TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
CENTRAL CAMPUS, PULCHOWK

A

REPORT

ON

3D OBJECT (PISTOL)

SUBMITTED BY:

Animesh Timsina (073-BCT-520)

Deepak Pandey(073-BCT-514)

Lakshya Pandit(073-BCT-520)

SUBMITTED TO:

Department of Electronics and Computer Engineering

06th March, 2019

Objectives:

The main objective of this project is to become familiar with creating 3D and 2D graphics objects in computer. Creating graphics in computer require various graphics routines to be performed and our objective is to gain knowledge in using such routines and to list out, they are:

- To acquire knowledge in 2D and 3D Geometric Transformation including basic transformations such as translation, rotation, scaling etc.
- To know about 3 - dimensional object representation using polygon tables including vertex table, edge table and surface table.
- To implement the different colors in 3D objects using various fill algorithms.
- To apply algorithms for visible surface detection and hidden line detection (Z – buffer).
- To apply algorithms on Illumination Models and Surface Rendering Methods such as Ambient light, Diffuse Reflection, Specular Reflection and Polygon rendering methods as Gouraud Intensity Shading.

Introduction:

Computer graphics has been evolving and developing more and more over time. With increasing time the features and results of computer graphics have been more efficient than before. Since the very beginning, different projects of computer graphics are being developed which used different algorithms for their implementation. Its major applications are:

a. Computer aided design: a major use of computer graphics is in design process particularly in engineering and architectural system but now almost all products are computer designed.

b. Presentation graphics: Computer graphics are also used to produce illustrations for reports or generate slides or transparencies for projectors.

c. Computer art: Computer graphics are widely used in both fine and commercial art applications.

d. Education and training: Computer generated models of physics, finance and economics are often used as educational aids.

e. Visualizations: Computer graphics are used to analyze large amounts of information to study the behavior of certain processes.

f. Image processing: Computer graphics are used on image processing to modify or interpret existing pictures.

g. Graphical user interfaces

Basic Theory:

When dealing with 3D graphics, the following five main steps which were used to obtain the final product are:

1. Modeling - creating objects that will be on stage
2. Texturing - determination of surface properties of objects to simulate the properties of real objects (color, texture, transparency, brightness, etc.)
3. Lighting - adding and placing light sources in the same way as it is done in the theater, for example; at this stage, also, setting shadows
4. The animation - changing the position of the object or its properties (color, transparency, etc.)
5. Visualization - creation of the final image or animation

Using these steps our 3D model, a model was rendered. The project is thus an application of many graphics algorithm and processes that we studied in syllabus.

Methodology:

The graphics image (a gun) was rendered using Babylonjs 3D engine that is based on WebGL/ Web Audio and javascript. The algorithms used to render the 3D image are discussed below:

