Smart Home IoT Simulator Documentation

Task: Python-based IoT simulator for a smart home automation system. In the system, following will be included: Thermostat, Smart lights and Security Camera. These devices can be controlled through a graphical user interface (GUI) known as the dashboard.

'`SmartHomeSimulator`'

Description: This class represents the main simulation program. It creates a Tkinter GUI with a dashboard containing controls for the thermostat, smart lights, and security camera.

'`__init__ (self, master)`' : It is a constructor method that initialises the SmartHomeSimulator object.

'`create_widgets(self)`' : used for creating and organising the GUI widgets of the devices and activity log.

'`log_activity(self, message)`' : logs an activity message to the activity log text widget.

'`adjust_thermostat(self)`' : adjusts the thermostat based on the user input for the desired temperature.
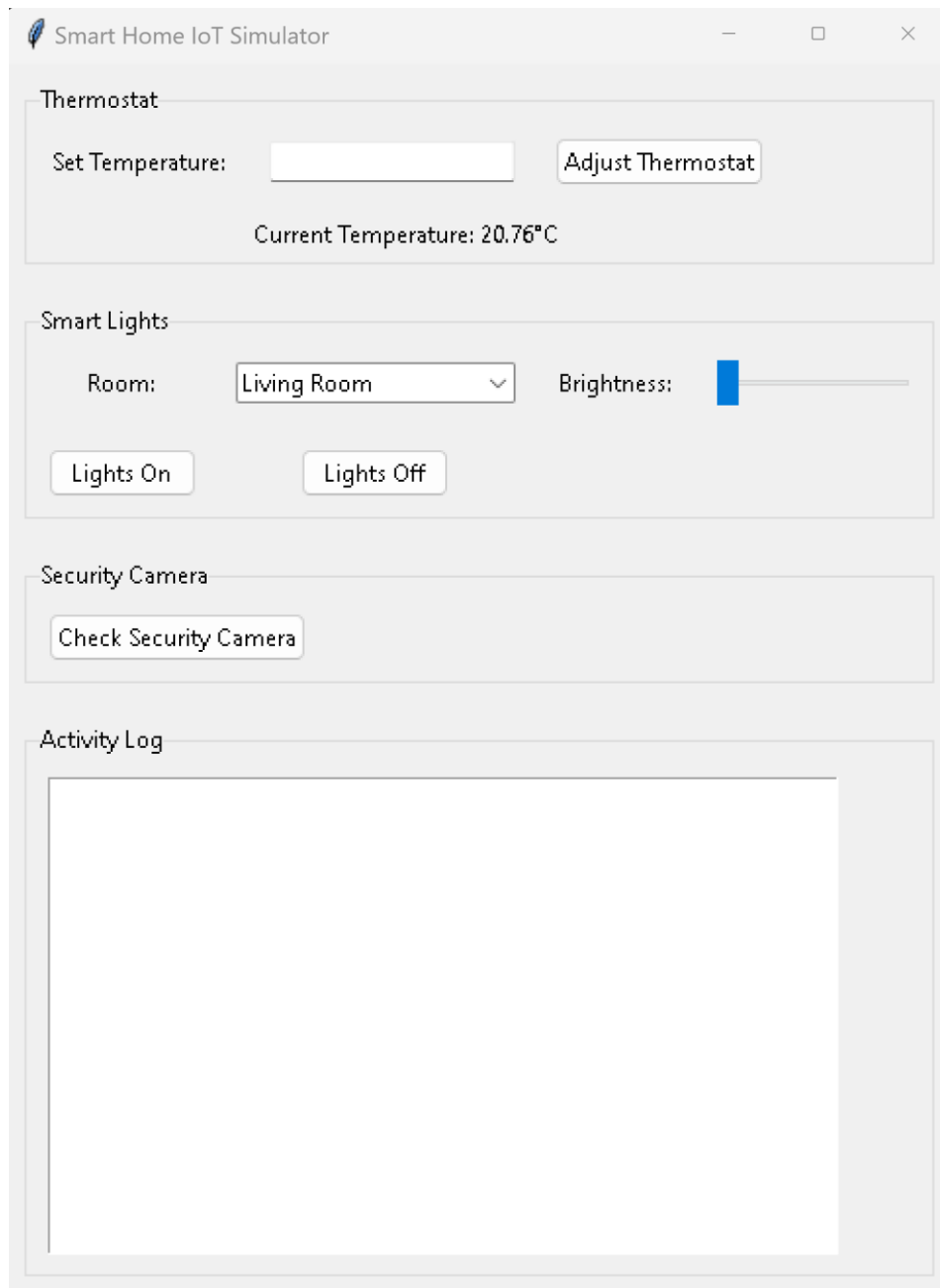
'`turn_lights_on(self)`' : turns on the lights in the selected room with the specified brightness.

