Sean Crowley
Camille Bossut
Suma Cherkadi
Elias Reta
Zihuan Wu

Project Proposal Draft

Project Proposal : Lab Implementation and Save States

Group Members

- Sean Crowley
  - 4th year CS
  - C, C++, Java, and Python
- Camille Bossut
  - 3rd year CS
  - Javascript, C, C++, Java, Python, Matlab
- Elias Reta
  - 2nd year CM
  - Java, Python, HTML, CSS
- Suma Cherkadi
  - 2nd year CS
  - Java, Python, C, Scala, Javascript, ReactJS
- Zihuan Wu(Mark)
  - 3rd year CS
  - C, Java, Python and R

Project Goals and Description

Problem to Solve/Goals:

The main goal with the Discrete Convolution GUI is to integrate the existing GUI with React and NodeJs. Implementing all our GUIs in React allows for a standardized implementation across ITS. For this specific demo, some goals include replacing all the current jQuery actions with React functions and modularizing the code using separate React components. The inclusion of React also provides us a streamlined way of saving and restoring states of the GUI. We also aim to make our different components in different files to ensure ease of expandability. This also avoids major conflicts with team members working on the same files.

After integrating React into this GUI, our next goal is to add access to a database. This would allow the GUI states to be saved and restored on a user to user basis and could provide more data points for further analytics. Saving the state of the GUI in a database allows for potential automation of lab checkoffs in the future. Along with this state, the relevant user data that accompanies it is useful in potentially tracking the growth and understanding that the user gains from interacting with the GUI. One of our goals as a team is research and find the best technology to connect the GUI to the database. We aim to contact teams from previous semesters to gain insight into how the database was accessed in the past.

Potential Issues:

The potential issues we may encounter are mainly setbacks due to most of us being unfamiliar with NodeJS and having little to no experience with databases. This means that we'll need to put in additional effort to learn those skills so that the process can stay at the pace we've planned for. In addition a few members are unfamiliar with JSXGraph, however most of the code that involves plotting is already complete. Our hope is that any changes to code involving JSXGraph will be minimal; mostly the existing code will need to be wrapped in NodeJS instead of JQuery.

Working Plan Chart

Time Schedule	**Task**	**Responsibilities**
Week 4 *Jan 27-Feb 2*	● Brainstorm project proposal ● Research React and get familiar with the library	Submit final project proposal draft.
Week 5 *Feb 3 - Feb 9*	● Complete final Project Proposal	Submit final project proposal.
Week 6 - 9 *Feb 10 - Mar 8*	● Convert current DisConvDemo to React format	Initialize each of the graphs Link them to HTML components as we did before Improve user input option and customization
Week 10 - 12 *Mar 9 - Mar 22*	● Learn the best way to write to/read from access a database for VIP ● Interface with other teams that have done this in the past	Consult other teams to get on the right track Work on best way to organize the data in the database
Week 13 - 15 *Mar 23 - Apr 19*	● Set up and write to/read from a database with GUI information	Read and write saved state to the database
Week 16 *Apr 20 - Apr 26*	● Final presentation ● Commit to master	Submit final code and presentation

Implementation Tools and Resources

- GitHub: https://github.gatech.edu/VIP-ITS
- Project Documentation - Notebook
- JSXGraph Demos: http://its.vip.gatech.edu/VIP/demos/