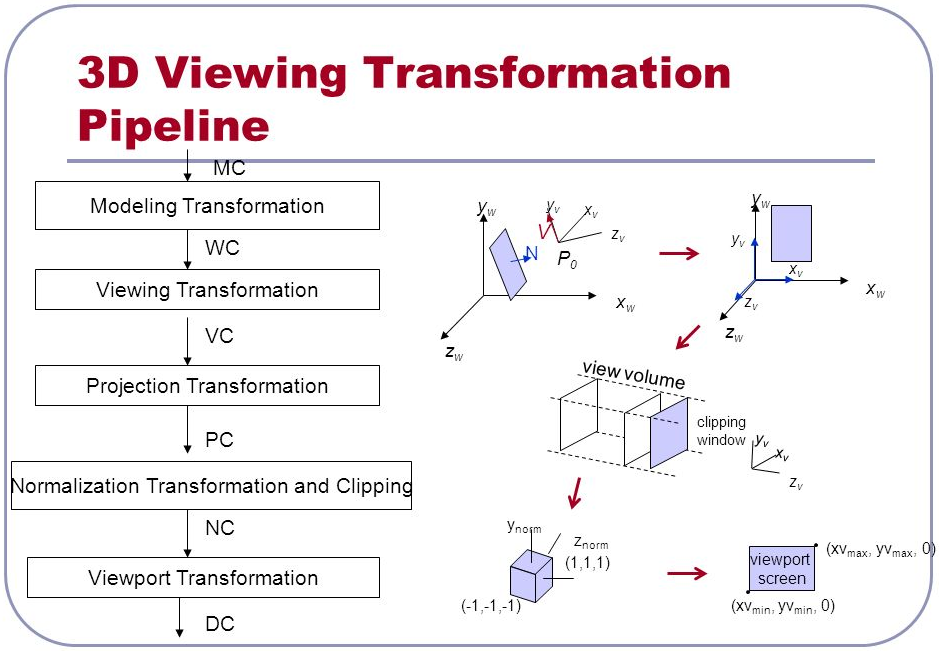
-> Blender was used to model the object

-> Babylon add-on was installed on Blender to export the model in JSON format

-> Functions were written in javascript

->HTML5 and CSS was used as rendering environment

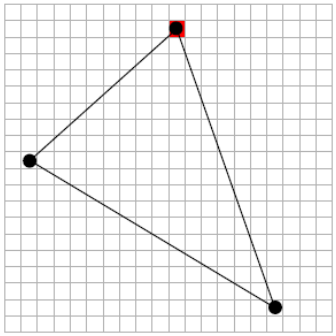
1: Viewing Pipeline:

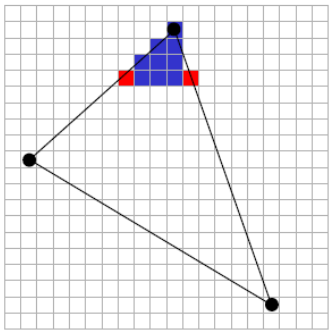


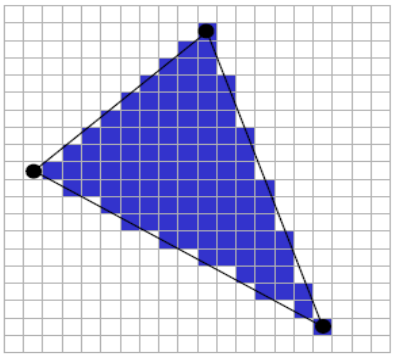
Generating a view of a three-dimensional scene is somewhat analogous to the processes involved in taking a photograph but we have more flexibility and many more options for generating views of a scene with a graphics package than with a camera. The figure above shows the general processing steps implemented for 3d modeling of the scene at a junction and converting the world-coordinates to device coordinates. First, the 3d object was modeled by generating coordinates manually. For modeling the structure, we made the design of car in Photoshop and then exported in object format through which we got the coordinates to be manipulated by our code. And thus, world coordinates were obtained. Then, the world coordinates were converted to viewing coordinates. Next, perspective projection operations were performed to convert the viewing-coordinate to coordinate positions on the projection plane. Then visible surface were identified using depth buffer and were rendered using Goraud's Shading Model.

2: Scan-line Algorithm:

Walk along edges one scanline at a time and then rasterize spans between edges.







3: Gouraud Shading and Illumination model:

This intensity-interpolation scheme, developed by Gouraud and generally referred to as Gouraud shading, renders a polygon surface by linearly interpolating intensity values across the surface. Intensity values for each polygon are matched with the values of adjacent polygons along the common edges, thus eliminating the intensity discontinuities that can occur in flat shading.

Each polygon surface is rendered with Gouraud shading by performing the following calculations:

- Determine the average unit normal vector at each polygon vertex.
- Apply an illumination model to each vertex to calculate the vertex intensity.

