**Final Graphics Report**

Dave J and Sean H

Major developments/breakthroughs(reference specific code please):

I have learned how to use the GridBagLayout layout manager. I first used this layout manager in “Version6.” I really like the freedom this layout manager brings to the table, as it uses a coordinate system to organize the GUI components.

GridBagConstraints c1 = new GridBagConstraints();

c1.fill = GridBagConstraints.HORIZONTAL;

c1.gridx = 0;

c1.gridy = 0;

c1.gridwidth = 3;

add(tfMain, c1);

In the above section of code (from “Version6”), I learned that a GridBagConstraint can be a row or a column, depending on if you set it to be horizontal or vertical. Here, I set it to be horizontal (c1.fill = GridBagConstraints.HORIZONTAL;) I also learned that (0,0) is at the top left of the screen, so here, tfMain would be in the top left. It is also useful to be able to set the grid size of a particular component (here, I set tfMain to be 3 grid units wide: c1.gridwidth = 3;).

Major Challenges/setbacks( reference specific code please):

My biggest challenge was in “Version8,” when the global variables in my program were not changing when I programed them to do so, and when I was receiving errors from a suspected parsing issue. The parsing errors highlighted line 175 for me as the problem line. The line was:

dWorkingNum1 = Double.parseDouble(sWorkingNum1);

This line takes sWorkingNum1, which accumulates all the values gotten with the getActionCommand function, and is supposed to convert it to a double value. However, this was not the case, as it gave me a errors. Despite trying other methods of parsing such as:

Version9.dWorkingNum1 = (Double.valueOf(Version9.sWorkingNum1)).doubleValue(); ,

I had no luck, and it still gave me the errors. I also tried researching global variables, and stumbled across a way of calling variables (for example, Version9.dWorkingNum1). However, this too did not help. I resorted to asking Gaurab for help and even he did not know what was wrong with it. At that point I was getting frustrated and called it a day for the time being. The next day, however, I ran this program again, and discovered that I was no longer getting all those errors, but the variables were simply not changing from what they were defined as. I tried using different global variables again, but I decided to go back to the last working version and try again. This time it worked, likely because I took a different approach to programming the calculator. That new approach was mainly giving the “equals” button and the “CE” button their own action listeners, names “Equals” and “Ce.”. This was mainly to reduce the amount of nesting I had to do, but it seems to have also fixed my problems.

Any modifications to your planned steps;

I followed you advice and changed the GUI to be more like a real calculator, as opposed to having a combo box where the user chooses an operation, as well as text fields for the user to input numbers using the keyboard. This proved to be a bit more challenging than I first thought with the changes, but it was interesting to see the phantom errors with the parsing of the getActionCommand.