left"
def go_right():
    if head.direction != "left":
        head.direction = "right"
def move():
    if head.direction == "up":
        y = head.ycor()
        head.sety(y+20)
    if head.direction == "down":
        y = head.ycor()
        head.sety(y-20)
    if head.direction == "left":
        x = head.xcor()
        head.setx(x-20)
    if head.direction == "right":
        x = head.xcor()
        head.setx(x+20)

#keyboard bindings
wn.listen()
wn.onkey(go_up, "w")
wn.onkey(go_down, "s")
wn.onkey(go_left, "a")
wn.onkey(go_right, "d")

#MainLoop
while True:
    wn.update()

    #check collision with border area
    if head.xcor()>290 or head.xcor()<-290 or head.ycor()>290 or head.ycor()<-290:
        time.sleep(1)
        head.goto(0,0)
        head.direction = "stop"

        #hide the segments of body
        for segment in segments:
            segment.goto(1000,1000) #out of range
        #clear the segments
        segments.clear()

        #reset score
        score = 0

        #reset delay
        delay = 0.1

        sc.clear()
        sc.write("Score: {}  High score: {}".format(score, high_score), align="center", font=("ds-digital", 24, "normal"))

    #check collision with food
    if head.distance(food) <20:
        # move the food to random place
        x = random.randint(-290,290)
        y = random.randint(-290,290)
        food.goto(x,y)

        #add a new segment to the head
        new_segment = turtle.Turtle()
        new_segment.speed(0)
        new_segment.shape("circle")
        new_segment.color("white")
        new_segment.penup()
        segments.append(new_segment)

        #shorten the delay
        delay -= 0.001
        #increase the score
        score += 10

        if score > high_score:
            high_score = score
        sc.clear()
        sc.write("Score: {}  High score: {}".format(score,high_score), align="center", font=("ds-digital", 24, "normal"))

    #move the segments in reverse order
    for index in range(len(segments)-1,0,-1):
        x = segments[index-1].xcor()
        y = segments[index-1].ycor()
        segments[index].goto(x,y)
    #move segment 0 to head
    if len(segments)>0:
        x = head.xcor()
        y = head.ycor()
        segments[0].goto(x,y)

    move()

    #check for collision with body
    for segment in segments:
        if segment.distance(head)<20:
            time.sleep(1)
            head.goto(0,0)
            head.direction = "stop"

            #hide segments
            for segment in segments:
                segment.goto(1000,1000)
            segments.clear()
            score = 0
            delay = 0.1

            #update the score
            sc.clear()
            sc.write("Score: {}  High score: {}".format(score,high_score), align="center", font=("ds-digital", 24, "normal"))
            time.sleep(delay)
wn.mainloop()
```

In []: