

MathNotes: Documentation

CS 6456: Project Milestone 2

Team Members

Elisabeth Hutzel
Madhura Bhave

Description of Work

We started with a blank Canvas and implemented drawing with single touch. This included previewing while dragging, undo and redo functions, and erasing. Then we implemented the creation of shapes: circle, square, triangle and the diagrams graph axes and curves. This we created in a modular way such that any additional shapes or diagrams can be easily added, allowing us to focus on the rest of the system without worrying about which shapes you can actually generate.

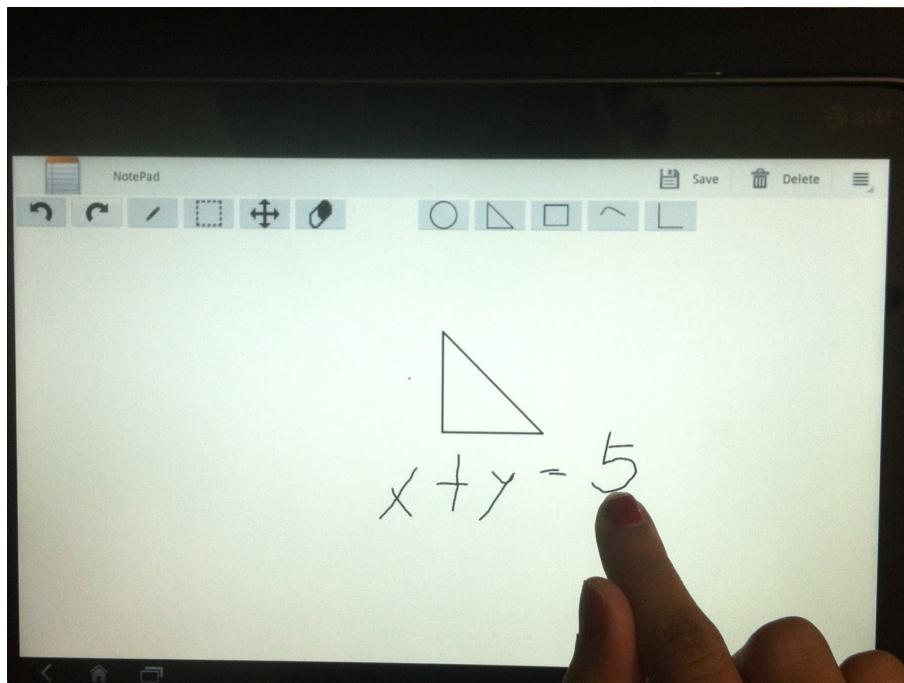


Figure X. Freehand writing.

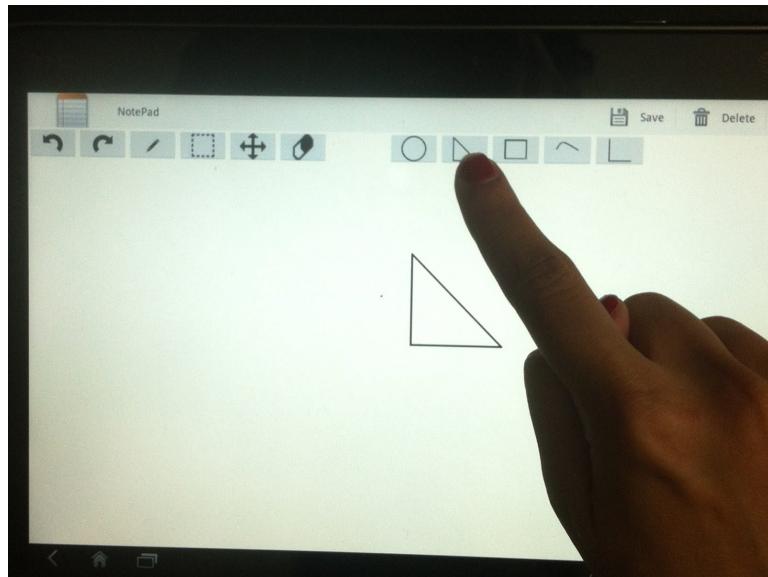


Figure X. Drawing Shapes.

Once we had the initial canvas system set up, we focused on implementing the multitouch part of the system. This included selecting an area of the canvas to manipulate using a finger drag from top left corner to the bottom right corner of the selected area. Then we made this selection modifiable through one and two finger integration including panning, scaling, and rotating. There are built-in functions for panning and scaling gesture detection in Android already, however for rotation gesture detection we looked to a MultiTouchController library (see 3rd Party Libraries). This took a major part of the work, as getting the selection to properly select, remove the selected area, and paste it into the final destination was very difficult to get working with the multitouch functionality.

