



- Home
- Syllabus CSE-4303-001 & CSE-53...
- Teaching Assistants
- Assignments
- Handouts, Notes, and Supplemen...
- Links to related graphic sites
- Topics for Exam_01
- Topics for Exam_02
- Old Tests

[Home](#) > [Assignments](#) > Assignment 03 (Due date Oct. 20, 2024)[« Previous](#) [Next »](#)**Assignment 03 (Due date Oct. 20, 2024)**

Computer Graphics

Assignment 03

DUE DATE: Oct. 20, 2024 11:59 PM

Purpose:

- Practice three-dimensional transformations.
- Practice three-dimensional clipping.
- Implement parallel projection mapping.

This assignment extends Assignments 01 and 02 by incorporating parallel projections and 3D clipping. Your task is to develop a graphics package that reads an input file and displays the parallel projection of the objects. All objects should be properly clipped. Additionally, all commands from the previous assignments should remain functional.

When you start your program, it should automatically read a file called "cameras.txt", which contains the camera's viewing parameters. This file should be read only once at startup and should not be read again when loading input files.

- **Load (Open) Input File**

This command loads all the data and parameters from the input text file and displays the objects using parallel projection. All objects should be clipped in 3-d and displayed in the viewport.

- **Fly Camera**

This command moves the VRP (View Reference Point) from Point A to Point B in 100 incremental steps, while keeping VPN (View Plane Normal), VUP (View Up Vector), PRP (Projection Reference Point), and all other viewing parameters constant. The intermediate results should be displayed. The user must specify the coordinates of points A and B.

Notes:

- Multiple cameras may be defined in the cameras.txt file.
- Each c line marks the start of a new camera definition.
- For this assignment, only process the viewing parameters for the first camera.
- In future assignments, views from multiple cameras will be displayed simultaneously.
- The set of r, n, u, p, w, and s lines defines the viewing parameters and viewport.
- If any parameters are missing, use the default values.

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 03](#) (To run the demo, make sure the "cameras.txt" file is in the same folder as the exe file")

Recommended Steps:

- Read the "cameras.txt" file and calculate the composite matrix for parallel projection.
- Read the input file and process each face:
- For each vertex, apply the sequence of transforms to convert the vertices from object space to world space (as specified by the user in Assignment 02).
- Apply the composite matrix for parallel projection.
- 3D Clipping:

- Clip each side of a face against the standard parallel volume bounded by the planes $x = -1$, $x = 1$, $y = -1$, $y = 1$, $z = 0$, $z = 1$.
- If the result is not rejected after clipping:
 - Drop the z value (as in Assignment 01, now you have 2D data).
 - Map from window to viewport and draw the line on the screen (this is from Assignment 01).
 - Use the bounds $xwmin = -1$, $ywmin = -1$, $xwmax = 1$, $ywmax = 1$.

[⬆ Top](#)