CS3035 Assignment 3: Model-View-Controller

Due: Wednesday, October 23, 11:55pm

Overview

In this assignment, you will demonstrate your ability to develop applications that use the Model-View-Controller pattern. You will develop a simple graph editor with multiple views, where the user can interactively add to and manipulate objects from the graph.

Part 1: A Basic JavaFX Graph Editor

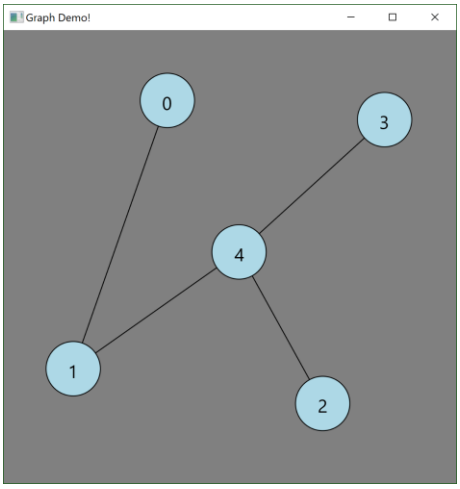
Build a simple GUI in JavaFX that allows the user to create and manipulate a graph.

Interface requirements:

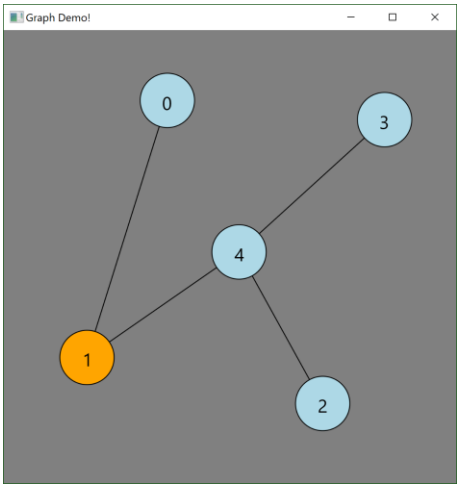
- A main panel with size 500x500; this is the area where the user will interact with the graph (see below)
- The main panel shows all graph vertices, labels, and edges

Interaction requirements:

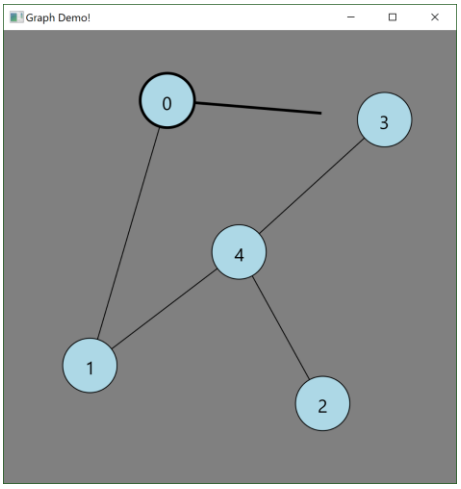
- Clicking on the background of the main panel creates a new vertex, drawn as a blue circle with an integer label
- Clicking down on an existing vertex selects that vertex (shown by drawing the vertex in orange); dragging then moves the vertex. Releasing the mouse returns the vertex to blue
- If the user Shift-Clicks a vertex, the vertex gets a thicker black border, and then when the user drags, an edge is drawn to the mouse location. If the user releases the mouse on another vertex, the edge is added to the graph (if the user releases on the background, the edge is discarded).



Main graph view



User clicks on vertex 1



User shift-clicks on vertex 0 then drags

Software requirements:

- You must implement the system using Model-View-Controller, with correct separation between these components
- Create separate classes for the Model, the View, and the Controller, following the examples given in class and available on D2L
- Build the system using the following classes:
  - **Main:** JavaFX application class
  - **GraphModel**: Class to store the model
  - **Vertex**: Represents a vertex (circle) in the graph
  - **Edge:** Represents an edge in the graph
  - **GraphView:** A custom view for drawing the graph
  - **GraphViewController:** Controller for the GraphView

Resources for part 1:

- MVCSquareDrawing.jar - MVC example from class: in Examples folder on D2L
- Slides for MVC number 7 and 8.
- GetNotifiedOfChangeInSimpleList.jar example

