I learned a lot doing this project quite honestly, it was a challenge. I came across a lot of issues in the process of doing this, and I would say I could've managed my time much better. For example, I never got around to adding poly-lines, ellipse, and curves as shapes. Some of the more challenging aspects of this assignment was well for one, coming up with a class hierarchy that was flexible enough to handle most operations smoothly on the shapes, I believe I achieved this to some extent, my mistake was in assuming that all the shapes were going to be vertex based and so my abstract shape class only did transformations at the vertex level and it was up to the inheriting classes to define an intuitive interface for specifying coordinates and dimensions for a given shape then translating the shape into an array of vertices so that the shape class can then take over. Of course, this would be all well and good if I had defined all my shapes in such a manner that they were all vertex based, but given that the canvas API provided a method to draw circles, all I needed was a position and a radius to draw one, so generating a vertex array of all the points on the circle than transforming each one was rather pointless, no puns intended, same could be said for the ellipse. It probably would've been also useful to have a getExtentX, getExtentY, drawBoundingBox methods defined in the shape class, because as I've come to find later, we sometimes want to draw a bounding box around our shapes, or use the dimensions of the bounding box to do calculations, for example with group operations. I did get around to saving the canvas, but in a limited manner, the user is able to save and load from localstorage, I was at some point trying to use NodeJS to enable me to read/write from files by running a local server, but I never got around to doing so. I did however implement operations on groups as a whole, as an extra credit feature, although it's a bit buggy. If you click on an empty space on the canvas and drag, you can see the beginnings of a selection box. The implementation of the ShapeGroup was very much facilitated by the Composite Pattern. Not sure if the menu selection counts as an extra credit feature but the way

you draw and drop shapes into the canvas behaves much like gliffy. The user is also able to clear the screen of shapes, I would not count this as extra credit though. All in all, there was a lot to be learned here, not just about the mathematics involved in Computer Graphics, but also program design in general.