

Helion's Snapshot Module

Nii-Kwartei Quartey

Christopher Newport University

July 26, 2020

Abstract

During my undergraduate research, I spent my time working with a home automation program known as Helion, specifically, its Snapshot module. I was tasked with learning new material and completing part of the webpage that were unfinished. I also had to get a little creative when working on a design that users could find appealing. There were times I found working on Helion difficult but overall, working with Helion's Snapshot Module is something that will help me improve with my undergraduate studies.

Introduction

Helion is a natural home automation which generates scenarios by identifying the regularities in user-driven home automation sequences, which are generated by the users. Helion's Core hypothesis is "home automation sequences created by humans are implicitly natural therefore, we can use statistical language modeling to analyze corpora of sequences and predict useful home automation scenarios that enable the design and evaluation of security systems" (Manandhar). It collects routines from users through surveys that collect information on routines specified in a natural language format and clues for when routines typically execute, which are known as execution indicators.

I was working with a part of Helion known as the Snapshot Module. Snapshot tracks the states of individual devices and the home as events that are executed in the home. For example, the "locked" state of the door lock or setting the user's home mode as "home" or "away"



In the image above, I've attached an example of possible states, or events, that Snapshot executes. Snapshot takes images of the current house states and makes predictions based of the images.

Objectives

When I started working on Helion, there were some tasks I was assigned to complete as I made progress, here are the objectives I was able to complete during my time:

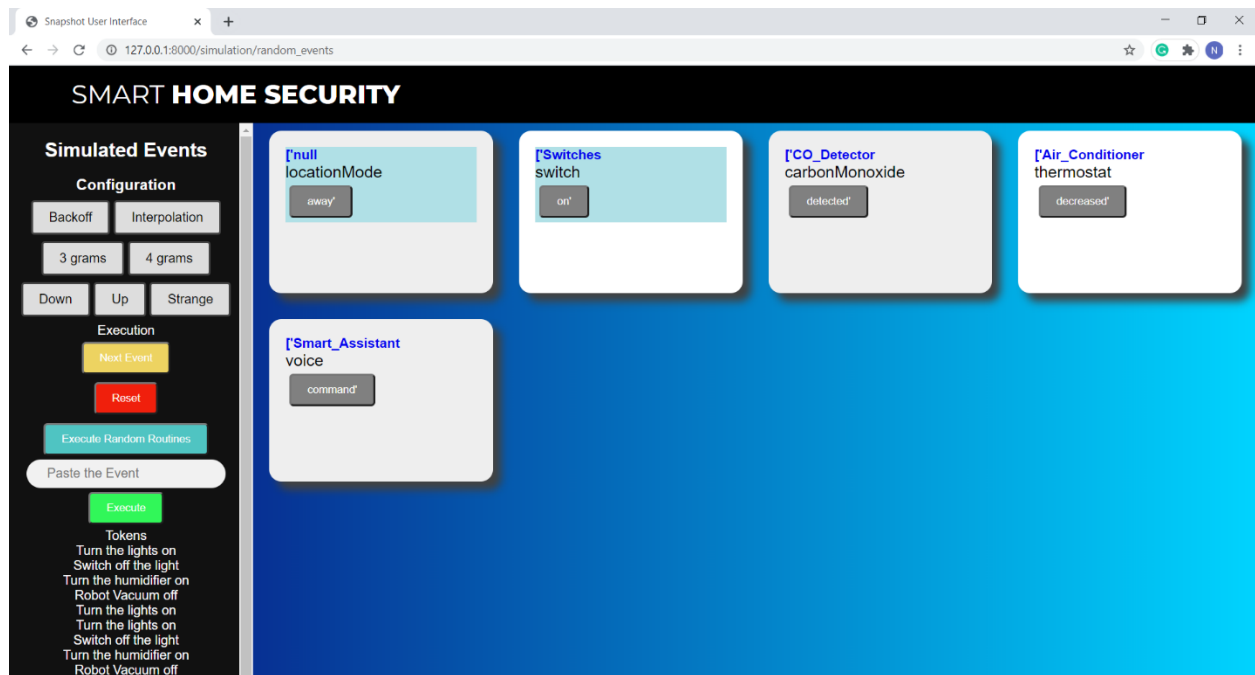
- Understanding Helion's Execution Engine
- Read through Helion's publication to understand how I would be using the snapshot module user interface
- Learn how to use Python Django
- Add Linux to my program to run the Helion program
- Implement my custom webpage to work with the Helion program
- Implement an event search bar into the program
- Making the features of the original Snapshot module user interface given to me work with the webpage I created (Ex. Reset and Executing Random Events)

