Project Cecropia

An Integrated Server for the SlothBot

Yilun "Allen" Chen

1 Overview

1.1 Abstract

Project Cecropia aims to provide real-life application for the SlothBot swarm. The SlothBots are fall-safe wire-traversing robots developed in Robotarium at Georgia Tech. They will later be deployed to Atlanta Botanical Garden for enviornment monitoring purposes. Project Cecropia offers a solution to allow the collection of the data trasmitted from the SlothBots, along with the capability to store and visualize the data on a web page that it serves.

This document aims to provide an overview of how the project works for future developers of the project. Details of how the project is structured, how it achieves different functionalities, how to deal with emergencies/bugs will be provided.

1.2 Tech Stack

Here is a general overview of how the project is structured. Main building blocks of the project and their rationales are provided, so that future developers of the project may use this as a reference.

Web Framework: Express & Node.js
Node.js as a web framework has high integrity when it comes to building web applications. Its package manager NPM offers a wide variety of packages, providing almost all functionalities desired.

Database: MongoDB
MongoDB uses JSON-like syntax as the form of its document. This feature/characteristic means that its documents can be easily translated into a form that is easy for JavaScript to use.

User Interface: JQuery & Bootstrap jQuery and Bootstrap offer a wide choice of tempates that we can straight up use and build upon, while adding a lot of additional handy functionalities that can be used in overall development process.

Package Management: NPM
Almost all JavaScript packages can be found on NPM. It is extremely easy to use and it also offers great mobility - When you try to move the server to another host, you can easily setup the needed environment.

Data Visualization: CanvasJS (JavaScript Library)
Canvas JS is a licensed JavaScript library that allows you to create graphs with massive amount of data points. It has superb performance in handling mass data and is highly configurable.

Server/Data Visualization API: DataParser (Customized JavaScript function)
A customized JavaScript class that takes in all the data points retrieved, sort them by timestamp and dates and outputs a CanvasJS-friendly cluster of data to be put directly into the data-visualization process.

Testing Platform: RDP (Customized Python Script)
A script to generate continuous mock data and send it over to the server. Depends on the pacakge 'request'.

2