Project Overview

CH4 is an organic chemistry application intended to help students learn about organic chemistry and organic compounds via hands-on learning. It allows students to create organic molecules, learn about their properties and interact with their reactions.

Key Algorithms and Features

Splash Screen

The animated splash screen shows an animated app logo made up of organic atoms.

Chart, bubble chart

Description automatically generated

Project View

After the splash screen loads, the project view screen is entered. The user can create a new .ch4 file, open an existing .ch4 file or open recent files from “Recent” and sample files created by me from “Samples” for users to view.

Chart, treemap chart

Description automatically generated

Workspace

When a user opens a file or makes a new file, they will be brought to the Workspace screen.

Chart, bubble chart

Description automatically generated

Features

1. Drag and drop atoms

By clicking on an atom in the “Atoms” titledpane, the atom will appear in the canvas. The atom is draggable. By right-clicking on the atom, users can create bonds and delete the atom. The atoms also have custom tooltips which display useful information about the atom for users to learn.

1. Canvas

The canvas is zoomable by scrolling your mouse and pannable via holding the right mouse button and dragging over the canvas.

1. Creating bonds

Bonds are created via rightclicking on an atom in the canvas and dragging the other end of the bond to another atom. If the bond is successfully formed, the target atom flashes green. A bond can be deleted by rightclicking on it and selecting the “delete” option. The bond is bound to the atoms, so when the atoms it is bound to are dragged, the bond moves with the atoms. Legend (Bonds) TitledPane is for the user’s reference; Bonds cannot be created from there.

1. Clear All Atoms

All atoms will be cleared from the canvas.

1. Determine Properties

If the molecule is complete and the user clicks this button, CH4 will analyse the molecule created by the user and display its properties.

Properties:

* Mass of molecule in g/mol
* Formula of molecule created
* Identifying the number and types of organic functional groups present in the molecule. This was done by iterating through the molecule multiple times and by detecting and analysing the bonds and position of atoms in the molecule.
* Reactions and other properties of the molecule, derived from the functional groups identified

1. Experiments (wip)

As for now, there is one experiment – turning up the heat using a slider, which causes the molecule to vibrate (animation). When the heat is on, the user cannot edit the molecule. However, there is a bug present: once the heat is turned off, the molecule still continues vibrating and the user has trouble editing the molecule. This is due to bugs in the multithreading and Timeline class I used to animate the vibration of the molecule.

1. File I/O

Users can save and load .ch4 files of their molecule into the app.

1. Dark/Light mode

Users can go to Preferences in the top MenuBar and select if they want dark or light mode, which renders the app accordingly. The theme is stored in a .txt file and read to determine if the app is currently in light or dark mode.

1. Search functionality

Users can go to Help in the top MenuBar, type in a search term, press the search icon which pops up a WebView displaying their search term entered into Google. This is useful as students can easily search for organic compounds, terms and information they do not understand or wish to learn more about.

About Program

A picture containing graphical user interface

Description automatically generatedThe About Program page can be shown by going to the top MenuBar and selecting the Help option, then selecting About Program.

It can be displayed in dark or light mode depending on which mode was set.

Features:

1. Internalisation

I implemented internalisation using a ResourceBundle. There are 4 language options: English, Chinese, Malay, French.

Model-View-Controller design

An fxml file was created for every scene and put in “View” folder. Models went in the “Model” folder and application controllers went in the “controller” folder.

Graphical user interface, text, application

Description automatically generated

OOP design

Graphical user interface

Description automatically generated with medium confidence

Atom: Abstract Superclass (abstract as not meant to be instantiated)

Subclasses: Carbon, Hydrogen, Oxygen, Nitrogen

AtomNode: Abstract Superclass, extends Circle (not meant to be instantiated)

Subclasses: CarbonNode, HydrogenNode, OxygenNode, NitrogenNode

Testing

Manual testing was conducted to test every features. Friends were asked to test the program. Bugs were identified and fixed. An example is File I/O resulting in the molecule having no atoms as the molecule was not instantiated in the File I/O method.

Reflections

A large obstacle faced was the drag and drop functionality of the atoms and a zoomable and pannable canvas. This was as there would be many atoms on the canvas to keep track off. In the end, I created a CanvasPane class for the Canvas, and the NodeGestures for drag and drop functionality and SceneGestures for the zoomable and pannable canvas.

I have learnt much more about JavaFX and different controls through this project. I find that JavaFX may not be suited for real life app development as it is difficult to manage and code overall, but it was definitely a learning experience. I learnt to appreciate the OOP design of Java as it allowed my app to be more organised and function more smoothly.

If more time was given, I would have implemented Nomenclature – naming of the molecule as a feature. This would allow the user to search up the molecule online and learn more about it. Unfortunately, I realised this was a difficult task as nomenclature for organic compounds is difficult and has many factors and functional groups to take note of, and could not complete it in time. I would also have added highlighting where the functional groups are on the molecule in order to allow students to see where it is, and take note of its position.

I found the task quite enjoyable and I liked being able to pick the topic I was going to code about.