

For project 6 I was able to find a piece of software called “FontForge” (<http://fontforge.sourceforge.net/>), which is a tool for editing True Type fonts (.ttf files). Using this tool, I opened the “Times New Roman” font located on my Mac (/Library/Fonts/Times New Roman.ttf). Once I had the font loaded in the tool, I was presented with a wealth of information.

However, the steps required to get the font into a format where I could easily use it in my Matlab project were not trivial. The tool provides a “properties” window that will let you walk through the Bezier points one by one, but the sheer number of points made this too painful to attempt. After some experimentation, I found that the tool would export individual letters from the font to PostScript files. Using the information in the “Reality Check” on page 188 of the text, I was able to write a crude translator in Perl that would translate the PostScript files to Matlab function files (one for each letter: “g”, “m”, and “u”).

Because I was able to use FontForge, I took on the added assignment of exploring a transform which would slant the letters. I hypothesized that for some constant,  $c$ , the transform:

$$\begin{aligned}x &\mapsto x + y*c \\ y &\mapsto y\end{aligned}$$

where:

$x, y$  are the vectors containing the  $x$  and  $y$  coordinates of the Bezier control and end points.

would slant the letters something similar to what you would expect in an italic font. I was pleased with the results (they looked as I had expected). While it may seem obvious that transforming the Bezier endpoints in such a manner would yield the results just described, I don't think it's as obvious that transforming the control points will yield these results. My experiment seems to indicate that the transform works for control and end points.

Following are descriptions for each of the attachments to this document.

- **convps**: This is the Perl script that translates PostScript files into Matlab m files.
- **gen**: This is a shell script that calls convps for each letter in “gmu”. I included as an example of how to use convps.
- **g.m, m.m, u.m**: These are the m files generated by convps. They include the Bezier points for the letters.
- **bezierPlot.m**: This is the m file containing the code we were given to start the project. I modified it to calculate two widths for each character; the width with no scaling and the width with scaling. Depending on how you wish for spacing between the characters behave when scaling occurs, you would choose one of these over the other. In my plots, I used the width before scaling.

- **plotGMU.m**: This file calls bezierPlot for each of the letters and spaces them apart from each other. It is designed to be called in one of two ways. It can be called from another m file (command line style), or it can be called from a Matlab GUI callback function. In the latter case, it is heavily tied to the GUI I designed. This is because names of the widgets whose values it is looking for are hardcoded into the plotGMU.m file.
- **project6.m**: This file is supplied to run the project from the command line.

Also attached to this document are four images. They are (in order):

- **“gmu”**: The letters “gmu” plotted with no Bezier points and no slanting.
- **“gmu” With Points**
- **“gmu” With Points, Slanted Left**
- **“gmu” With Points, Slanted Right**

Missing from this document are two files, project6gui.fig and project6gui.m. The former of these two files is the GUI definition file generated by GUIDE, Matlab's GUI creation tool. The latter is the m file generated by GUIDE that contains all of the initialization and callback functions for the GUI. Neither of these files is very interesting: the former is a binary file, the latter is very verbose and most of it is generated. It wasn't worth printing it out for the few lines in it that I changed. However, if you have a soft copy distribution of this project, they will both be included because they are required to start up the GUI!

If you do have a soft copy of this project, go into the project directory and type project6gui.m (or open the file of the same name in Matlab and click the “run” button). If you're running a later version of Matlab, you should see the GUI pop up. Enjoy!