

Due Tuesday September 20 at 6:00 am.

## Assignment 2—Hello 3-D world

Extend your application to allow the user to navigate a 3-D scene interactively. Concretely:

- Develop a planar pinhole camera (PPC) class
  - o Stores vectors  $a$ ,  $b$ ,  $c$ , and  $C$ , as well as scalars  $h$  and  $yon$
  - o Also stores inverse of camera matrix for efficient projection (should be updated every time the camera changes)
  - o Constructor from resolution and horizontal field of view
  - o 6 degrees of freedom (dof) navigation (3 translations and 3 rotations)
  - o Zoom, change of resolution
  - o Linear interpolation of PPCs
  - o Projection of 3-D points
  - o Visualize method that draws the PPC frustum given a desired view PPC and a frame buffer; show a few pixel rows and columns on hither quad; parameters should be line color and spacing of pixel rows and columns
- Develop a triangle mesh class
  - o Stores vertices (coordinates and colors) and connectivity data defining triangles, as well as number of vertices and number of triangles
  - o Load from file (see accompanying code and geometry files)
  - o Render method that draws the mesh given a desired view PPC and a frame buffer;
    - correct visibility by z-buffering, enhance your frame buffer class with a depth channel
    - point-based mode: vertices drawn as 2-D dots of size  $k \times k$  ( $k$  is a parameter)
    - wireframe mode: projected triangle edges are drawn as lines, color is interpolated between vertex colors
- Develop a path class
  - o Stores keyframe PPCs and number of frames between each consecutive key frames
  - o Loads and saves path from text file
  - o Render method that draws the centers of projection of the keyframes as big points connected by line segments, given a desired view PPC and a frame buffer.
- User interface should allow the user to
  - o Navigate the desired view (6 dof)
  - o Change field of view of desired view
  - o Change rotation and translation steps
  - o Define, save, play back path
  - o Save frame buffer image

- Save and load current desired view
- Miscellaneous
  - Application should start out with meaningful view and navigation steps for the geometry loaded
  - Make a scene with the geometric models given; the geometric models should be placed in a directory called geometry off the current directory
  - Save an interesting path with multiple key frames

### Extra credit

- Make a movie file corresponding to your path (2%)
- Non-linear path interpolation (2%)
- Contour line rendering mode (2%)

### Turn in instructions

Turn in your work in an archive submitted via Blackboard Vista. The archive should contain:

- Source code and executables
- Path text file
- DO NOT turn in the geometry files
- External libraries used
- Code should compile, link, and run
- A Readme.txt or Readme.doc file that lists special GUI features and extra credit attempted

---

© [popescu@cs.purdue.edu](mailto:popescu@cs.purdue.edu), 2011