CSC 335 Project 4: Mastermind GUI

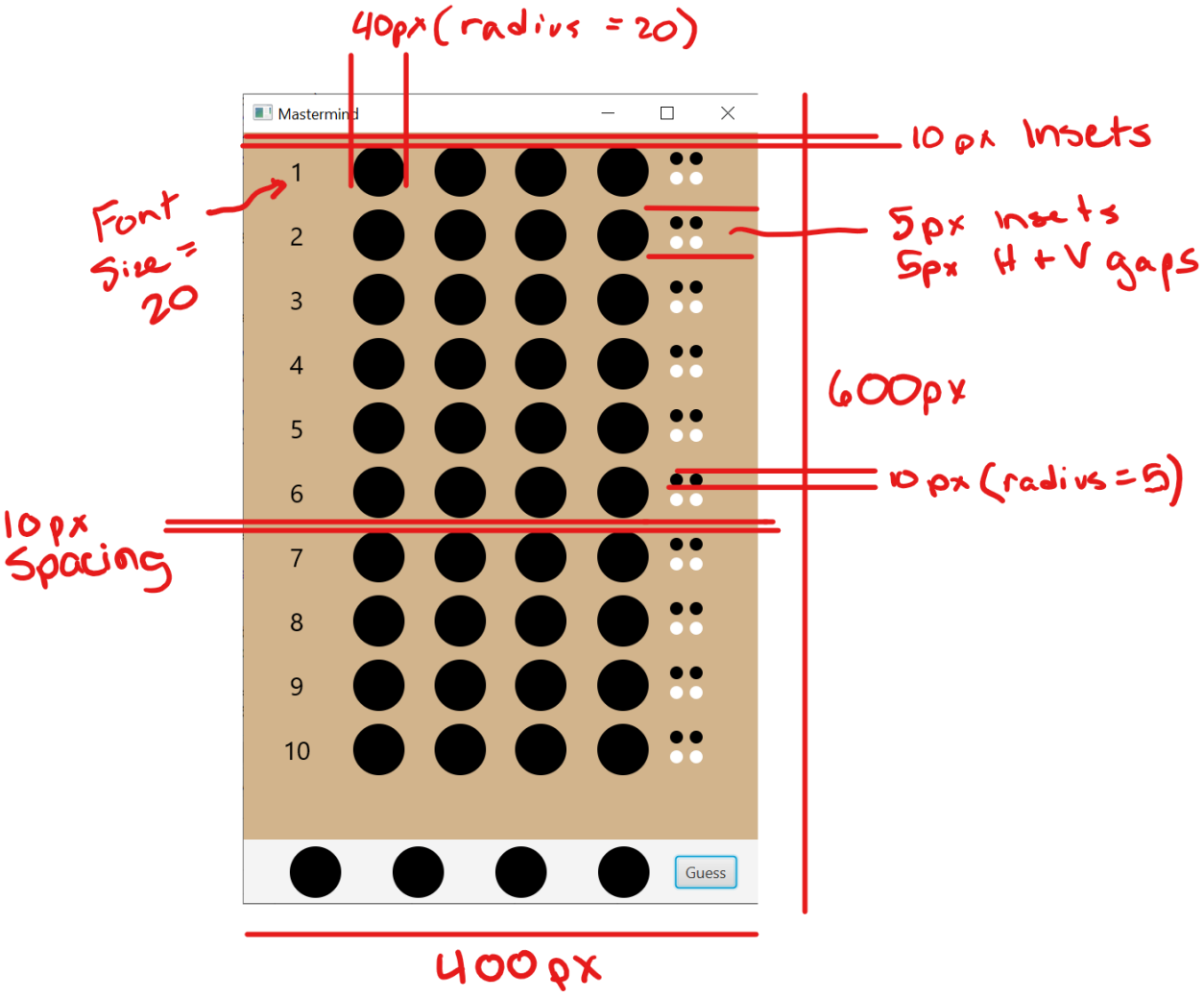
Due: Monday, March 29, 2021, by 11:59pm

GitHub Classroom Link: https://classroom.github.com/a/pnqmV3Gi

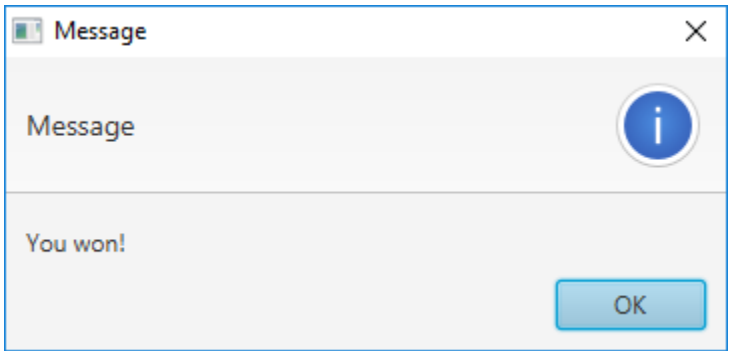
Project Description

For this project, we will use JavaFX to create GUI for our Mastermind game.