- Linearly interpolate the vertex intensities over the surface of the polygon

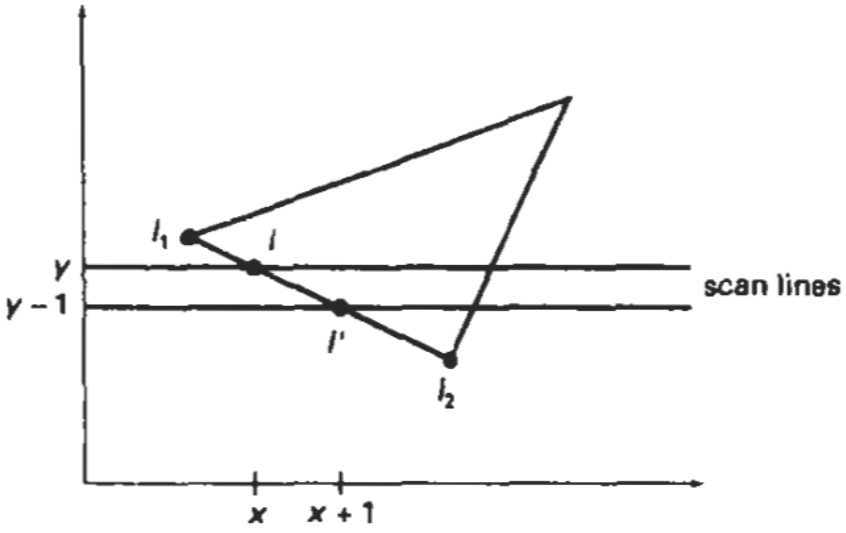


Fig 1. Incremental interpolation of intensity values along a polygon edge for successive scan lines

Incremental calculations are used to obtain successive edge intensity values between scan lines and to obtain successive intensities along a scan line. As shown in Fig. 14-46, if the intensity at edge position (x, y) is interpolated as:

$$I = \frac{y - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y}{y_1 - y_2} I_2$$

then we can obtain the intensity along this edge for the next scan line, y - I, as:

$$I' = I + \frac{I_2 - I_1}{y_1 - y_2}$$

4:. Visible surface detection Z-buffer method:

In computer graphics, Z buffering also known as depth buffering is the management of image depth coordinates in three-dimensional graphics. It is one solution to the visibility problem which is the problem of deciding which elements of a rendered scene are visible and which are hidden. When an object is rendered the depth of a generated pixel (z-coordinate) is stored in a buffer (the z buffer or depth buffer). This buffer s usually arranged as two dimensional arrays (x-y) with one element for each screen pixel. If another object of the scene must be rendered in the same pixel,

the method compares the two depths and overrides the current pixel if the object is closer to the observer. The chosen depth is then saved to the z-buffer, replacing the old one. In the end the z buffer will allow the method to correctly reproduce the usual depth perception; a close object hides the farther one.

ALGORITHM:

a. Initialize the depth buffer and refresh buffer so that for all buffer positions (x,y),

depth(x,y)=0, refresh(x,y)=I(background)

b. For each position on each polygon surface compare depth values to previously stored values in the depth buffer to determine visibility.

Calculate the depth z for each (x,y) position on the polygon

If z>depth (x,y) then set

Depth(x,y)=z, refresh(x,y)=Isurf(x,y)

Where, I(background) is the value for the background intensity

Isurf(x,y) is the projected intensity values for the surface at pixel position (x,y)

5: Projection Techniques:

Perspective view is the real view as perceived by an eye. For real 3d modeling of any object, perspective projection algorithm is implemented to convert 3d world coordinates to 2d screen coordinates. For simplicity, camera/eye is aligned perpendicular to the XY-plane and the projection plane is parallel to XY-plane. Each vertex of the object in 3d is projected onto the projection plane and 2d projected coordinates is calculated. Then, necessary viewport transformations are applied in order to obtain the 2d screen coordinates. The viewing system used is **Right Hand Viewing System.**

