Software Documentation

GroveDraw v0.5

University of North Carolina Wilmington

Dr. Nathaniel Grove

Dr. Devon Simmonds

Zachary Parks

Table of Contents

[General Project Information 3](#_Toc515917315)

[Project Summary 3](#_Toc515917316)

[Software Information 4](#_Toc515917317)

[Front-end software stack 4](#_Toc515917318)

[**Molecule-Drawing Application** 4](#_Toc515917319)

[**jQuery** 5](#_Toc515917320)

[**Fabric** 6](#_Toc515917321)

[**GroveDraw Source Code** 7](#_Toc515917322)

[**Classes Source Code** 8](#_Toc515917323)

# General Project Information

## Project Summary

(This is the original summary given as a description of Dr. Grove’s project)

Chemistry is described by many as the “central science” – the nexus through which mathematics and the physical sciences are connected to the biological, environmental, and geological sciences. Because of this, many science, technology, engineering, and mathematics (STEM) students are required to take chemistry courses, although the reasons for this are often not entirely clear to them. A prime justification for these requirements is the assumption that a robust understanding of the discipline provides the framework through which learners can begin to develop one of the most important and fundamental scientific skills, that is, the ability to connect underlying atomic and molecular structure with the chemical and physical properties these materials exhibit in the macroscopic world around us. This connection process is a complex one, requiring students to construct and manipulate a series of representations of chemical structure and to use those representations to extract vital information. Unfortunately, many find this process to be quite difficult – and we hypothesize given the sheer number of independent pieces of information that must be juggled simultaneously by the learner in working memory – hindered by issues associated with cognitive load.

Mastering the ability to construct and manipulate Lewis structures is an important first step along the journey to reaching representational competence. Lewis structures, like the one for ethanol shown below, include not only the types and numbers of atoms in the molecule, but also incorporate the sequence in which those atoms are bonded together.

Our previous research sought to understand the barriers that students encountered while learning to construct Lewis structures, and a recent grant from the National Science Foundation has provided financial support that will allow us to take the results of this research and create an adaptive learning system that will support novice chemistry students as they learn to construct these essential representations. As proposed, there will be two key components of the adaptive learning system: (1) a diagnostic tool that will pinpoint a student’s Lewis structure creation ability and (2) a structure creation program that will task students with crafting Lewis structures that will directly address the deficiencies identified by the diagnostic component of the system. The structure creation program will rely on a Socratic-style feedback model to provide targeted guidance to the students as they draw their structures. Work on this grant will require students to collaborate closely with a team of chemists, chemistry education researchers, and computer scientists. Not only will this work provide the ability to create the system but will offer the opportunity to directly interface with end users (novice chemistry students) to produce a user-friendly and intuitive interface.

# Software Information

## Front-end software stack

The software is designed to be a web-based application. The software consists of the actual molecule-drawing application as well as account maintenance and user management capabilities.

### **Molecule-Drawing Application**

The molecule-drawing application, or “canvas”, is written in **JavaScript**, **HTML**, and **CSS**. HTML and CSS are used to display the web page itself. For the display of the canvas page itself, the code is quite simple and can be seen in ***canvas.html*** and ***canvas.css*.** In ***canvas.html***, you will find the various scripts that are included in this page and that which are depended upon by the application.

As of this writing, the [essential](#Backbone Dependencies) dependencies on the canvas page and molecule-drawing application are

***jquery-3.2.1.min.js***

***fabric.js***.

You will also find other dependencies in ***canvas.html*** where most of the molecule-drawing software is implemented. These dependencies are

**Grovedraw.js**

**Classes.js**

**Grovedraw.js** is indeed the “main” source code for the canvas and more in-depth in a later [section](#GroveDraw Source Code). **Classes.js** is a separate source code file (that could be combined with **Grovedraw.js** but was not done for an easier development process) that contains information about the **fabric.js** classes made for GroveDraw. (More information is available in a later [section](#Classes Source Code)).

#### **Backbone Dependencies**

### **jQuery**

From the **jQuery** [website](https://jquery.com/):

“jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.”

jQuery is a JavaScript library that greatly simplifies JavaScript programming. It is a library that is powerful, versatile, and easy to learn and use. It is a common library using in what is called “vanilla” JavaScript to make it easier to control event-driven functions and programmatic actions. Canvas uses jQuery in the same way many other projects may use it, which is to allow quick access to page elements in order to use them in the code. Because of this, as of now, jQuery is an essential dependency of GroveDraw and its script ***must*** be included in the ***canvas.html*** file.

Currently, GroveDraw v.0.5 uses **jQuery 3.2.1.**

### **Fabric**

From the **fabric.js** [website](http://fabricjs.com/):

“Fabric.js is a powerful and simple  
JavaScript HTML5 canvas library

#### Fabric provides **interactive object model** on top of canvas element

#### Fabric also has **SVG-to-canvas** (and **canvas-to-SVG**) parser”

Fabric.js is used in GroveDraw to handle the drawing, manipulation, and implementation of the visual representations of the molecules that are drawn by the students. (The pictures!)

Fabric.js allows for images to be loaded onto a canvas in an interactive object model; what this means is that images can behave just like JavaScript objects, with properties and functions. Fabric.js is what gives the images used to construct the molecules their specific functionalities.

The newest version of **fabric.js** is **2.3.0**

GroveDraw is using **fabric.js 1.7.22 (could probably be updated)**

### 

### **GroveDraw Source Code**

### **Classes Source Code**