# A Project on Computer Graphics

# -

# **3D Mirror**

By:
Aashish Poudel (060BEX401)
Anup Shrestha (060BEX405)
Bikash Sharma (060BEX407)

# Objectives

- Utilize various techniques and algorithms of Computer Graphics.
- Draw output primitives such as line, circle, ellipse, polygons and use them to draw various 3D shapes.
- To be able to understand 3D object representation using polygon tables including vertex table, edge table and surface table.
- Become familiar with attributes of objects like line type, color, fill styles, fill color etc.

# Objectives…

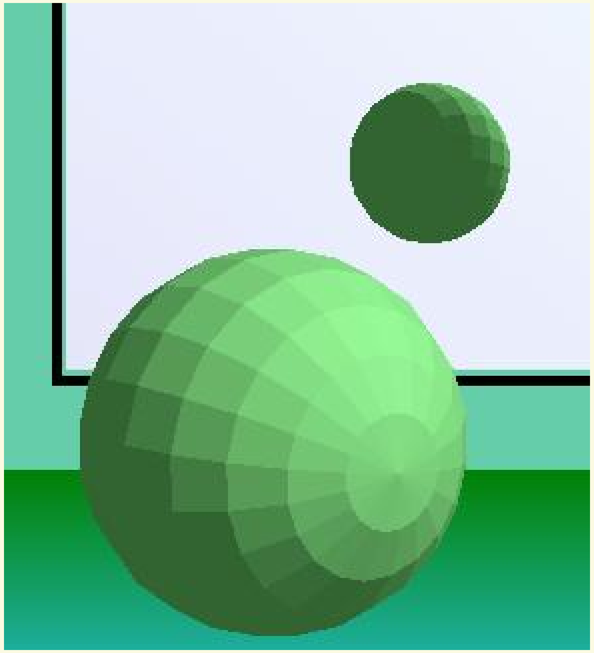
- To simulate the mirror reflection of some specific 3D objects.
- Perform various 3D transformations like rotation, scaling on objects and view their corresponding changes in mirror image.
- Understand algorithms for visible surface detection.
- Understand algorithms on Illumination Models and Surface Rendering Methods such as Diffuse Reflection, Specular Reflection, Ambient Reflection and Polygon rendering methods such as Constant Intensity Shading.

# Introduction

- Computers - powerful tool for rapid and economical creation of 2D and 3D graphics.
- Simulation of virtual 3D world can be done in a 2D computer screen.
- Used in science, engineering, medicine, business, industry, government, art, entertainment, advertising, education and trainings, in user interface, data visualization, television, commercials, motion pictures, simulation and much more.

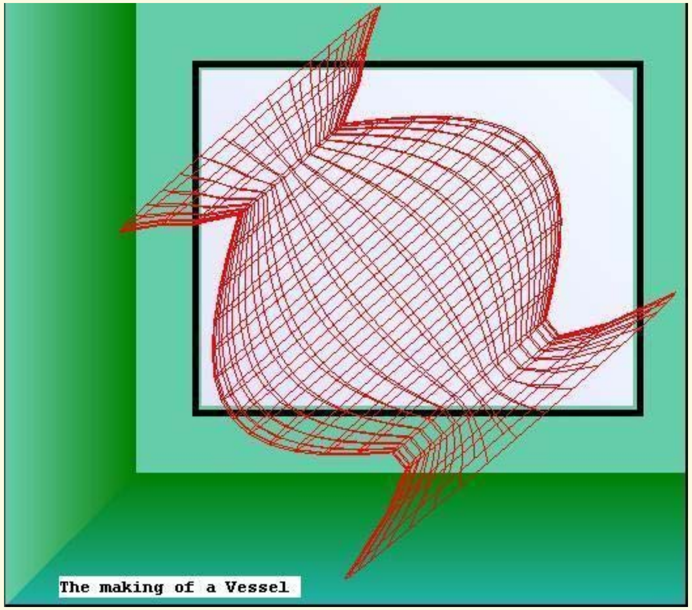
# Introduction to 3D Mirror

- This project - visualization of different 3D objects from the Z axis  and their mirror images in a 3D world.
- 3D objects are modeled by using vertex and surface tables.
- Algorithms like hidden surface detection and different lighting models are used for improving the realism.



