Assignment 7

JavaFX Layout Panes and Event Handling

Goals:

- Write GUI programs using JavFX Layout pages.
- Write Event Handling program.

JavaFX Notes

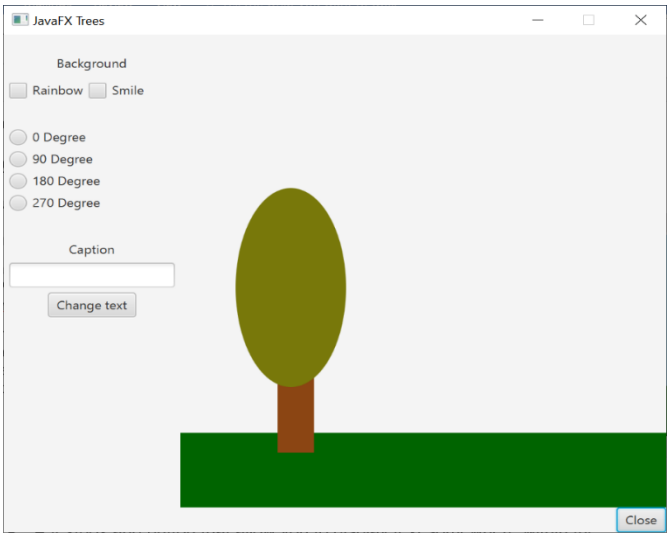
It is important to follow the methods taught in the slides for JavaFX. If you stray from them, you will likely receive significant penalties. If there is something you'd like to try and are unsure if it is allowed, contact your instructor.

You Can Use: HBox, VBox, BorderPane

Program:

You are going to build a JavaFX Application using part of your Lab 6 (or a similar alternative you've created yourself) as a starting point.

Place the background, grass and the animal from Lab 6 on a Pane that will be placed in the center area of a BorderPane.



Similar to the Message example from the lectures. Checkboxes, radio buttons, a textbox, and button will change what is displayed in the center pane. The application must include

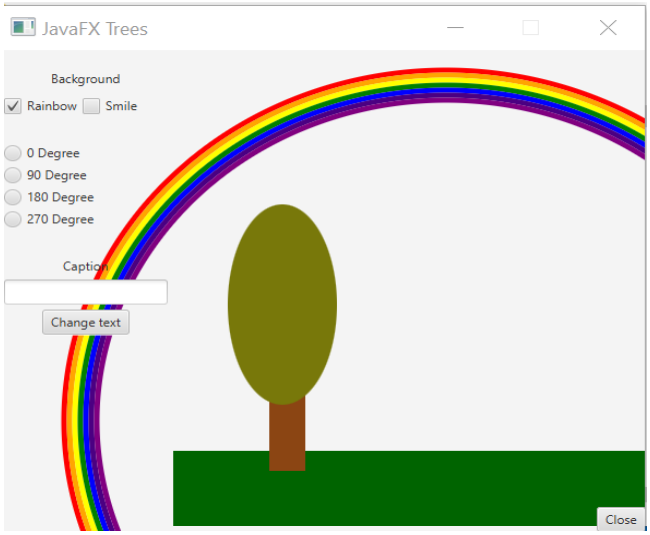
- Two or more checkboxes. When clicked they hide/display elements from the background (don't worry whether an element is truly background or not. As long as an element appears and disappears, it is ok)
- A set of four radio buttons that control the rotation of one of the elements on the screen.
- A textbox and button that allow you to display text somewhere within the graphics. The text can be positioned anywhere, but must have a color/font that makes it easy to read.
- A button that closes the window when clicked
- All graphical elements affected by the controls should involve at least two shapes.

- Elements can be hidden/visible/rotated when your Application starts, as long as the graphics match the state of the checkboxes/radio buttons. For example, if the checkbox is checked, the element should be displayed.

Hints:

- You can just rename the root Pane that you are drawing everything on. Then make a BorderPane the new root pane. Then add the original Pane to the center area of the BorderPane. Give that Pane the correct size for your graphics.
- Set up the rest of the elements in the layout. Use only HBoxes, VBoxes, and alignment/margins/padding to position everything.
- You don't need to match my example, but you should produce something similar. Your program should:
    - Use the same basic structure with similar clusters on the left side. You must have pairs of checkboxes and radio buttons.
    - The basic alignment (left/center/right) of elements should be the same
    - There should be padding/margin used to create separation from the edge of the window and between elements in the window
- The left area contains a VBox containing a Text element (background), an HBox (check boxes), a VBox (radio buttons), a Text element (Caption), and the textfield and a button.

Please Read the following Notes before you start



Solving Clipping (only if necessary)

If your graphics from Lab 6 went beyond the edge of the Pane, they would have been cut off naturally in Lab 6. But with the BorderPane layout, they will bleed into the other areas of the BorderPane like the rainbow in the image to the right. But, this is easy to solve.

You need to create a Rectangle that is positioned at 0,0 and the same dimensions as the pane for your graphics. Then use the clip method. Then everything will be cut (clipped) off at the edge of the pane.

```
treePane = new Pane();

treePane.setPrefWidth(300);

treePane.setPrefHeight(400);

Rectangle clip = new Rectangle(0,0,300,400);

treePane.setClip(clip);
```

Make Everything Work By Adding Event Handlers

Now, create event handler inner classes for the needed operations. See the following sections for help on making things hidden/visible or rotated

- **Don't create a separate event handler for every checkbox, radio button, and other element, but don't just create a single event handler called by everything. I used four event handler classes for my application.**
- **The Close button event handler just needs to call stage.close() on your stage object. Please note that an inner class can be defined inside a method body as well as inside a class body.**
- **The change text button should update a Text element you have in the pane with the graphics. If the Text element is displaying the empty string, nothing will be displayed**

Grouping Shapes Together

To simplify the code, it is best to add all of the graphical shapes of the elements being changed to a Group object

- Create a Group object. Then add all shapes that make up an element to one group.
  - In my example, I have a rainbow group, a smile group, and a tree group.
- Then instead of adding the shapes to the pane, add the shapes to the Group and add the Group to the pane.
- Now you can use the Group to more simply hide/show or rotate an element.
- You can do this with the start method or within an inner class for an element

Here is example code for the Smile in the example.

