Computer Graphics - 234325

Winter 2019/2020

HW3 – Wireframe renderer & Transformations

Release Date: 09/12/2019

Submission Date: 30/12/2019 23:59

Description:

Please note: Please download new iritSkel.cpp, iritSkel.h files and replace the ones in your visual studio solution. These new files include a method for computing a 4X4 matrix inverse.

In this exercise you are required to upgrade your renderer from the previous homework to support the following:

1. Solid rendering (in addition to the current wireframe) with hidden surface removal.
2. Multiple light sources (parallel and point).
3. Flat, Gouraud and Phong shading models.

For that you need to implement the following:

1. **Polygon scan conversion:**
   You can assume in your implementation that you deal only with convex polygons, all the vertices attributes (color, normal, depth) should be interpolated at this stage.

2. **Z – buffer:**
   Use the Z-buffer algorithm (image space – after projection) to eliminate hidden surfaces.
   Hint: you can use the depth value after the perspective warp, as it still reflects monotonically the original transformed z value.

   We don't require optimal memory consumption, so you are not required to implement the scan-line z-buffer version.

3. **Light calculation:**
   You need to calculate the final color for each visible pixel (in solid rendering mode) according to the Flat/Gouraud/Phong light model presented in lectures which includes ambient, diffuse and specular intensities.

Features:

Add the following features, with appropriate and intuitive user interface:

1. **Light management:**
   Add a way to specify the different parameters for several light sources (up to 7), each light source can be enabled or disabled, and has the following attributes:
   - position (relevant in case of point light source)
   - direction (relevant in case of parallel light source)
   - Diffuse and specular color intensities.

   Additionally, provide a way to specify the ambient intensity and color for the whole scene, and the specularity exponent (one for the whole scene).

   You are strongly recommended to use the light sources dialog that came with the skeleton code.

2. **Flat/Gouraud/Phong Shading models:**
   Add a way to select which shading model is used. If the normal is not given per vertex, you need to calculate it as in the previous homework.

3. **Solid rendering:**
   Since you are implementing a software renderer, the expected performance for full solid scene rendering won’t be in a real time rate.

   Add “Render” menu with two sub menu items:
   - **“On Screen”:** Renders the scene on the screen (Full rendering with all the scene parameters)
   - **“To File”:** In this option the scene should be rendered to a file in PNG format. The user can specify the size of the image to save (the window size by default). If a different size than the window size is chosen, the object should be scaled appropriately for the render process, just as if the window was resized to the given size (this change should only be reflected in the saved file). You should use the PNG library linked to the given skeleton.

4. **Back face culling:**
   Add an option to draw only front facing polygons in both rendering modes (wireframe and solid).
   Note: This should work in orthographic and perspective modes.

5. **Normal inverse and calculation control:**
   Add a way to invert (flip them 180 degrees) the normals of the model (face and vertex normals).

   Add a checkbox that indicates whether to calculate the vertices’ normals always or to read it from the file. If the checkbox is checked then the vertices normal should be calculated as defined in the previous HW (as an average of neighboring faces normals).

6. **Background image:**
   Add a way to select a background image from a PNG file, and also provide a way to cancel this selection.

   Provide a way to select one of the following image layouts in the background:
   - Stretch: the image is stretched to fit the whole rendering area (in both directions)
   - Repeat: the image is repeated in both directions according to its original dimensions, until the rendering area is covered.

   This option is relevant only in solid rendering mode (in Render menu).

7. **Silhouette highlighting:**
   Add a way to enable drawing the silhouette line of the model. A silhouette edge is an edge that connects front and back facing faces. Provide a method for highlighting the Silhouette, either by a user selected color or a thicker line, or any other user intuitive method you think of.

Final Notes:

- Submit a zip file containing your code at the web site. Do not submit the Release or Debug directories or the build files like CGWork.ncb. If your zip file takes more than 5 megabytes, you are doing something wrong.

- DO NOT USE any external code without permission and neither should you use other IRIT functions (use only the functions provided through the skeleton) without permission.

- Submit electronically a single zip file, named <ID1>_<ID2>_HW3.zip, where ID1 and ID2 are your id numbers. The file should contain the following:

  - A readme.txt file which includes your names, ids and emails.

  - The whole homework project with the solution files (VS2017). Again, don't include the intermediate and compilation files in the Release and Debug folders.

  - Late submission should be coordinated with the TA. A penalty of 3 points will apply for each day of delay, if not justified.

Good luck and enjoy!