Kenniel Torres

Assignment 4 - Report

I. Overview

This language was built from Python with the use of the PLY library in order to bring forth a declarative and functional approach. It is the basis of a language that will facilitate the communication between devices. It will provide simple functionality to create local servers and allow communication with external servers.

II. Description

This language was built with simplicity in mind; thus, some features are very basic. Nonetheless, this new language made for simplifying communication between devices can startup a server and/or client with just a few commands. Once the project is running in a terminal, you have the freedom to use the default server port of 8000, or set it to any number you wish, as long as it is 1024 or greater, using the command *SET_SERVPORT {port}*; this must be done prior to starting up a server and client. If you remain unsure of the server's port number, use the command *GET_SERVPORT*. Afterwards, use the command *START_SERVER* to create and initialize a server. Then, make sure the client's port is the same as the server's by using the command *GET_CLIPORT*. If they are different, use the command *SET_SERVPORT {port}* to match the port of the server. Finally, use the command *START_CLIENT* to create and initialize the client, and automatically connect to the server.

Once the server and client are connected, you may communicate with one another in a back-and-forth manner. The client sends the first message, and waits for the server's response. Once the server's message is sent, it waits for the client's response and the cycle of communication begins. If you want to disconnect and terminate a client, use the command *CLOSE_SESSION*; this will also return you to the main program. If you wish to stop the server, use the key bind '*ctrl + c'*. Finally, in order to terminate the main program, use the command *CLOSE*.

III. Usage & Commands

The port has to be the same for both the server and client to establish a connection, and must be 1024 or greater. The server has to be started before the client. In order to run a server and client simultaneously, two separate terminals must be open: 1st with the server and the 2nd with the client. Only one server and one client can be started at a time.

- START_SERVER
  - ⇒ Allows you to create and initialize a server with a default port of 8000.
- SET_SERVPORT {port}
  - ⇒ Allows you to set the port in which the server will listen for connections, prior to starting one.
- GET_SERVPORT
  - ⇒ Allows you to access the current server's port, prior to starting one.
- START_CLIENT
  - ⇒ Allows you to create and initialize a client, and connect to a server automatically, with a default port of 8000.
- SET_CLIPORT {port}
  - ⇒ Allows you to set the port in which the client will try to connect to the server, prior to starting one.
- GET_CLIPORT

⇒ Allows you to access the current client's port, prior to starting one.

- CLOSE_SESSION

    ⇒ Allows you to terminate a client once it has been started.

- CLOSE

    ⇒ Allows you to terminate and exit the main program.