'`turn_lights_off(self)`' : turns off the lights in the selected room.

'`check_security_camera(self)`' : simulates checking the security camera for movement detection.

'`update_temperature(self)`' : updates the temperature in the simulation (scheduled every 5 seconds).

'`update_temperature_graphs(self)`' : updates the temperature graphs on the dashboard.

Smart Home IoT Simulator — □ ×

Thermostat

Set Temperature: [   ]　　Adjust Thermostat

Current Temperature: 20.76°C

Smart Lights

Room: [Living Room ▾]　Brightness: [▮————————]

Lights On　　　Lights Off

Security Camera

Check Security Camera

Activity Log

How to run the Simulation

1. Make sure that you have Python installed on your machine.
2. Install all the required libraries. You can install them with `pip install tkinter pillow` on cmd
3. Run the smart_home_simulator.py
   `python smart_home_simulator.py`

Using the Dashboard

1. Thermostat:
   - Set the desired temperature in the entry field.
   - Click on the "Adjust Thermostat" button to simulate thermostat adjustment.
2. Smart Lights:
   - Choose a room from the dropdown menu.
   - Adjust the brightness using the slider.
   - Click on "Lights On" or "Lights Off" button to turn them on or off.
3. Security Camera
   - Click on the "Check Security Camera" button to check for movement.
   - The activity log will display whether movement is detected or not. If detected, the log will display "Movement is detected. Intruder detected! Capturing snapshot." The camera will capture an image and display the captured image in a new window. If no movement is detected, the log area of the dashboard will display "No movement detected."
4. Activity Log: displays a history of activities, including thermostat adjustments, light status changes, and security camera checks.

Notes

- The simulation includes random movement detection for the security camera.
- Temperature is simulated and updates every 5 seconds.
- This is a simple code and simulates sensor data.
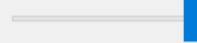
Smart Home IoT Simulator

Thermostat

Set Temperature: `22` [Adjust Thermostat]