Though I completed many of the objectives I was tasked with, there were a few I was unable to get to. Implementing so of the features listed above or studying the information required to start on the program took longer than expected:

- Have a working delete button for events in the snapshot user interface
- Implement an event search panel for user convenience

Implementations

The main task of working with Snapshot was to make the webpage interface more user friendly. In the image below, I've attached the result of what I have created



Some of the features I implemented to make the interface more user friendly were the colored execution buttons, the configuration buttons change colors once hovered over, a gradient background, the side bar for all functions, and an autocomplete search bar. As mentioned in the objectives section, a feature I was unable to implement was a delete method. I had planned to have an “x” mark on the top right corner of every event and once clicked on would remove the event. Under the green execute buttons, the Tokens section, was supposed to be a history bar. The events listed under “Tokens” were an example of how the history bar would work like, though I was unable to get to that as other implementations, like autocomplete took more time for me.

Difficulties

During my time working with Helion's Snapshot Module, I encountered quite a few problems. The first minor issue I encountered was with Linux. I installed Linux with a dual boot on my computer to switch between Linux and Windows as the commands needed to run Helion were not compatible with Windows. When switching between the two, I ran into an issue once I logged in, I would immediately be sent back to the login screen, even if my password was correct. I tried the troubleshooting fixes, but none worked for me, so I ended up reinstalling Linux. The last problem I encountered was with starting up after dual booting. My screen would remain black once I selected to boot in Linux. I tried the fixes for this problem and it still didn't work, so I reinstalled Linux yet again.

Another minor problem I encountered while working was with CSS and JavaScript files. When implementing my paper from HTML to Django, I was anticipating it would be as simple as copy and paste. I had stopped working in Django because I felt I could complete an interface faster in HTML since I already had knowledge working with it, which worked in my favor because I

completed more than I expected to because of that. When it came to transfer my work into Django, I didn't know the buttons all needed a csrf token tag to work, I had to adjust URL files, and set proper directory paths. CSS and JavaScript directories were set correctly but didn't work for some reason, even with a load static tag. I ended up using style and script tags to get them working, making my index.html file roughly 800 lines when it only should have been about 300.

The biggest issue I faced was with the autocomplete search bar on the program. The autocomplete feature works just like a google search. Once a few characters are input, the search bar will print suggestions for the user, though the issue I had was that the search bar would return the wrong result



In the image shown above, the program was supposed to execute a light bulb event but returned a null event. I knew where the issue was within the code as I based the code for the autocomplete off how snapshot executes random events by using Python's `random.choice()`. I had set my code to return only one line from the list of available events because I didn't know how to search for the name of the specific event in the text file. This was a problem I worked with my mentor on how to fix and we figured out we needed a `request.post()` method for the button I had created to search in the text file.

Conclusion

In conclusion, my experience working with Helion has helped me to become a better programmer in my undergraduate studies. I learned how to create my own functions with Python Django, I've gained experience with JavaScript, Linux, and Django, and I learned how to implement different programming languages to work as one program. I believe I gained the most experience with Django as I now know how to get HTML working with Django, and I have an understanding on how to use views, URL, and templates with Django. I also believe working on

Snapshot has improved my knowledge with HTML and CSS. I had created a website from scratch prior to the Snapshot webpage though looking at it now, I see how much I have improved with organization and styles. Overall, I believe what I have learned from working with Helion with help me with my undergraduate studies.

References

Manandhar, Sunil; Moran, Kevin; Kafle, Kaushal et al. *“Towards a Natural Perspective of Smart Homes for Practical security and Safety Analyses”*

<https://www.adwaitnadhkarni.com/pdf/manandhar-oakland20.pdf> pp.1-4