Erik Cohen, Cassidy Correll, Anton Dimitrov, Jasper Loverude
CS361: Object-Oriented Programming

Project 1: Learning to Use JavaFX

Design Overview

Our design process in this project was to break the instructions into smaller testable chunks. This was useful not only to divide work amongst the team but also to create working code that meets our build specifications. This was the first project so part of the project was figuring out the first place to start. We copied an example and then decided how to use methods to encapsulate code blocks. We created the toolbar, menubar, and text field objects in methods in order to hide some of the created actions and styling code. This is useful because each code block has a single responsibility. A good example of this is the "Start" function. While it would have been much simpler to keep all of the code in this one method, our solution allows the start function to only handle creating and organizing the "stage". Likewise, the "createToolBar" method is only expected to create the toolbar, its buttons, and the button events.

Elegant Design Features

As previously mentioned, we tried making more cohesive code blocks by subdividing them into methods. This is more elegant because the reader is assured that all code in a section will be about the same thing. Having that, if someone else wanted to work on the same code in the future, they would have an easier time updating it - since all the GUI elements are defined in separate methods, making errors would be less likely to happen, or if they did, they would be easier to fix.

Class Structure

One of the requirements for this project is to keep all of the code in a Main.java class. While we'd ideally define the GUI elements as different Java classes, we did our best to stay within Main.java by defining separate methods for each required GUI element instead. As a result, we have created methods for the menu ("createMenuBar"), toolbar ("createToolBar"), textfield ("createTextField"), and the creation of the stage ("Start"). With the restriction of having all the code in Main.java, we made do by having the relevant GUI elements that interact with another element be passed as parameters in the method designated for the element they interact with. Hence that is why, for example, a method such as createMenuBar takes the parameters TextField textField and Button hello.

Shortcomings

An aspect of our code that is slightly inelegant is that the “createMenuBar” and “createToolBar” methods are tightly coupled with other methods/objects. This is a problem for future maintainability. We are unsure of a solution that would not use additional classes or objects as parameters.

Division of Labor

We met in Davis to work on the project with “Pair Programming” in mind. We worked together each taking turns writing code on our personal machines (and debugging the javaFX install!). The code from our project is stored on Github.