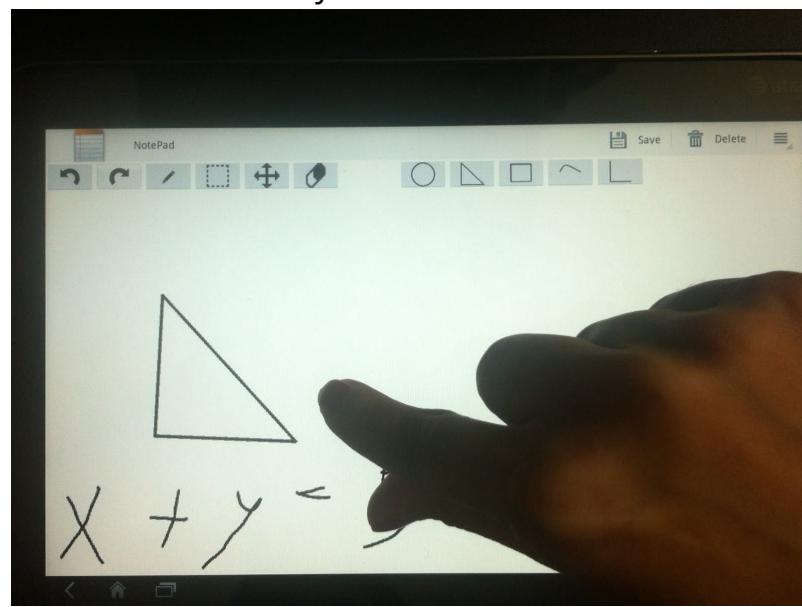
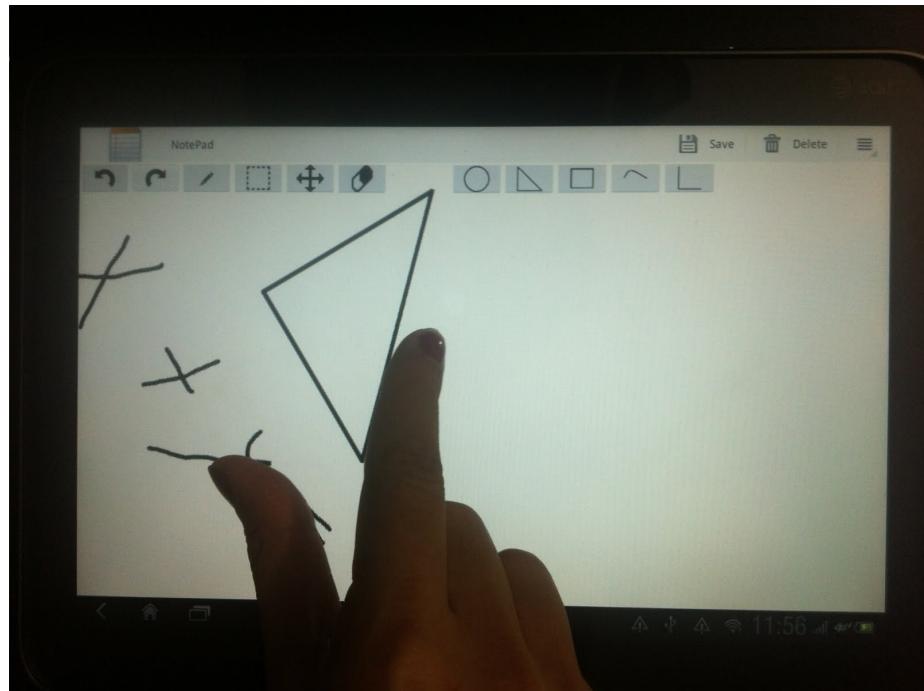


Figure X. Zooming.



Figures X. Rotating.

Last, we wanted to create a polished notebook like feeling so we created a starting menu with a list of the pages created in the main drawing interface, each one with the ability to open, edit, and save changes. Most of the work here was saving the Canvas and populating the list.

There were some other things we wanted to implement, but we wanted to focus on the UI and multitouch part of MathNotes as that is the main part of the software and what we learned about in the class. We chose to not implement the SiGeR recognizer as that is not new UI contribution at all and focused instead on less known interaction techniques. We also removed the ability to type input, as we felt this detracted from the multitouch interaction, is difficult to type on the smaller Android device virtual keyboards, and you can just write everything with your finger or pen anyways. This also makes integration between objects and text much easier, allowing the user to simply draw all over the object and then select the text and the object to resize, move, or rotate them together rather than requiring the user to choose text input, type the text and then having the text floating apart from the object. Since we did not implement the SiGeR recognizer, we also did not implement the advanced mathematical expressions, as this felt useless without the ability to do actual math on the expressions. Instead the user can simply draw everything, rather than having to open a menu with the mathematical symbols and insert, position, and resize them.

Development Environment

Our development environment was Android (<http://developer.android.com>) SDK API level 12. We used the Android Emulator for most testing purposes as we did not always have access to a physical Android device; this significantly slowed down development as the emulator is extremely slow and does not have multitouch capabilities. The tablet we used for physical testing and demoing was an HTC Jetstream (<http://www.htc.com/us/support/htc-jetstream-att/>) running Honeycomb 3.1. We also used the Eclipse IDE with built-in ADT (Android Development Tools).

An Android app starts with an Activity class and a manifest file. The manifest file describes the fundamental characteristics of the app and defines each of its components. We have two main Activity classes, the NotesList and the DrawingActivity. The NotesList is just a list of pages created by the user where each page is created in the DrawingActivity.

3rd Party Libraries:

To assist in creating a multitouch interaction for rotating objects, we found this great library: <http://www.2dwarfs.com/tutorials/android-multitouch-controller>. Android has built in features for panning and scaling using the typical two finger pinch, however the MultiTouchController created by 2dwarfs integrated those functions into their library, so we just used all of the multitouch features from the MultiTouchController library. All the code in MultiTouchController was not written by us.

We also looked at the Android sample code (<http://developer.android.com/tools/samples/index.html>) to help us get started, since neither of us had programmed for Android, or any other multi touch software, before. We started out using the NotePad example to set up the notebook setting we wanted for MathNotes, however the NotePad only had the simplest text processing, so we ended up only using the NotesList file of the sample code as the start for the Activity and we launch our DrawingActivity from the New button, rather than the NoteEditor.

All the other code was written by ourselves or generated by the Eclipse ADTs (such as the xml files, although we modified some of those by hand as well).