CSSE1001

23/10/2009

Assignment 3 – Reflection

Name: Justin Mancinelli

Student Number: 42094353

Project Title: Antioch

The Process

Antioch took a lot of my time. I didn't realize how time consuming it was going to be partially because I didn't think I would use Tkinter, partially because, partially because I didn't know I was going to have so many projects simultaneously. Truthfully, though, it was Tkinter that made this take so much longer than I had hoped

Most of the coding wasn't difficult, in fact, I was able to get all of the libraries, the canvas interaction, and loading and saving files finished in one week. However, the GUI coding was tedious. Tkinter comes with many basic widgets which implement basic interaction. I needed to create many of my own widgets and interactions based on those native to Tkinter.

I started with the top-level menu since I had described it in detail in the initial design document. After some trial and error, I converged upon a data-structure I deemed most suitable. The trials included testing the parsing methods with the structure and the errors included menu items not showing, menu items not cascading, and menu items which wouldn't configure with the correct attributes.

The next thing I worked on was the toolbar. Some very strange things happened during this stage when trying to make the toolbars draggable. I'll explain more in the next section but it had to do with Tkinter's geometry/drawing manager.

I was afraid that the geometry manager would freak out about resizing the sidebar just as it did with dragging toolbars but I tried anyway. I tried to code them in such a way that would limit the number of Tkinter methods I called because I was beginning to comprehend how slow Tkinter really is.

The canvas (workspace) was saved as the last basic GUI part. Assignment 2 gave me enough experience with the canvas to make this part rather simple. Once the grid-lines were placed I began to port Logisim libraries and components for use with

Antioch. I coded them in such a way that should allow for the program to be extended with a simulator rather simply (though Tkinter may make it a lot harder).

Once I got all the GUI widgets ready for display and interaction, I worked on loading and saving Logisim ".circ" files. Logisim saves its circuits as valid XML and Python comes with several XML parsers—I chose the fastest and most basic one.

What went wrong

I wanted toolbars to be draggable within their container and automatically rearrange themselves or be dragged outside the container and become a top level window. Tkinter's bindings and backend didn't allow for this. I created very simple methods to allow dragging and tested them with simply changing the color of the toolbar, but sometimes the geometry manager would loose track of them and they would change the wrong color or completely disappear.

I wanted the grid lines on the workspace to be dotted but after playing around with the –dash and –stipple keywords, I couldn't get that to happen correctly. Making the gridlines low contrast was a good compromise.

Tkinter keeps track of objects on the canvas in a list by order of creation. It does have methods to deal with bounding boxes and coordinates of drawn objects but has no way of grouping objects except through tags and thus there's no way to know whether a group of objects are colliding with another group of objects except by keeping track of the right tags and seeing if they appear in the same bounding box. Since every component is a group of canvas objects, I had to write my own methods for selecting groups of objects, moving groups of selected objects, and determining whether a group of objects are physically within in the same area as another group of objects. I did not enjoy discovering the quirks of the Tkinter canvas class.

What went right

It was a good decision to start with the menu. Tkinter has a very easy API for working with menus and since I worked out all the menu items in the initial design document, I simply needed to code it up.

I settled for a different interaction for the toolbars than I had originally intended. Once I fixed that, all the rest of the toolbar code progressed smoothly.

The sidebar actually behaves rather well. You can clearly see the slowness of Tkinter because the colours of the sidebar do not refresh instantly (you can see them wave). I'm happy with the sidebar even though I didn't have time to make it useful.

Allowing for the saving and loading of Logisim circuit files was actually quite painless.

Once I understood how to use the XML parser, I could use string methods to convert the parsed strings from the Logism circuit file into Python code compatible with Antioch that could be evaluated with Python's "eval" method. Saving a file was even easier because of choosing suitable tags on initiation of canvas objects. The worst thing about saving a file was the tedium of making sure those tags got placed into valid XML for use by Logisim.

Bugs, Time Management, Conclusion

Bugs count as strange things that happen to parts of the application which have been implemented. Bugs don't include missing functionality which hasn't been coded. All the bugs I've noticed in Antioch have to do with the canvas.

The method I'm using to determine if a component is or should be selected works well as long as the bounding boxes are not touching. If the bounding boxes touch or overlap then more than just the component you want to select are selected. That highlights another bug. I have code which disallows components to be placed on top of one another but it's based on the position of the mouse cursor. It's conceivable that the mouse cursor is not within the bounding box of another component when the dragged component is dropped and yet still have the two bounding boxes overlap. The other big bug is that my component designs do not correspond exactly to Logisim's which means that doing a direct copy of coordinates leads to messy results when loading from one program to another.

All of the above bugs can be fixed given more time. I usually have very good time management and I was doing the best I could to do this project along with 3 others including homework assignments. In retrospect, I think I should have started on more difficult aspects of the project, though as mentioned above, I didn't expect Tkinter to be so difficult to work with. If I changed to PyQt or some other GUI manager at the time Tkinter became difficult, I would have gotten even less finished.

I believe I used my time wisely. I finished all my other projects and other assignments (which I think went rather well) and I was able to get a lot of Antioch finished. A circuit design program without simulation is really more of a GUI programming task anyway, so I think I've done well considering.