Part 2: Adding Interface State

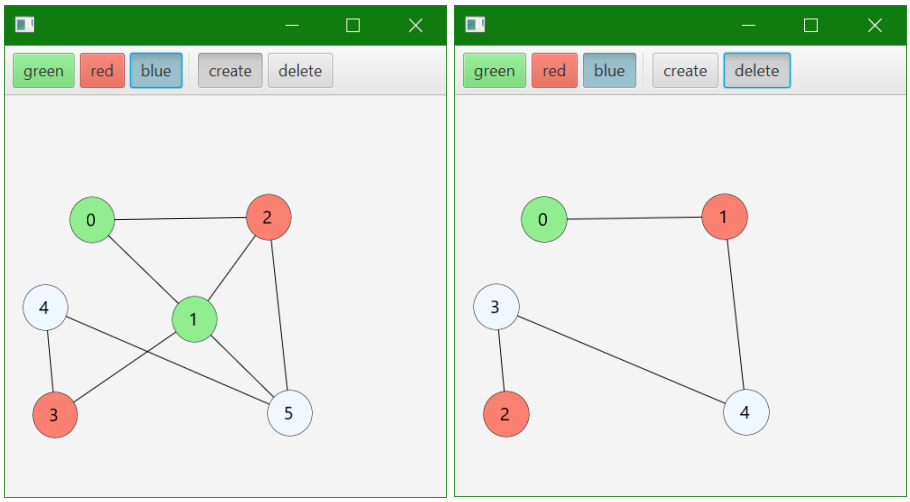
In the second part of the assignment you will extend your system from part 1 to add the ability to make vertices of different colors and to delete vertices.

Additional interface and interaction requirements:

- You will create a ToolBar in FXML, that contains two groups of ToogleButtons. One for colors to use when creating the Vertices, and one for actions to take (either to create Vertices and Edges, or to delete vertices).
    - There is a separator between both groups of buttons.
    - Buttons are colored corresponding to the resulting Vertex color.
- When in create mode the interface behaves as in Part 1, but additionally creates vertices of different colors.
- When in delete mode, the only action that can take place is that a vertex can be deleted.
    - When a vertex is deleted, other vertices are relabeled (see the images below).
    - All edges connected to the deleted vertex are also deleted (see the images below).
- To make sure that delete mode is visible while in delete mode, the cursor should change to a crosshair while over any vertex, but appear normal otherwise.

Additional software requirements:

- You must continue to implement the system using Model-View-Controller, with correct separation between these components, as best as possible.
- You can add a Color instance variable to your Vertex class, if you don't already have one.
- You should add the following classes/fxml documents:
    - **toolbar.fxml:** provides the toolbar view
    - **ToolBarController:** the controller for the toolbar.
    - **InteractionModel:** Stores the state of the interface that can queried by other classes.
- You should update your existing classes to make use of the new classes appropriately
- The GraphView and Toolbar can be added to the scene in a BorderPane



The Interface for Part 2. Deleting Vertex 1 from the left, results in the graph state on the right. Note the renumbering and deletion of edges connected to the original Vertex 1.

Resources for Part 2

- ToggleButton Tutorial: https://docs.oracle.com/javafx/2/ui_controls/toggle-button.htm
- How to do ToggleButtons/Groups in FXML: https://stackoverflow.com/questions/34010509/adding-radiomenuitem-to-togglegroup-in-fxml
- https://docs.oracle.com/javafx/2/ui_controls/toggle-button.htm
- https://docs.oracle.com/javafx/2/api/javafx/scene/Cursor.html#TEXT

This assignment is to be completed individually; each student will hand in an assignment.

What/Where to Hand In

- Two jar files, one for each part of your assignment. **Ensure that you have exported your source files in your JAR (recall that you can unzip and view the files in your jar file)**. See assignment 0 for instructions on how to create a JAR file.
- A readme.txt including your name and student number, and any notes that can assist the grader in evaluating your assignment.
- Hand in your three files (two jar files, one for each part, and one readme.txt) to the D2L.

Evaluation

Marks will be given for producing a system that meets the requirements above, and compiles and runs without errors. Note that no late assignments will be allowed, and no extensions will be given without medical reasons or pre-agreed arrangements.