Farhad Kamangar kamangar@uta.edu

Assignment 04 (Due date Nov. 3, 2024)

Computer Graphics

Assignment 04

DUE DATE: Nov. 3, 2024 11:59 PM

Purpose: Practice in perspective projection and clipping

Add perspective projection to your previous assignment and include multiple cameras. This assignment must include parallel and perspective projections including corresponding clippings.

When you start your program it should automatically read a file called "cameras.txt". This file contains the information about the cameras. Notice that "cameras.txt' file may include information about multiple cameras.

- **Load (Open) Input File**
  This command is the same as the previous assignment. except that it shows all the views from different cameras (parallel, perspective) simultaneously.

Notes:

- All the commands from the previous assignments should be functional in this assignment.
- The format of the "cameras.txt" file is the same as the previous assignment, except it will include multiple cameras.
- The format of the input file is the same as the previous assignment.
- If any of the parameters are missing then the default value should be used.
- Boundaries of the viewports must be displayed.
- Viewports may be overlapping.

- Your program must display all the viewports (for all the cameras) simultaneously.

Detail format of the cameras.txt file:

```
c       // Start the definition of a new camera.

i  camera_name  //Name of the camera. Default=""

t Parallel / Perspective  // Define camera type. Default= parallel

r <x> <y> <z>   //Define VRP (WC). Default= 0,0,0

n <x> <y> <z>  //Define VPN (WC). Default = 0,0,1

u <x> <y> <z>  //Define VUP (WC). Default=0,1,0

p <x> <y> <z>  //Define PRP (VRC). Default=0,0,1

w <umin><umax><vmin><vmax><nmin><nmax>    //Define View Volume (VRC). Default= -1 , 1, -1, 1, -1, 1

s <xmin><ymin><xmax><ymax.    //Define viewport (normalized coordinates). Default=0.1, 0.1, 0.4, 0.4

// Definition of another camera may start here

c      // Start the definition of another new camera....

...
```

Detail format of the input file:

Each line of the input file will contain either a vertex or a face definition as described below:

```
v <x1> <y1> <z1>  //Define a vertex

v <x2> <y2> <z2>

.

.

.

v <xn> <yn> <zn>

f <u1> <v1> <w1> //Define a face (u,v, and w are integers corresponding to the vertex number))

f <u2> <v2> <w2>

.

.

.

f <um> <vm> <wm>
```

Notes:

- Each "v" line defines a new vertex with the given x,y,z coordinates.

- Each vertex is given a unique identifier starting from 1 (*not 0*).
- Each "f" line defines a new triangular face whose corners are the vertices with the given identifiers u,v,w.
- Your program must include 3-dimensional clipping.
- Boundaries of the viewports must be displayed.
- The user should be able to resize the canvas and the content of the canvas should be automatically refreshed proportionally.

Sample cameras file:

- cameras.txt

Sample input files:

- Pyramid
- Cow
- Bunny
- Teapot

Demo solution for Assignment 04

Recommended steps to complete this assignment:

- Read the "cameras.txt" file and calculate the composite matrix for each camera (some of the cameras are defined as parallel and some as perspective).
- Read a data file
- For each face in the data file:
    - Multiply each vertex by the sequence of transforms to bring the vertices from object space to world space. These transforms are the ones which are specified by the user in assignment 02. Note that these transformations are different from the parallel or perspective projection transformations. (Same as previous assignment)
    - for each camera in cameras list:
        - Multiply each vertex by the composite matrix for corresponding projection (parallel or perspective).
        - For each side of a face:
            - if the current camera is parallel: (Same as previous assignment)
                - Clip the line in 3d against the standard parallel volume bounded by x=-1 ; x=1 ; y=-1 ; y=1 ; z=0 ; z=1 planes..
                - If the result after clipping is not rejected:
                    - Drop the z value of the results after clipping (use the x and y of

as two dimensional data to be used for mapping from window to viewport).

- else:
  - Clip the line in 3d against the standard perspective volume bounded by x=z ; x=-z; y=z ; y=-z; z=zmin ; z=1 planes..
  - If the result after clipping is not rejected:
    - Calculate the new x' and y' by dividing the x and y of each vertex by its z. i.e., x'=x/z and y'=y/z. Use x' and y' as two dimensional data for mapping from window to viewport.
- Map from window to viewport and draw the line on the screen(this part is from assignment 01). Notice that the xwmin=-1 , ywmin=-1 , xwmax=1, and ywmax=1

Top