Web-based animation with Galant

Professor Matthias Stallmann

# Contact Information

*Name(s) and email(s) of mentor(s) from Sponsoring Company/Organization (one by line if multiple).  
Formatting as “First Last <email@email.com>" is preferred.*

Matthias Stallmann <mfms@ncsu.edu>

# Sponsor Background

*In just a couple of sentences, describe the sponsoring entity to introduce students to your organization and mission in a way that shows how this project is relevant to you.*

Dr. Stallmann is a professor (NCSU-CSC) whose primary research interests include graph algorithms, graph drawing, and algorithm animation. His main contribution to graph algorithm animation has been to make the development of compelling animations accessible to students and researchers. See mfms.wordpress.ncsu.edu for more information about Dr. Stallmann.

# Background and Problem Statement

*Provide background on perceived problem for the student team to solve. What is the motivation for this project?*

**Background.**

[Galant](https://github.com/mfms-ncsu/galant) (***G***raph ***al***gorithm ***an***imation ***t***ool) is a general-purpose tool for writing animations of graph algorithms. More than 50 algorithms have been implemented using Galant, both for classroom use and for research.

The primary advantage of Galant is the ease of developing new animations using a language that resembles algorithm pseudocode and includes simple function calls to create animation effects.

The most common workflow is

* Create a graph either externally or using Galant’s graph editor
* Upload an algorithm animation program created externally using a program editor
* Compile and execute the program using arrow keys to step forward and backward in a list of animation events

**Problem statement.**

There are currently two versions of Galant: (a) a sophisticated, complex Java version that requires git, Apache ant, and runtime access to a Java compiler; (b) a web-based version, [galant-js](https://github.com/mfms-ncsu/galant-js) (https://github.com/mfms-ncsu/galant-js), was developed by a Spring 2023 Senior Design Team and enhanced by teams in Fall 2023 and Spring 2024. The latter has been used in the classroom (Discrete Math) and several algorithms have been successfully implemented. However, there are some major (and minor) inconveniences from a usability perspective.

# Project Description

*What are your initial thoughts about a potential solution that addresses the problem presented above? Briefly describe your envisioned software solution to be developed by the student team. Use cases and examples are useful. Provide insight into how this potential solution will be beneficial to end users.*

Some enhancements are required to put the useability of galant-js on par with the original Java version. The Java version has been used extensively in the classroom and in Dr. Stallmann’s research. The JavaScript version already has clear advantages.

The biggest challenge is to establish a mapping between physical positions of nodes on the screen and logical positions of nodes in a file that describes a graph. For most graphs this is simply a matter of keeping track of a scale factor and doing the appropriate transformations during editing (and when an algorithm moves nodes). There are, however, special graphs whose nodes are points on an integer grid. The mapping must be maintained both during editing and algorithm execution.

Other usability enhancements are related to the user interface: placement of menus, keyboard shortcuts, and more general accessibility features.

And there are enhancements related to algorithm implementation and execution.

A detailed list of desired enhancements is in feature-requests.md at the root of the repository in the dev branch:

<https://github.com/mfms-ncsu/galant-js>

To run the current version, go to https://galant.csc.ncsu.edu

The Java version can be downloaded at [https://github.com/mfms-ncsu/galant](https://github.com/mfms-ncsu/galant-js)

# Technologies and Other Constraints

*Provide a list of technologies that you expect students to use. Indicate if each technology is a requirement or a suggestion. If flexible or up to students to choose, please state that. Indicate preferred paradigms (e.g., desktop, cli, web-based, mobile, etc.) to be used in this project for each major component of the expected solution. If flexible, please state that. Indicate if there are any other limitations or constraints you would like the students to know about (e.g., licensing constraints, legal issues, IP issues, etc.), or state that there are none.*

Students are required to learn and use JavaScript effectively. The current JavaScript implementation uses React and Cytoscape for user interaction and graph drawing, respectively. An understanding of Cytoscape, in particular, will be required to address the challenge related to node positions on the screen.

The tailwind plugin in used for style sheets.

**Students will be required to sign over IP to sponsors when the team is formed (please check box only if this is the case for submitted project).**