The Laptop Orchestra of Louisiana (LOLs) was started in 2008 but encountered a problem by wanting to engage undergraduate students with no computer experience in the performance of laptop-mediated music. The challenge was to consistently and reliably launch bleeding-edge software with little downtime as possible to create a smooth, seamless performance without long pauses between pieces.

By using the Internet of Things for inspiration, the Orchestra of Things (OoT) was born. In the OoT, all objects (computers, joysticks, phones, speakers) are considered things. Things can send or receive control signals. Things can be “meta” things, containers of other things that pass messages to and from the outside world. Things can do nothing at all. Things can do any or all of these. But they all represent musical objects of one sort or another.

I would like to give all glory and praise to God for keeping and protecting me, my family for encouraging me, the MARC to BRIDGE family at Albany State University for supporting me and Louisiana State University’s Center for Computation & Technology for taking me in as one of their own. I have been showed much love during my stay and I am forever thankful for the experience.

The server is composed of Socket.io, Node.js and JavaScript to run and communicate to the terminal.

Functions that connect buttons to webpages, create/updates the SQLite3 database and sends the homepage to the browser.

Created a table by building and inserting the data cell by cell and row by row with the help of the appendchild function.

Data from the server’s database is retrieved as an object in an array form and placed into the cells.

A refresh button allows the data to be displayed upon request.

I.P. Address, port number and instrument. These string values are then sent to server to store in the database while also generating a specific identification number.

Gimbal beacons were introduced. The configuration to receive information such as the name of the instrument and a specific identification number would be given to each beacon and conveniently provided this information once within range of the server.

An iOS application using the Apple developer and the Swift language was to be created; however, while learning the fundamentals there was an update to the language. This slowed down production and an alternate route had to be created.

In order to incorporate physical instruments and other things, Gimbal beacons were introduced. The configuration to receive particular information is still in the process of researching. It is with high hopes that the proximity function would allow information such as the name of the instrument and a specific identification number would be given to each beacon and conveniently provided this information once within range of the server.

During the research, a few problems were encountered that could have possibly prolonged the finished product.

• An iOS application using the Apple developer and the Swift language was to be created; however, while learning the fundamentals there was an update to the language. This slowed down production and an alternate route had to be created.

• In order to incorporate physical instruments and other things, Gimbal beacons were introduced. The configuration to receive particular information is still in the process of researching. It is with high hopes that the proximity function would allow information such as the name of the instrument and a specific identification number would be given to each beacon and conveniently provided this information once within range of the server.

Future Works

To continue improving the communication of the orchestra, it is with high hopes to incorporate both physical Gimbal beacons and virtual ones. These beacons will be imbedded in the website, physical instruments, speakers, and Apple/Android applications in order to include mobile devices such as phones, iPads and tablets.

References

