

Team Name: Agents of S.H.I.E.L.D. (Systems, HTML Interface, Temporal Logic Division)

Project Name: GridTLI GUI (Project Code Name: Hydra)

Team Members: Alex Bennett, Fan Cao, Kat Elkind



Project Description

The goal of this project was to create a user experience for the Grid-based Temporal Logic Inference (GridTLI) tool in the form of a web application. This tool is a subset of the systems engineering tool, Phoenix. While GridTLI allows biologists to visualize and modify the behaviors of complex biological systems (in particular, genetic logic circuits that are common in synthetic biology), for many it can be very intimidating to access these tools via command line utilities. The web app allows the user to draw or provide Signal Temporal Logic (STL) formulas describing the desired behavior of a system, which can then be exported and analyzed in GridTLI.

Major Software Components

Canvas and Paper.js

The ability to draw in a web app was implemented using the <canvas> tag from HTML5 in addition to the open-source tool Paper.js. A single paper variable was added to the global scope, allowing Paper.js to be handled directly using JavaScript. This is easily scaled such that it would be possible to implement multiple tabs in future versions.

Grid

The graphing area was one of the two Paper layers, controlled by the function drawGrid. Paper.js Rectangles were created in order to form the grid; this choice made them very easy to manipulate individually. Paper.js Lines and PointText objects were used to create the axes and the horizontal axis label.

Canvas

The canvas was the other Paper layer. It is a transparent area that exists in the same area as the grid, and is the space where the user draws and edits the lines. The canvas serves as the Parent to the lines on the graph.

Important Functions

colorBoxes - This function checks to see if a line crosses into a grid box or not. Each time it is called, it iterates through the existing boxes and returns their color to transparent. If a line crosses into the box, the box changes color and the function moves onto the next box.

changeValues - This function calls the new values the user has entered and resizes the grid by calculating the new axis values and calling drawGrid.

drawGrid - This function begins by deleting the existing grid, if one is present. It then uses the current axis values (minima, maxima, and threshold) to draw new rectangles and axes. It does not affect what is currently drawn in the canvas layer. Rectangles are added to their own group.

changePathValues - This function converts the canvas coordinate values into the correct values according to the graph. It is necessary for the data to be useful once exported to GridTLI.

Tools

Tools are built into Paper.js, used to manipulate the drawing area. Specific tools are described below. Each follows a series of checks that ensure each line is a function.

selectLine - This tool deselects the current path and reselects the one that has been clicked on

deleteLine - This tool deletes whichever line is clicked on, regardless of which line is selected

drawLines - This tool draws a continuous line; the number of data points is inversely proportional to mouse speed. Each time it is used, it deselects the current line and begins a new line

drawPoints - This tool can append points to a selected line or draw a new line point-by-point if no line is selected. If the shift key is held down, it can be used to delete a point on an existing line.

movePoints - This tool lets the user move points on a selected line. It checks that the line being drawn is a function even while modifications are being made. It engages smoothing to avoid harsh changes in behavior, but the smoothing is modified to ensure the line is still a function.

downloadCanvas – This tool lets the user download the canvas in JSON format. The JSON file gets downloaded as a data.json.

uploadCanvas – This tool allows for upload of a canvas design saved as a JSON format to the Grid TLI tool. The points on the grid get plotted to the Grid TLI tool.

STL Editor

A multifunctional text editor that performs automatic syntax highlighting on STL logic symbols and operators. Background themes and fonts may be customized to the users preference.

Compile Code

Download the file from GitHub into a single folder. Open index.html in the browser.