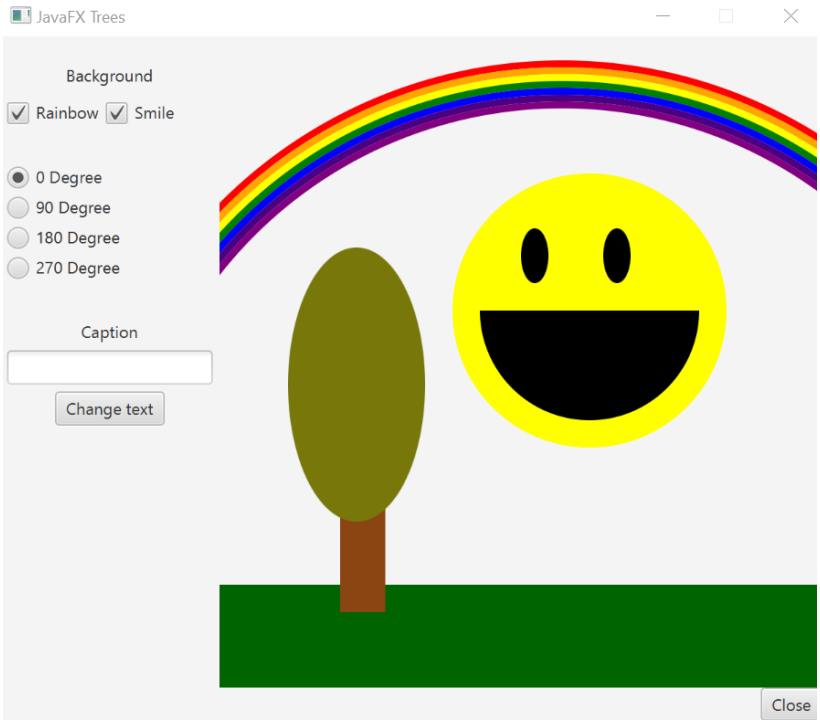
```
smile = new Group();
smile.getChildren().add(new Circle(150,200,100,Color.YELLOW));
smile.getChildren().add(new Arc(150,200,80,80,180,180));
smile.getChildren().add(new Ellipse(110,160,10,20));
smile.getChildren().add(new Ellipse(190,160,10,20));


treePane.getChildren().add(smile);
```

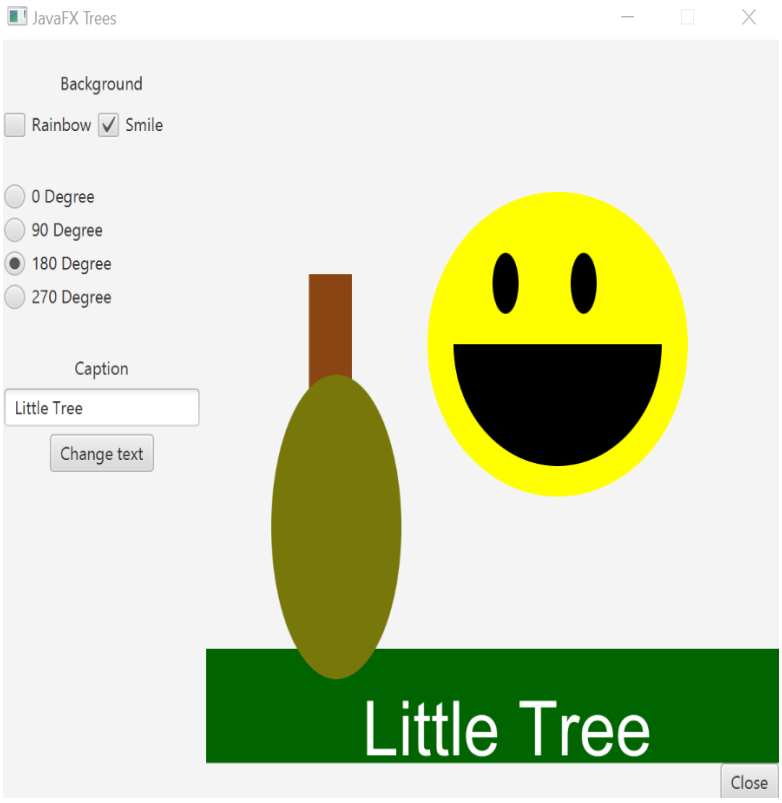
Hiding/Showing or Rotating Elements

Once you have the groups set up, this is simple.

- To make a group hidden/visible use .setVisible(boolan visible)
  - smile.setVisible(true) makes the smile appear
- To rotate a group, use setRotate(double angle)
  - tree.setRotate(0) sets it upright and tree.setRotate(90) turns it 90 degrees (sideways)
  - The group will automatically rotate around its center-point

More Example Images





Submit a single ZIP (Compressed) file to BrightSpace containing:

The source code for all the classes for this assignment.

Marking Scheme

- [35] GUI Layout
- [15] Rotate a GUI element
- [15] Make items (for example rainbow and Smile) visible and invisible.
- [15] Display Text
- [10] Terminate the program by clicking on close button.
- [5] Documentation: Javadoc for all the methods headers, purpose of the program, inline comments
- [5] Programming Style: Meaningful variable names and constants following the conventions, consistent indentation.