Assignment 2: Rasterization Deadline: **March 9th, 2021 at 23:59**

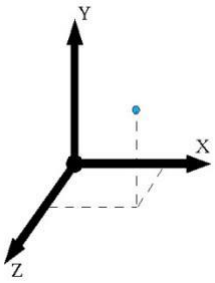
In this assignment, you will implement the Scan Line Algorithm for the Gz library. With your provided API functions, the main application will read series of triangles from the text file **`Tris.txt`**.

All the data you need for this assignment is put in the zip file hw2.zip. Check the file **`handout.pdf`** for some overview of the assignment. Besides some files already provided in assignment 1, there are other files have been updated, please check them:

File	Description	Type
Gz.h Gz.cpp	Updated to support Gz::begin(GZ_TRIANGLES)	Incomplete
Tris.txt	An input text file contains the list of triangles. You can figure out the format by reading the source code in file main.cpp or check the description bellow.	Data files
TeaPot.bmp	The sample bmp-format results. Note that you are supposed to generate a result looks like this file, but not exact pixel-by-pixel.	

Notes:

1. Format of the file **`Tris.txt`**
   - First line contains the number of triangles
   - A group of 3 consecutive lines contains information about 3 vertices of a triangle. Two groups are separated by a blank line.
   - Each line in a group contains information about vertex, includes coordinates (x y z - 3 real numbers), and color (RGBA - 4 real numbers).
2. Coordinate system convention
   In assignment 1, you were not required to follow any convention about the coordinate system. However, from assignment 2, you must use the coordinate system described in the figure beside.



Requirements:

1. Do the assignment independently.
2. You must submit all your source code, project files (MSVC project or Makefile), your results and your report. TA may test your implementation by changing some options, changing the source code of the main program, or changing the data file.

3. You need to write a detail report(50% points of the assignment, pdf format), you should state the assignment problem, explain the algorithm or method you use, explain details of implementation, discuss your results and etc.
4. Save your results as images.
5. Upload every necessary file to Github
6. In your Github readme file, put your name and student ID there, include coding environment and compiling method (command) if necessary.
7. You can only use the libraries we provided.
8. You will lose points if violate any requirement above.