**Work in Progress Report #1.5**

**Major developments/breakthroughs(reference specific code please):**

I created a counter in PanQuest that removed buttons and labels based on what the counter was set at when the buttons was pressed the time previous. For example if you chose Functions then the counter would become 2 so the next time you choose a unit it reads that the counter is 2 and gets rid of all buttons associated with Functions.

**Major Challenges/setbacks( reference specific code please):**

The biggest setback I’m having is once you choose a unit having it send to a new class once you choose a question. When you choose a question in PanQuest and click its button I want it to use an action listener to send it to a different class where it would then run something in the display depending on what you chose but when I try to send the information it won’t work since it’s not in the public class. I can’t have it in the main class because it’s using information passed from a different class already. I’m working on having the information from the different class be turned into a variable that I could then put into the public class.

**Any modifications to your specifications/release schedule:**

I’m not sure yet if I will have to delay my release schedule but I don’t think I will have to depending on how long it takes to figure out the problems with passing between classes.

**Description of your scratch/test program:**

**Describe the generic concept you needed to test out:**

The concept I worked on this time was having a counter that removed buttons and labels based on what was chosen previously.

**Source any web site/book that helped you with that concept:**

N/A

**Describe the code and the lesson that you learned from it**:

The code has it so that when you choose a unit it makes the counter either 1 or 2 (depending on which unit) and then the next time you choose a unit before creating new buttons it picks up on what the counter is and then removes all of the buttons and labels accordingly. This was useful because it showed me that you can leave trails for you code to pick up when you come back to that point again later.