CS 350 Project Specifications v0.1

Part 1: Glyph Loader

The view component of the architecture renders an aircraft carrier as a three-dimensional wireframe drawing. The definition of the wireframe resides in a text file. Your task is to build the class that loads and interprets the contents of this layout file, builds the corresponding data structure, and makes it available to my architecture as a `LayoutBundle`.

The public specifications are as follows:

The class is called `GlyphLoader` in package `acg.architecture.view.glyph.loader`.

`GlyphLoader(String filename)`
The constructor takes the fully qualified filename of the definition file.

`LayoutBundle load() throws IOException, InvalidLayoutException`
Loads and processes the layout file and returns a layout bundle containing the edges and circles.

The provided jar file (`projectV0_1.jar` or newer) contains the following support classes:

`EntryCircle`	Defines a circle entry as a vertex, radius, and color on an arbitrary coordinate system.
`EntryColor`	Defines a color entry.
`EntryEdge`	Defines an edge entry as a line between two vertices and a color.
`EntryVertex`	Defines a vertex as an (*x*,*y*,*z*) point on an arbitrary coordinate system.
`LayoutBundle`	Defines a collection of edge lists and circles.

See the Javadoc for more details.

The text file is in comma-delimited form as follows:

- Color entries are defined by the character **`c`** in the first field, the unique positive integer color index in the second, and the hexadecimal RGB color code directly preceded with the character **`#`** in the third; e.g., **`c,1,#abcdef`** defines color **`1`** as **`abcdef`**.

- Vertex entries are defined by the character **`v`** in the first field, the unique positive integer vertex index in the second, and signed *x*, *y*, *z* double values in the remaining fields; e.g., **`v,1,60.8,523.5,-8`** defines vertex **`1`** as **`(60.8,523.5,-8)`**.

- Circle entries are defined by the character **`o`** in the first field, the vertex index in the second, the color index in the third, and the radius in the fourth; e.g., **`o,47,5,10`** defines a circle centered at vertex **`47`** with color **`5`** and radius **`10`**.

- Edge entries are defined by the character **`e`** in the first field, the start or end vertex index in the second, and the color index in the third; e.g., **`e,1,4`**. A subsequent edge entry without an intervening blank line continues from the previous vertex to it as part of an edge list; e.g., **`e,2,5`** would create an edge from vertex **`1`** to vertex **`2`** with color **`5`**. A blank line or end of file ends this list and starts a new one, if there are more entries.

Color and vertex entries may appear in any order before usage. Circle entries and edge lists may appear in any order.

Comments begin with the character **`;`**. They may appear anywhere and continue to the end of the line. Blank lines are irrelevant except in edge entries. Trailing commas and whitespace are irrelevant everywhere.

Exception messages need not be especially informative, but `InvalidLayoutException` must indicate the line number of the error in the layout file, if possible.

Use only standard Java, no external tools, libraries, etc.