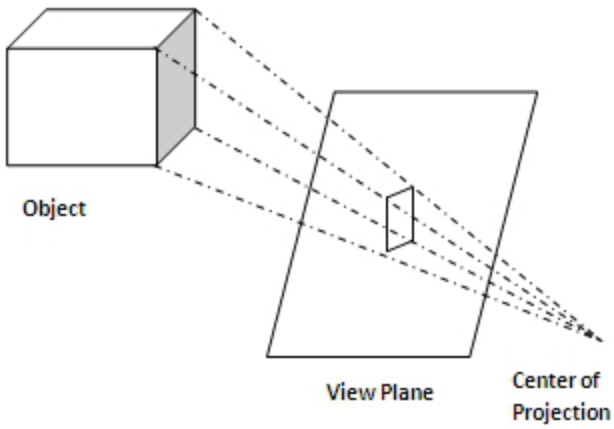


Fig 5.1. Perspective Projection

6: 3D Rotation:

Rotation in 3d is performed about arbitrary axes parallel to coordinate axes passing through the centre of the structure. The whole structure is rotated in three directions by translating the arbitrary axes to the coordinate axes. The formulae used for rotation are:

Rotation about x axis:

y' = y*cos (xangle) - z*sin (xangle)

z' = y*sin (xangle) + z*cos (xangle)

Rotation about y axis:

x' = x*cos (yangle) + z*sin (yangle)

z' = -y*sin (yangle) + z*cos (yangle)

Rotation about z axis:

x' = x*cos (zangle) + y*sin (zangle)

y' = x*sin (zangle) - y*cos (zangle)

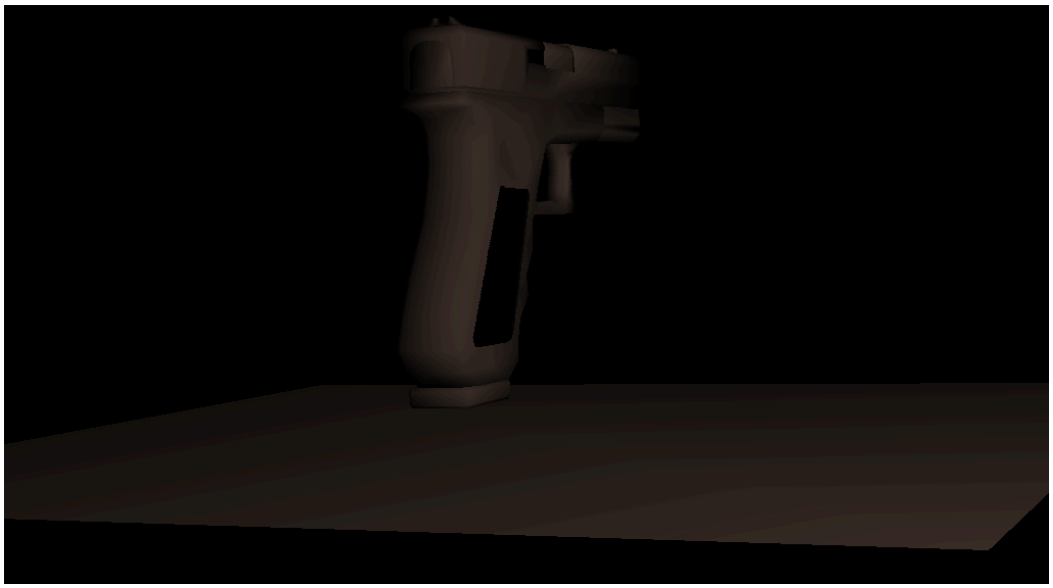
Outputs:

Light Source in front

Light Source to the left

Light source to the right

Different positions:



Conclusion:

Hence, in this project we gained sufficient amount of knowledge related to Computer Graphics that can also be applied to other graphics related projects. Also the application of computer graphics principle to develop a system useful in real life was made.Most importantly, we learned several algorithms related to line drawing, shading, transformation, projection, et al. We also learned to work as a team and learned the optimization of system, as the field of computer graphics requires extensive processing.

We also learned to know our limitations which can be very useful for future projects and systems

References:

https://www.davrous.com

https://doc.babylonjs.com/