Kristina Frye, Konstantin Macarenco, Tyler Poole

CS 420/520

June 8, 2015

Final Project

1 Architecture

2 Challenges

The biggest challenge of this project was to get the interface between the javascript createjs library and the Grace compiler working. In order to get the graphics working, we examined the existing GraphicsDraw object to see how it worked, figured out how the calls would work in straight javascript, and then used the Javascript console in the Chrome browser to step through the code and figure everything worked together. One of the complications is that graphics are usually drawn in the main browser window. However, the goal in this case was to open a secondary "pop-up" window and draw the graphics in that window. Since this is not the usual way that Javascript is typically used, we ended up having to use a few workarounds.

One of these workarounds involves the event listener for the click handler. The createjs library normally allows the developer to assign click handlers to the individual shape objects that are created. So, if you create a circle, you can assign a click handler directly to the circle. This handler only is activated when the circle is clicked. However, because we are using the pop-up window, this turned out not to be possible. This appears to either be a bug or a missing feature in createjs, but the event listeners on individual shapes are not working when used with the pop-up window. However, the event handler for the "stage," which is the createjs specialized version of the HTML5 canvas, does work. So we assign all event handlers to the stage and check the mouse position in order to determine which object was clicked. This is not as clean or efficient as adding the event handler directly to the shape, but it works. Another possibility in the future would be to designate part of the Grace compiler webpage for the graphics display. This would get rid of the "pop-up" issue and would allow a much greater variety in mouse handling.

Another challenge was adding sound capability. We were hoping that we could load the sound files when the graphics library was loaded. That way the user who is not working with the graphics library would not have to load the sound fields into the browser every time the Grace compiler was used. However, this didn't turn out to work very well. When loaded with the graphics library, the sound would play the first time the program loaded. However, if we tried to run the program again, the sounds would not play unless we reloaded the entire Grace browser webpage. Therefore we added the sound load into the main Grace index.html file, choosing only a small sampling of sounds with a relatively small file size.

3 Achievements

We are proud of the resulting user interface of the createGraphics library. Although it doesn't yet have a large number of features, we have provided a good framework that could easily be built upon, and the existing user interface should be easy for the beginning programmer to work with. We have made it as "Gracelike" as possible and provided default settings whenever they were needed in order to make it quick and simple to use. We think it's going to be a good improvement on the existing graphics library once some additional functionality is added.

To include in the writeup:

The overall architecture of the project: the objects you created and their responsibilities. This is intended only as an overview; the details should be in the code. (2 pages maximum). Include diagrams if appropriate.

The challenges that you faced and how you overcame them. (1 page maximum)

Anything that you are especially proud of (or embarrassed about). (1 page maximum)

If you did the project as part of a team, 1-4 above are team deliverables.

I also want an additional item from each individual: a summary of what you contributed to the team effort and a description of how well the team worked together (1 page maximum).