**Chemical Structure Simulator (CSS)** Documentation

Presented by  
 **Jorge Marcano  
 Ariel Volovik  
 Samy Arab**

February 16, 2020

# Table of Contents

[Table of Contents ii](#_Toc38007516)

[1 Introduction 1](#_Toc38007517)

[1.1 Purpose of the documentation 1](#_Toc38007518)

[1.2 How to use this document 1](#_Toc38007519)

[2 Prerequisites 1](#_Toc38007520)

[2.1 Getting this application 1](#_Toc38007521)

[2.2 Dependencies 1](#_Toc38007522)

[2.2.1 Java 1](#_Toc38007523)

[2.3 Specifications 1](#_Toc38007524)

[3 Purposes 1](#_Toc38007525)

[3.1 Teaching Aid 1](#_Toc38007526)

[3.2 Research Aid 2](#_Toc38007527)

[4 Features 2](#_Toc38007528)

[4.1 Periodic Table 2](#_Toc38007529)

[4.1.1 Summary 2](#_Toc38007530)

[4.1.2 How to use 2](#_Toc38007531)

[4.1.3 What to expect 2](#_Toc38007532)

[4.1.4 Technical Description 3](#_Toc38007533)

[4.2 Text Input 3](#_Toc38007534)

[4.2.1 Summary 3](#_Toc38007535)

[4.2.2 How to use 3](#_Toc38007536)

[4.2.3 What to expect 3](#_Toc38007537)

[4.2.4 Technical Description 3](#_Toc38007538)

[4.3 Loading Screen 4](#_Toc38007539)

[4.3.1 Summary 4](#_Toc38007540)

[4.3.2 How to use 4](#_Toc38007541)

[4.3.3 What to expect 4](#_Toc38007542)

[4.3.4 Technical Description 4](#_Toc38007543)

[4.4 Tabs 4](#_Toc38007544)

[4.4.1 Summary 4](#_Toc38007545)

[4.4.2 How to use 5](#_Toc38007546)

[4.4.3 What to expect 5](#_Toc38007547)

[4.4.4 Technical Description 5](#_Toc38007548)

[4.5 Lewis Structure 5](#_Toc38007549)

[4.5.1 Summary 5](#_Toc38007550)

[4.5.2 How to use 5](#_Toc38007551)

[4.5.3 What to expect 5](#_Toc38007552)

[4.5.4 Technical Description 5](#_Toc38007553)

[4.6 Three-Dimensional Display 6](#_Toc38007554)

[4.6.1 Summary 6](#_Toc38007555)

[4.6.2 How to use 6](#_Toc38007556)

[4.6.3 What to expect 6](#_Toc38007557)

[4.6.4 Technical Description 6](#_Toc38007558)

[4.7 Screenshots 6](#_Toc38007559)

[4.7.1 Summary 6](#_Toc38007560)

[4.7.2 How to use 6](#_Toc38007561)

[4.7.3 What to expect 7](#_Toc38007562)

[4.7.4 Technical Description 7](#_Toc38007563)

[4.8 Saving/Loading Molecules 7](#_Toc38007564)

[4.8.1 Summary 7](#_Toc38007565)

[4.8.2 How to use 7](#_Toc38007566)

[4.8.3 What to expect 7](#_Toc38007567)

[4.8.4 Technical Description 7](#_Toc38007568)

[4.9 Settings 8](#_Toc38007569)

[4.9.1 Summary 8](#_Toc38007570)

[4.9.2 How to use 8](#_Toc38007571)

[4.9.3 What to expect 8](#_Toc38007572)

[4.9.4 Technical Description 8](#_Toc38007573)

[4.10 Manual Mode 8](#_Toc38007574)

[4.10.1 Summary 8](#_Toc38007575)

[4.10.2 How to use 8](#_Toc38007576)

[4.10.3 What to expect 8](#_Toc38007577)

[4.10.4 Technical Description 8](#_Toc38007578)

[5 Shortcuts 8](#_Toc38007579)

[6 Appendix 9](#_Toc38007580)

[6.1 Structures 9](#_Toc38007581)

[6.2 Sorting 9](#_Toc38007582)

# Introduction

## Purpose of the documentation

The purpose of this document is to allow the user to fully understand all the features of the application and how to effectively make use of them.