# Vertex and Surface Table

- Vertex Table
    - stores the coordinates of the vertices. i.e. x, y, z and w (homogenous)
- Surface table
    - 4 vertices of the surface
    - ABCD parameters
    - color value of surface



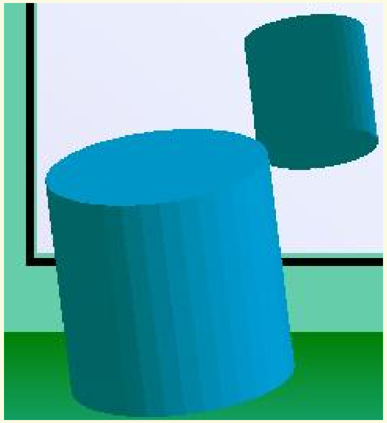
# Visible Surface Detection

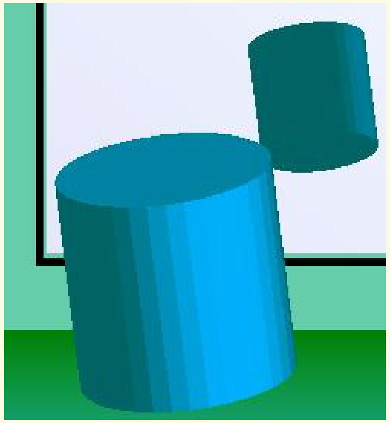
- ‘Back-Face Detection’ algorithm is used.
- Here the A, B, C, D parameters are calculated using the three vertices taken in anticlockwise order and solving the matrices.
- If V be a vector in viewing direction from the eye position then the surface faces backwards if $V.N>0$.
- Here viewing direction is towards negative Z axis. So $V.N=-C$

# Visible Surface Detection …

- So the surface with ‘negative C’ indicates that the back face of the object is seen and we do not draw it. Only the surface with ‘positive C’ is drawn.
- The painter’s algorithm has been implemented in vessel by sorting all the surfaces in a scene according to their distance from viewplane.

# Illumination and Surface Rendering





- Basic Illumination Models
- Diffuse Reflection
  - Fraction of incident light that is diffusely reflected can be set for each surface with parameter ‘d’
  - d = **N.L**/ |N||L|
- Specular Reflection / Phong Model
  - Result of reflection of the incident light from a concentrated region.
  - s = (2*d*C)/|N|

# Transformations

- Rotation
  - About all 3 Cartesian axes simultaneously.
  - Matrix multiplication
- Scaling
  - from 0 to 2 times of original size.
- Dragging / Translation
  - Left mouse button to drag left, right, up, down.
  - Right mouse button to drag in/out.

# Reflection about Mirror

- The Image tracks all the transformation in the real object.
  - Rotation and Scaling
    - Rotation about x & y axes result in opposite rotation of image.
    - Rotation about z axes result in same effect in image.
    - Scaling changes the size of both object and image
  - Dragging in all 6 directions
    - Dragging sideways (left, right, up, down)
    - Dragging in/out

# Methodology and Development Tools used

- **Object oriented** approach is chosen for this project.
- Event driven programming is also necessary to increase the user interactivity
- The development tool used for this purpose in our project is ‘Visual C#.NET”

# Problems Faced and Solutions

- The programming language ‘C#.NET’ was itself new for us.
- constructing Polygon tables.
  - Unable to use ‘Edge table’ due to the limited use of **pointers** in C#.
  - So we used only ‘Vertex and Surface tables’.
- visible surface detection
  - working on the ‘Z-buffer’ algorithm was slow and didn’t produce good results.
  - Instead we used ‘back-face detection’ algorithm

# Future Enhancement

- Due to time limitation and the gravity of the theory involved in Computer Graphics
  - Implementation of efficient visible surface detection algorithm for removing hidden surfaces.
  - Shadow  implementation.
  - Gouraud or Phong Shading for Surface Rendering.
  - Different View plane of looking the object.
  - A better Perspective Projection algorithm for both object and image.

