

Smart Home IoT Automation Simulator Documentation

Written and Implemented By

JAMOLOV ISROILBEK (DXFV5Y)

Course: **Python**

Faculty of Informatics
Eötvös Loránd University

Date: **08/11/2023**

INTRODUCTION

This documentation provides a comprehensive guide to the Smart Home IoT Automation Simulator. The simulator emulates various IoT devices within a smart home environment, such as smart lights, thermostats, and security cameras, and includes a central automation system and a GUI dashboard for interaction and control.

CLASS DESCRIPTIONS

1. IoTDevice (Abstract Base Class)

- Base class for IoT devices.
- Attributes:
 - 'device_id' : Identifier for the device.
 - 'status' : Current status of the device (On/Off).

2. SmartLight

- Represents a smart light with brightness control.
- Inherits from IoTDevice.
- Additional Attribute:
 - 'brightness' : Brightness level of the light (0 to 100).

3. Thermostat

- Represents a thermostat with temperature control.
- Inherits from IoTDevice.
- Additional Attribute:
 - 'temperature' : Temperature set on the thermostat (15 to 30 degrees Celsius).

4. SecurityCamera

- Represents a security camera with infrared and recording capabilities.
- Inherits from IoTDevice.
- Additional Attributes:
 - 'infrared' : Boolean indicating if infrared is enabled.
 - 'recording' : Boolean indicating if recording is enabled.
 - 'motion_detected' : Boolean indicating if motion is detected.

5. AutomationSystem

- Manages IoT devices and their interactions.
- Methods include *device addition*, *automation_rule_execution*, and *device_behavior_simulation*

6. GuiController

- Manages the GUI for the smart home dashboard

- Interacts with the 'AutomationSystem' to reflect and control the state of the devices.

METHOD EXPLANATIONS

» IoTDevice Methods:

- ⊗ 'turn_on()' : Turns the device on.
- ⊗ 'turn_off()' : Turns the device off.

» SmartLight Methods

- ⊗ 'set_brightness(brightness)' : Sets the brightness of the light.

» Thermostat Methods

- ⊗ 'set_temperature(temperature)' : Sets the temperature of the thermostat.

» SecurityCamera Methods

- ⊗ 'toggle_infrared()' : Toggles the infrared mode.
- ⊗ 'toggle_recording()' : Toggles the recording mode.

» AutomationSystem Methods

- ⊗ 'add_device(device_type, device)' : Adds a device to the system.
- ⊗ 'execute_automation_rules()' : Executes predefined automation rules.
- ⊗ 'simulate_device_behavior()' : Simulates random behavior of devices.

» GuiController Methods

- ⊗ 'setup_gui()'
 - Sets up the GUI components for the smart home dashboard.
 - Initializes frames for light, thermostat, and camera controls, along with status listbox and automation rules labels.
 - Each device frame contains a toggle button, a status label, and a control element (scale for light brightness and thermostat temperature).
- ⊗ 'update_device_status()'
 - Updates the GUI to reflect the current status of all devices.
 - Calls 'execute_automation_rules()' of the 'AutomationSystem' to ensure the latest state is reflected.
 - Updates the listbox with the status of each device (light, thermostat, camera).
- ⊗ 'toggle_device(device_type)'
 - Toggles the on/off status of a specified device (light, thermostat, camera).

- Invokes `'turn_on()'` or `'turn_off()'` methods of the respective device based on its current state.
- ⊗ `'adjust_brightness(value)'`
 - Adjust the brightness of the SmartLight
 - Takes a brightness value as input and sets it using `'set_brightness()'` of the SmartLight object
 - Also handles turning the light on or off based on the brightness level
- ⊗ `'adjust_temperature(value)'`
 - Adjust the temperature of the Thermostat
 - Takes a temperature value as input and sets it using `'set_temperature()'` of the Thermostat object
 - Updates the GUI to reflect the new temperature setting.
- ⊗ `'random_detect_motion()'`
 - Simulates random motion detection for the SecurityCamera
 - Toggles are `'motion_detected'` attribute of the camera.
 - Updates the GUI to reflect the change in motion detection status.
- ⊗ `'run()'`
 - Starts the main loop for the GUI
 - Performs an initial update of the device statuses before entering the main loop

INSTRUCTIONS TO RUN THE SIMULATION AND USE THE DASHBOARD

1. Setting up the environment

- Ensure Python is installed on your system.
- No additional dependencies are required for this simulation as it uses standard Python libraries.

2. Running the simulation

- Open a command line interface (CLI).
- Navigate to the directory containing the Python script.
- run the script by typing `'python script_name.py'`, replacing `'script_name.py'` with the actual file name.

3. Using the dashboard

- Once the script is running ,a GUI window title "Smart Home Dashboard" will appear.

- The dashboard has separate control sections of the SmartLight, Thermostat, and SecurityCamera.
- Use the toggle buttons to turn each device on or off
- Adjust the brightness of the SmartLight and the temperature of the Thermostat using the provided sliders.
- Simulate motion detection for the SecurityCamera using "Random Detect Motion" button.
- The status listbox at the top of the dashboard displays the current status of all devices.
- The GUI is interactive and will automatically update the device statuses based on user actions and simulated behavior.

TEST CASES FOR SIMULATION AND AUTOMATION SYSTEM

1. SmartLight control test

- Turn the SmartLight on and off using the toggle button.
- Adjust the brightness and verify that the status listbox reflects these changes.

2. Thermostat control test

- Toggle the Thermostat on and off
- Change the temperature setting and check if the new temperature is accurately displayed in the status listbox.

3. SecurityCamera functionality test

- Test the camera's response to the "Random Detect Motion" button.
- Verify that the motion detection status updates correctly in the GUI.

4. Automation rule verification

- Ensure that the light turns on automatically when motion is detected by the SecurityCamera.
- Confirm that the infrared mode of the camera is activated when the SmartLight is off.
- Confirm that the camera is recording only when it is on and may not record even though it is on.

5. GUI responsiveness and accuracy test

- Interact with the dashboard controls and observe the responsiveness of the GUI.
- Verify that the GUI accurately reflects the current state of all devices and responds correctly to user inputs.

6. Simulated bahvior test

- Allow the simulator to run for a period, during which it randomly changes the state of devices, brightness of the SmartLight, temperature of the Thermostat, and Recording status of the SecurityCamera.
- Check that the GUI accurately reflects these random changes.