## How to use this document

The documentation is split into different sections. Each section will contain a single feature. Each section will be further divided into multiple subsections: general summary of the feature, how to use it, what to expect from it, and a more technical description of the feature. At the end of this document, there will be a list of shortcuts that can be used in the application to speed up some tasks. In the appendix, there will be the features of the algorithms.

# Prerequisites

## Getting this application

The entirety of this application, along with the optional source code, is available on GitHub. To install the executable and/or the source code, please visit and clone:

<https://github.com/volovikariel/ChemicalSimulator>

## Dependencies

### Java

To run this application, the computer running this application must also have JRE (Java Runtime Environment) installed.

## Specifications

Since the algorithm this application is based on is a very broad and thus very computationally expensive, it is highly recommended to have at least 2GB of memory available on the device.

# Purposes

## Teaching Aid

The main purpose of this application would be to help teachers in the classroom when teaching about Three-Dimensional VSEPR modelling. It allows for a much clearer way of visualizing than hand-drawn Three-Dimensional models and is interactive, allowing both the teacher and the students alike to interactive with the molecules.

In addition, it allows for the teaching of likelihood of different molecules in real life, although the algorithm is not as complex as reality due to all the nuances, the teacher can clarify some points and show, out of all the possibilities, which is most likely and why.

If the teacher has a specific model they want to show the class a subsequent day, but do not want to wait for the algorithm to exhaust all the possibilities, both the screenshots and the possibility to save the molecule to load up later can come in handy.

## Research Aid

A secondary use of this application could be in labs and in research in general, although rudimentary, the algorithm can exhaust the possibilities of molecules, thus if the researchers now the composition but want to see what molecules might form and in what shapes, this application can help.

If the researcher needs to save the molecule for later use, they may easily do so. Similarly, if they need a picture of the molecule for a paper of an article, they can save a screenshot of the rendered Three-Dimensional model as a professional model.

# Features

The features that follow are all features that affect the user directly, sparing the overly technical descriptions of the algorithms, which are accessible in the appendix.

## Periodic Table

### Summary

The Periodic Table is available as a form of reference, and as a form of input for the user.

### How to use

The user can drag and drop individual atoms into the top portion of the screen, this will add it to the list of atoms that will be considered for the algorithm. The user can also right click to remove them, or alternatively drag and drop them back into the periodic table. The user can also delete individual atoms by pressing backspace after having added it.

The user can also clear all of the elements on the top portion of the screen, and all the elements previously added to the list of atoms that are going to be used to calculate the solutions, by pressing the “Clear All” button in the top left.

### What to expect

The user can expect intuitive controls for the addition and removal of atoms. The user can also expect to have a periodic table which looks familiar to the one they’re used to, that is to say: it is ordered in the standard form so that the user knows what column the atom is situated in, it is color coded to facilitate finding metals and non-metals and other subgroups, it shows the atom’s atomic number.

### Technical Description

The periodic table is composed of a set of drag-able objects, each object is associated with a single element. The information visible to the user on an element, is part of what's used for the algorithm to calculate the possible solutions. There is more information associated with each element in the CSV file which we’ve created, which include electronegativity and more. When the user adds a single element to the top of the screen, it is added to a list of atoms that will have their solutions calculated by an algorithm. If the user presses backspace, right clicks an object on the top portion of the screen or drags it back into the bottom part of the screen, it is then removed from the list of atoms for which the algorithm with calculate the solutions.

## Text Input

### Summary

The text input is a way for the user to input the set of atoms for which the solutions will be computed. If the user intends to write an equation with many of the same atom, it bypasses the monotony of manually placing the same atom over and over by dragging and dropping from the periodic table.