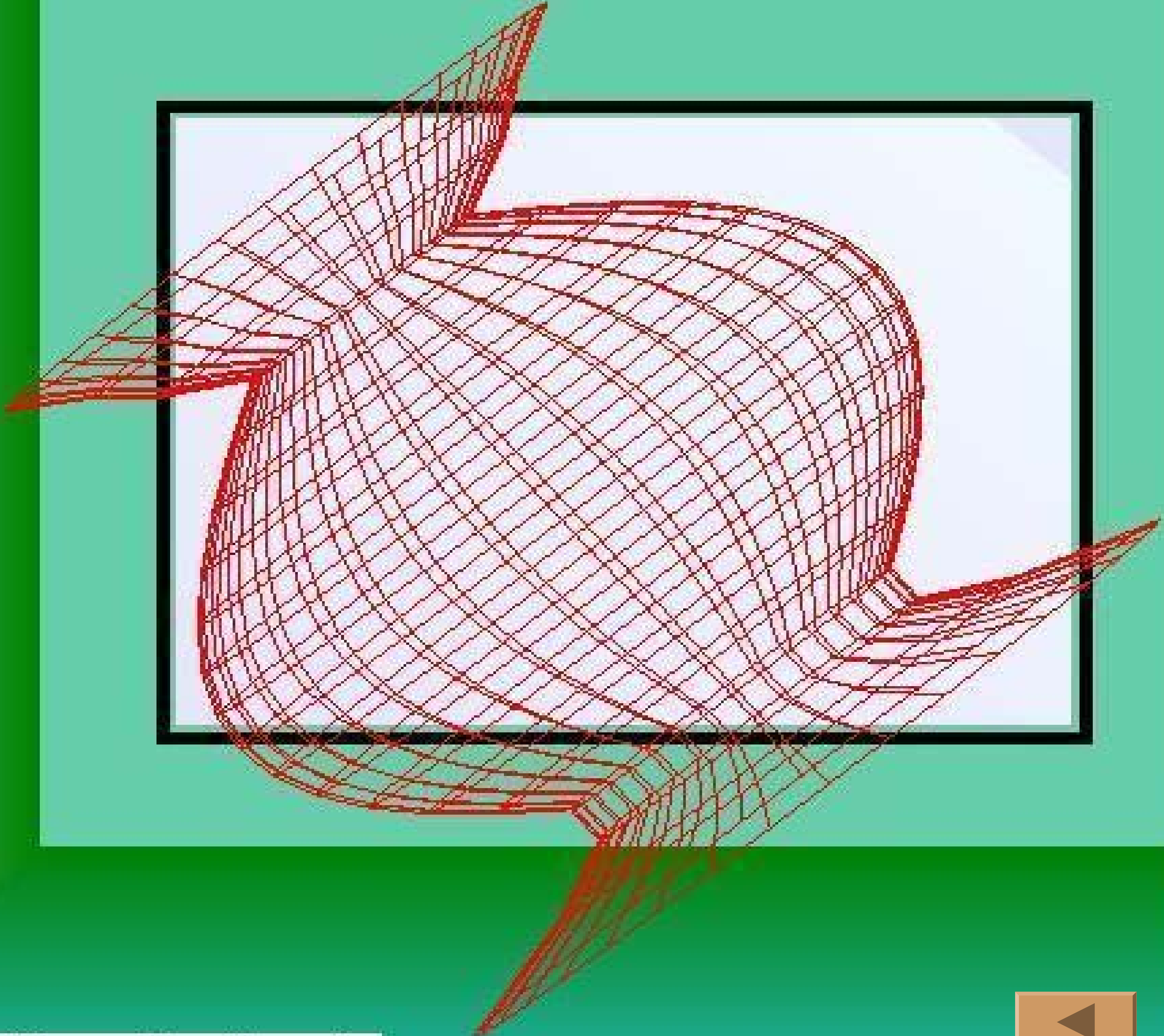
# Conclusion

- In a limited time, we tried our best to utilize most of the graphics algorithm for 3-D drawing and rendering.
- We learned many techniques and algorithms used in Computer Graphics field.
- knowledge of C++ and graphics theory was crucial for implementing the graphics techniques in ‘C#.NET’ platform.

# References

- **Computer Graphics,** 2nd Edition – Donald Hearn, M. Pauline Baker
- **Professional C#,** 3rd Edition – Simon Robinson, Christian Nagel, Jay Glynn, Morgan Skinner, Karli Watson, Bill Evjen
- **SAMS Teach Yourself C#.NET in 24 days.**
- www.google.com
- **Oreilly C#.Net Programming**, 2nd Edition – By Jesse Liberty

# Suggestions/Questions???



The making of a Vessel