Cartoon

Help Session: Saturday, October 17th, 4:00 pm - 6:00 pm in Salomon DECI

Design Questions Handin: Friday, October 16th, 10:00 pm

Early Handin: Thursday, October 22th, 11:59 pm
On-Time Handin: Friday, October 23th, 10:00 pm
Late Due Handin: Sunday, October 25th, 11:59 pm

To run the demo: cs015_runDemo Cartoon
To run a snazzy demo\(^1\): cs015_runSnazzyDemo Cartoon
To install the DQ directory: cs015_install CartoonDQ
To install: cs015_install Cartoon

Silly Premise
New Concepts Covered
Assignment Specifications
Programming Tips
JavaFX
Possible Bells and Whistles
Handin Info

Silly Premise

Troy and Gabriella are soaring, flying, and are all set to star in this year’s annual fall production - and it’s driving Ryan and Sharpay crazy! They want to steal the show, but they need your help. While Gabriella and Troy are busy practicing for their upcoming roles, it’s up to you to build a sick cartoon to woo and wow the fans, and put Ryan and Sharpay back in the limelight. Stick to the status quo...and revel in the static flow of your newfound programming skillz. If your cartoon is really “breaking free,” it might even be shown to the entire class! Put Gabriella and Troy back where they belong, once and for good. This is CS15. It’s your turf.

\(^1\) Note that these demos were done in an older Java graphics package (Swing), so elements may show up differently if you try to replicate specific parts.
New Concepts Covered

- GUIs (Graphical User Interfaces)
- Graphical Containment
- Layout Management
- Complex association and sharing references
- Events and event handling
- JavaFX
- Refactoring code using helper methods

Assignment Specifications

Here's your chance to be creative with JavaFX. While we require that you implement some standard base functionality, we also want you to be creative with this. **This means you may not simply make the alien from the lecture slides.** The assignment specification is simple: create a graphical user interface that accomplishes a minor task, and uses a composite shape (a class that contains more than once instance of a javafx.scene.shape.Shape). After you have the base specs working, use your imagination, artistic ability (let's see it, you VISA majors!), and programming skills to make something that wows your friends. Think moving shapes, changing colors, and 3D rotating fractal landscapes. As with all projects, you are expected to follow CS15 design conventions. This includes use of a PaneOrganizer, constants, and helper methods as needed.

To provide some direction we require the following elements as base functionality. Before you even start thinking about extensions to this assignment, make sure you have the following:

- A composite shape, as discussed above. Some of the classes you could use are Circle, Rectangle, and Polygon. For example, you can have a composite shape of 3 Circles and a Rectangle.
- Use at least two different types of Shape, e.g. a Rectangle and a Circle, when creating your composite shape.
- A Pane that contains your composite shape
- A javafx.scene.control.Button that quits your program properly. This button will need an EventHandler that calls System.exit(0) or Platform.exit(), which exits the program.
- A javafx.animation.Timeline that animates your Cartoon in some way. Your composite shape must be part of the animation, or move in some capacity.
- A javafx.scene.control.Label with text that changes based on something in your Cartoon (for example, based on the changing location of your shape).
Make use of `javafx.scene.layout` package so that your components line up nicely within your GUI. Some useful layout containers are `BorderPane`, `VBox`, `HBox`, `FlowPane`, and `GridPane` classes.

By creating a GUI we hope that you will get a solid grasp of JavaFX. However, we don't want to squash your creativity by dictating the exact GUI layout. Nevertheless, if you are lacking inspiration, you may emulate the layout of one of the demos. In any case, be sure to explain why you arranged your GUI the way that you did in your comments.

### Programming Tips

This is the first of the projects using JavaFX. JavaFX is designed to be used for a wide variety of applications and thus has many tools that you can use for this and later projects. Check out this [tutorial](#) that Oracle provides on getting used to JavaFX. “Hello World, JavaFX Style” and “Form Design in JavaFX” will help you learn the basic layout design. Also, you can take a look at “Animated Shapes and Visual Effects” to learn how `Timeline` is used, but you do not need to get as fancy in your Cartoon project.

Because you'll be doing this in JavaFX, the stencil code we give you consists of just an `App` that takes care of the mainline. It's your job to override and fill in the `public void start(Stage stage)` method. Here you can instantiate any `Pane` or helper classes you create to manage `App`. Make sure to set a new scene and pass in your root pane.

For Cartoon, start by just getting your `Stage` to appear. Then, we suggest that you take care of all the GUI components, for example, a `Pane` to hold your labels, buttons, etc. – the sky’s the limit! Then take care of your graphics. It is a very good idea to code this assignment incrementally, making sure it compiles and does what you expect at each step.

### JavaFX

Cartoon is your first project without support code written by the TAs -- congratulations! From here on out, you'll be using lots of JavaFX code. Just like support code, all of this is written for you, so you don’t need to worry about how any of this is implemented; all you need to do is call methods as they are defined.

We highly recommend that you review the lecture slides of [Graphics I](#) and [Graphics II](#), and read through [JavaFX Shapes Documentation](#) and [CS15 JavaFX Guide](#) before you start, referring to them as you work on your cartoon. Make sure you understand the lecture slides and the [JavaFX Shapes Documentation](#) first, and refer to CS15 JavaFX Guide if you need more examples, or need some guidance in adding bells & whistles to your cartoon. The JavaFX
Guide provides all of the explanations of the classes, methods, and common pitfalls that you need to know for Cartoon. Reading these documents will save you significant amount of time coding and debugging.

Additionally, if you’re feeling ambitious, you can check out the Javadocs for loads of JavaFX Documentation -- but don’t worry if these are too complicated for now. We will teach you how to use Javadocs soon in lab, and you are not expected to use them for this project.

**Possible Bells and Whistles**

If you want to add extra credit to your assignment, take a look at the snazzy demos. Additionally, here are some possible ideas:

- Include other JavaFX elements, like `javafx.scene.control.Slider`, or something else to interact with your Cartoon in even more ways.
- Find a way to have key or mouse interactions! Talk to a TA at hours if you have questions about how to implement this.
- Add sound! All your favorite cartoons growing up at least had a good soundtrack, see if you can add one to your program.
- Add images! Read more [here](#).
- Make your Cartoon out of 3D shapes. You can read about how to use them [here](#).
- Make fancy transitions! There are many transition animation effects provided in JavaFX such as `FadeTransition`, `ParallelTransition`, `PathTransition`, `RotateTransition` and so on. Read more [here](#).

Remember that you should make sure that you have a fully functional program before working on extra credit. If you have any other ideas for extensions, you may want to talk to a TA before attempting to implement it. Remember, from the Course Missive:

> “[Extra credit is] to be done only after the original assignment is completed in full. You must include full documentation of your program’s additional functionality in the program header. In other words, if everything is working right, extensions will give your grade a boost, but if you did not do part of the assignment, the extensions will be worth nothing. This also means that extra credit will not be assessed for your project if you hand in a project late. This is not arbitrary; most extensions, while requiring extra work, do not teach new concepts.”

**Handin Info**

To hand in Cartoon, type the following command into a terminal:

```
cs015_handin Cartoon
```

As usual, your program must be fully commented. This means we are expecting to see header comments for every class and on any methods you write, but you do not need to comment
saying something like “//here I am declaring a private instance variable of type BorderPane” – we trust you know what you're doing. See the Style Guide for more information.