We will construct a new `MastermindGUIView` that is a `javafx.application.Application`. Here is a mockup of the layout:



When you win or lose, display a modal (`showAndWait()`) `javafx.scene.control.Alert`:



After the game is over, do not allow any further guesses.

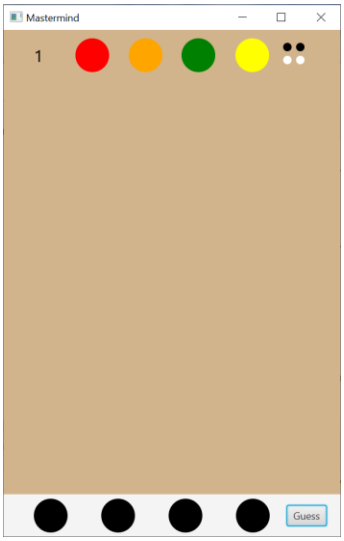
Implementation

Main Board

The main stage will be 600 pixels (px) high and 400px wide. It should be a `BorderPane` with a center and bottom. The center position will be a `VBox`, a container that stacks its contents vertically on top of each other. Set the background color to Tan and have a 10px Inset around the outside of the `VBox`, with 10px of spacing between rows.

The bottom pane will contain a `GridPane` with four `Circle` shapes, each initially Black. The fifth column will contain a `Button` with the text "Guess".

When you click guess, a new `GridPane` will be added to the center `VBox`. Inside of this `GridPane` should be a `Text` shape with the guess number in a size 20 font. Next, a copy of the guess's four colored `Circle`s should appear. The final column should contain a 4x4 `GridPane` that displays the Right Color, Right Place as the appropriate number of Black `Circle`s and the Right Color, Wrong Place as the appropriate number of White `Circle`s. There is no need to match positions in the 4x4 grid to the colored peg positions. Here is an example after one guess:



For the 4x4 feedback grid, have a 5px `Inset` around the outside and an Hgap (horizontal gap) and Vgap of 5px.

For both the bottom `GridPane` as well as the guess history `GridPane`, we want all the columns to be equally spaced. Use a `ColumnConstraint` to set the columns to 20% width for the bottom `GridPane` and 1/6=16.67% for the guess history `GridPane`. Set the horizontal alignment on the `ColumnConstraint` to center.

Event Handling

You will add a `MouseClicked` handler to each `Circle`. When you click a `Circle`, it should cycle through the colors of the game: Red, Orange, Yellow, Green, Blue, Purple, and then back to Red.

When the "Guess" Button is clicked (ActionEvent) take the current guess score it and display the results in the VBox history. Reset the four pegs back to Black and have them cycle through the colors, starting again with Red.

If the Guess button is clicked with any peg still Black, display an alert box (the same as the win box above) that informs the user they must pick four colors. Do not let them submit a guess until all four pegs are set to a color.

MVC

You are required to continue to use the MVC architectural pattern. You must have the following 5 classes:

1. `Mastermind` – This is the main class. When invoked with a command line argument of "-text", you will launch the text-oriented UI. When invoked with a command line argument 0f "-window" you'll launch the GUI view. The default will be the GUI view.
2. `MastermindGUIView` – This is the JavaFX GUI as shown above.
3. `MastermindTextView` – This is the UI that we built in project 2, moved into a separate view class.
4. `MastermindController` – This class contains all the game logic and must be shared by the textual and graphical UIs. You may not call into different controllers from the different UIs and you shouldn't change the methods from the ones we already found useful.
5. `MastermindModel` – This class contains all the game state and must be also shared between the two front ends.

Requirements

- A main class, that launches your view using:
  ```
  Application.launch(MastermindGUIView.class, args);
  ```
  for the GUI version or launches the text version, depending on the commandline argument given.
- At least the five (5) classes described above, with more if you deem them necessary.
- The GitHub repository is empty, so after you clone it, in the repository folder (somewhere like C:\Users\YOUR_USER_NAME\git\csc-335-project-4-GITHUB_USERNAME\) create a new Java

Project in eclipse (uncheck “Use default location”, browse to the repository path, and select the repository folder). Finally, import the files from the previous projects you’d like to start with.

- Do not forget to add JavaFX to the build path and run configuration as we did in lab 1.

Submission

As always, the last pushed commit prior to the due date will be graded.