Name - Sudhir Patidar

Project Name -Snake Game

```
import turtle
import time
import random

# Set up the screen
window = turtle.Screen()
window.title("Snake Game")
window.bgcolor("black")
window.setup(width=600, height=600)
window.tracer(0)  # Turn off automatic screen updates

# Snake head
head = turtle.Turtle()
head.speed(0)
head.shape("square")
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("red")
food.penup()
food.goto(0, 100)

segments = []

# 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
window.listen()
window.onkeypress(go_up, "w")
window.onkeypress(go_down, "s")
window.onkeypress(go_left, "a")
```

```
window.onkeypress(go_left, "a")
window.onkeypress(go_right, "d")

# Main game loop
while True:
    window.update()

    # Check for collision with food
    if head.distance(food) < 20:
        # Move the food to a random location
        x = random.randint(-290, 290)
        y = random.randint(-290, 290)
        food.goto(x, y)

        # Add a new segment to the snake
        new_segment = turtle.Turtle()
        new_segment.speed(0)
        new_segment.shape("square")
        new_segment.color("grey")
        new_segment.penup()
        segments.append(new_segment)

    # Move the end segments first in reverse order
    for i in range(len(segments) - 1, 0, -1):
        x = segments[i - 1].xcor()
        y = segments[i - 1].ycor()
        segments[i].goto(x, y)

    # Move segment 0 to where the head is
    if len(segments) > 0:
        x = head.xcor()
        y = head.ycor()
        segments[0].goto(x, y)

    move()

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

            # Hide the segments
            for seg in segments:
                seg.goto(1000, 1000)

            # Clear the segments list
            segments.clear()

    time.sleep(0.1)
```

```
---------------------------------------------------------------------------
TclError                                  Traceback (most recent call last)
<ipython-input-4-c9be25f1928f> in <cell line: 6>()
      4
      5 # Set up the screen
----> 6 window = turtle.Screen()
      7 window.title("Snake Game")
      8 window.bgcolor("black")

                                3 frames
/usr/lib/python3.10/tkinter/__init__.py in __init__(self, screenName, baseName,
className, useTk, sync, use)
   2297                 baseName = baseName + ext
   2298         interactive = False
-> 2299         self.tk = _tkinter.create(screenName, baseName, className, interactive,
wantobjects, useTk, sync, use)
   2300         if useTk:
   2301             self._loadtk()

TclError: no display name and no $DISPLAY environment variable
```

SEARCH STACK OVERFLOW