### How to use

The user can simply start typing the chemical compound’s equation and a text field will pop up, allowing the user to keep track of what has been written. The user can at any time press backspace, and an atom will be removed from the list. The user has **two main ways to input the equation**. The user can type an atom at a time, **HOH**, or they can type **H2O**, both will be properly interpreted. The input is **case sensitive**, this is to avoid bizarre cases such as inputting for example CO instead of Co (Carbon Monoxide vs Cobalt).

### What to expect

The user can expect the input to be rather intuitive and responsive. They can expect to have a seamless transition between using the periodic table to add elements and the text input, for instance, if the user wants to add several carbons, the user can drag and drop a single carbon and then using their keyboard, write how many carbons they want.

### Technical Description

The user can add atoms either by dragging and dropping atoms to the top part of the screen, and then removing them by right clicking them or dragging them back down. The user can also add atoms by simply typing them in, for instance, H2O, C2H4, and so on. Each way has a slightly different system for parsing.

a) If the user adds elements by using the keyboard:

1) If an element is followed by another, like OH, then it will be interpreted as one O that will be added to the list of atoms used for calculations.

2) If an element is followed by a number, then that is the amount of the previous element that will be added to the list of atoms used for calculations, like H2O will add 2 H’s.

b) If the user presses backspace:

If there are elements present on the portion of the screen, and the user presses backspace on one of these elements, then one copy of that element will be removed from the top portion of the screen. For instance, if the user added two H’s and typed in an O, there are only two H’s in the top portion of the screen, and in the text, it’s written H2O, if you press backspace once, the O will get removed from the text, if you backspace again, it’ll turn from H2 to H in the text, and it will remove copy of the element from the top portion of the screen.

c) If the user adds an element from the periodic table:

The element added will simply be added to the list which is seen in the text in the bottom right.

## Loading Screen

### Summary

The loading screen is visible whilst the algorithm is calculating the solutions. This is after the user has pressed enter and is ready to see their results, but before the results have been finalized.

### How to use

The user can leave the application whilst the algorithm is doing its job, the user will receive a notification in the bottom right of their screen once the algorithm has finished.

### What to expect

If the user wishes to look at the loading screen, they will see molecules swirling around.

### Technical Description

The loading screen is a loop that runs while there’s been no solution received. It’s H2O’s molecular representation spinning around an axis. The actual algorithm is running on a separate thread in the meanwhile.

## Tabs

### Summary

The tabs are what the Lewis Structure and the Three-Dimensional Display are contained in. The user can choose which tab they want selected by clicking the numbered tabs at the top of the screen.

### How to use

The user can play around with the Lewis Structure and Three-Dimensional Display in one tab, and then go to the next. The user can, easily navigate between tabs by tapping the left and right arrow keys once they have clicked on a tab (or if it has just loaded). If the user clicks elsewhere, they will need to re-highlight the tabs on the top before being able to move around with the arrow keys. The user can always change tab by simply clicking the desired tab.

### What to expect

The tabs are generally **ordered in terms of likeliness** to appear, which is calculated by the algorithm, so the solution in tab 1 is generally more likely to appear than in tab 10. Do note that this isn’t entirely accurate because there are many things to consider. Once the user has rotated or moved the molecule, it remains in that position, so the user can compare their angles and such by jumping between tabs.

### Technical Description

Each tab contains a specific solution given by the algorithm. During the loading phase, it calculates all of the Three-Dimensional rendering along with the Lewis structure rendering. The positions and transformation done on one tab is saved throughout the life of the tab, meaning that switching tabs will not reset it.

## Lewis Structure

### Summary

The Lewis Structure is the 2D representation of the molecular compound.

### How to use

The user can click and drag to **translate**/**move** the Lewis Structure around, the user can also **zoom** in and out with the scroll wheel if the Lewis Structure is either too big or small to fit on the screen, or simply wants to single out an aspect.

### What to expect

The user can expect a simple to understand Lewis Structure. This is not a perfect Lewis Structure, it does not show lone pairs, but it provides enough information as to how the specific solution looks.

**Note:** Sometimes, due to the rudimentary Structure logic, there may be times where some letters may overlap with others.

