Assignment 2b - Due Tuesday February 21

Create the data structure for the graphics state. It will contain a structure that holds 3D point data and a number of matrices in addition to the relevant data about raster dimensions, current mode, etc. Note that your graphics state only need be as complicated as is required to properly render **simple.rib**. We will add to it as necessary when our RIB files get more complicated.

Write the appropriate routines for creating the graphics state and initializing it to the proper default values. Note that since Begin and End are not defined in RIB (meaning, you don't have to worry about destroying old graphics states and creating new ones beyond the single state) you can just have a single static graphics state variable if you like.

Write the appropriate virtual functions for WorldBegin, Points, and WorldEnd where necessary in Gabe's parsing class.

Finally, put it all together. Write a program that takes **simple.rib** as input, properly tweaks the graphics state, makes an image, and writes it out as a TIFF file using Gabe's TIFF wrapper function **WriteTiff**.

Here's the latest (working) version of **simple.rib** for reference:

```
WorldBegin
Points "P" [0 0 1]
WorldEnd
```

Each of the three commands in **simple.rib** has a well-documented impact on the graphics state (see the spec for details). Use these definitions to guide your implementation of both the state and the functions. For instance, `WorldBegin` sets the world-to-camera transformation to the current transform (which defaults to I) then resets the current transform to I. `Points` obviously adds geometry to the state, defined in an object coordinate system, but also remembers the current transformation because the points must be taken from object to world coordinates on their way down the display pipeline. Read through the spec as you implement each of these commands and add what you need to the graphics state to make the commands do what they're supposed to do.

Hand in your code for all of this stuff, along with your final output TIFF image.