Current Temperature: 20.45°C

Smart Lights

Room: `Living Room ▾`   Brightness: ——————————▮
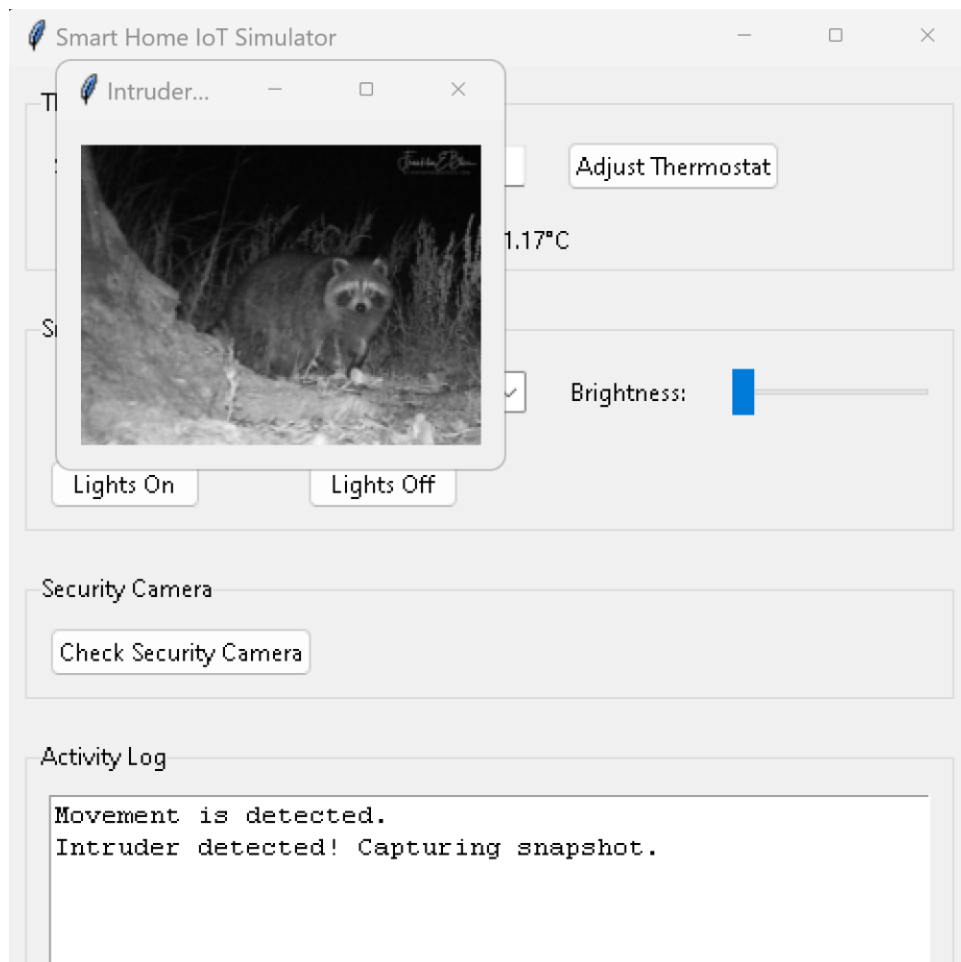
[Lights On]   [Lights Off]

Security Camera

[Check Security Camera]

Activity Log

```
Heater is turned on until 22.0°C.
Lights in Living Room are turned on with brightnes
s 100%.
```

Test cases for Smart Home IoT Simulator

Test Case 1: Adjust Thermostat

Description: Verify that the thermostat adjusts the temperature correctly based on user input.

Steps: Open the simulator. Enter a desired temperature (e.g., 25°C) in the thermostat. Click the "Adjust Thermostat" button.

Expected Result: The simulator should log the activity, indicating whether the heater or AC is turned on, and the current temperature label should be updated.

Test Case 2: Turn Lights On and Off

Description: Confirm that smart lights respond to user commands to turn on and off.

Steps: Open the simulator. Choose a room from the dropdown menu. Adjust the brightness using the slider. Click "Lights On" or "Lights Off" button.

Expected Result: The simulator should log the activity, indicating whether the lights are turned on or off. The room's brightness status should be updated.

Test Case 3: Security Camera Movement Detection

Description: Check if the security camera detects movement as expected.

Steps: Open the simulator. Click the "Check Security Camera" button.

Expected Result: The simulator should log the activity, indicating whether movement is detected or not.

Test Case 4: Temperature Graph Updates

Description: Validate that temperature graphs are updated when interacting with the thermostat and lights.

Steps: Open the simulator. Adjust the thermostat or turn lights on/off.

Expected Result: The temperature graphs on the dashboard should be updated accordingly.

Test Case 5: Invalid Temperature Input

Description: Ensure the simulator handles invalid temperature input gracefully.

Steps: Open the simulator. Enter a non-numeric value or an empty string in the thermostat. Click the "Adjust Thermostat" button.

Expected Result: The simulator should log an error message, indicating the input is invalid.

Test Case 6: Simulate Continuous Temperature Updates

Description: Check if the simulator continuously updates the temperature.

Steps: Open the simulator. Observe the current temperature label.

Expected Result: The temperature label should be continuously updated every 5 seconds.

Test Case 7: Change Room and Adjust Lights

Description: Verify that changing the room in the smart lights section updates the light status and brightness.

Steps: Open the simulator. Choose a room (e.g., Living Room). Adjust brightness and turn lights on/off. Change the room (e.g., Bedroom) and observe the status.

Expected Result: The simulator should display the current light status and brightness for the selected room.

Notes

- These test cases cover various aspects of the simulator, including thermostat adjustments, smart lights control and security camera checks
- The test cases assume the initial state of the simulator is reset for each test.