### Technical Description

The Lewis Structure is generated from a specific solution given by the algorithm. It takes the solution and recursively places the atoms in their respective location, labelled by the element’s symbol and a line per bond between elements. Loops within the molecule are handled first and hence are generally in the center.

## Three-Dimensional Display

### Summary

The Three-Dimensional Display is a 3D representation of the chemical compound (the molecule).

### How to use

The molecule, or any ion can be **translated**/**moved** by pressing down the left mouse button and moving the cursor to where you want the group to be.

The molecule (not ions since they are symmetrical spheres) can similarly be **rotated** by pressing down the right mouse button and moving the cursor.

The molecule can be **zoomed in and out** by using the scroll wheel.

A molecule’s atom can be hovered over and its **symbol** will be **displayed** at the bottom of the screen.

### What to expect

The user can expect the angles between atoms to be close to what they are theoretically according to the VSEPR model. The user can expect to see the proper bond counts for each solution. They can also expect to see the formal charge on each atom if it is not neutral. The atoms are distinguishable one from the other, for instance hydrogen atoms as seen in the group are of a lighter color and of a smaller size. The user can also see its symbol by hovering over it.

### Technical Description

The molecular representation of the compound is created with a set of spheres to represent the atoms, and cylinders to represent the bonds. The Three-Dimensional representation is generated from a specific solution given by the algorithm. It takes the solution and recursively places the atoms in their respective location. Loops within the molecule are handled first and hence are generally in the center. The user can rotate the 3D compound, this is done by rotating the axis itself on which the whole compound is reliant. Translating simply changes the coordinates of the group of spheres and cylinders.

## Screenshots

### Summary

The screenshots allow the user to export the models as an image to insert in another document easily or simply to have as a PNG.

### How to use

To use, the user must simply go to the menu and click **Save**, from there they can choose the type of screenshot they want to save.

### What to expect

There are 3 types of screenshots. The first, **Save Screen**, will save the whole applications screen. Since this is the only one available either in the Selection or the Loading page, it is mostly used when wanting to save the input page. The second, **Save 3D**, is only available in the Results page, and will save a screenshot of the Three-Dimensional model of the currently selected solution. The third, **Save Lewis**, is also only available in the Results page, and will save a screenshot of the Lewis Structure of the currently selected solution.

### Technical Description

This feature uses a simple built-in framework that takes the screen and re-renders it as a PNG image.

## Saving/Loading Molecules

### Summary

This feature allows the user to save a specific solution found so that it can be loaded at any other moment in the Selection page, thus saving a lot of time.

### How to use

To save, the user must simply go to the menu, click **Save**, and then choose the **Save Molecule** option. This will prompt a window to select where to save the file.

To load, the user must simply go to the menu, click **Algorithm**, and then choose the **Load Molecule** option. This will prompt a window to choose the file to open.

### What to expect

The user can expect a **.mol** file type, standing for a **molecule** file. This file is a type used specifically for this application and will not work with other applications. When loading the file, the application will load it as if it came directly from the algorithm itself, but must faster due to not having to calculate the solutions. It will similarly load the Results page, as if it were a normal input.

### Technical Description

The file is saved using the same interface used for the communication between the application and the algorithm. This allows for the difference between loading a file and using the algorithm to be minimal. The application then has no difficulty reading from the file efficiently.

## Settings

### Summary

### How to use

### What to expect

### Technical Description

## Manual Mode

### Summary

### How to use

### What to expect

### Technical Description

# Shortcuts

F1 – Help

ESC – Close the window

ENTER – Submits the entered list of atoms or returns to the selection page

# Appendix

## Structures

## Sorting

The sorting is done using a bubble sort algorithm, but to use it, we need to have values for each of the atomic elements. Luckily, each of them has an atomic number, which is basically the way they’re ordered from lowest, to highest atomic mass. To use the bubble sorting algorithm, first we convert them to their corresponding atomic numbers by accessing our CSV data, sort them, and turn them back into their symbolic representation.