VISVESVARAYA TECHNOLOGY UNIVERSITY

Jnana sangama, Belagavi, Karnataka-590018

CG MINI PROJECT REPORT

ON

BUMP MAPPING AND 3D LIGHT ANIMATION OF DONUT SHAPE

Submitted in partial fulfillment for the award of the degree of bachelor of engineering in computer science and engineering

SUBMITTED BY:

Kirti Tiwari (1KT20CS031)

Under the Guidance of

Prof. Kusuma.H.P

Assistant professor, Dept of CSE, SKIT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRI KRISHNA INSTITUTE OF TECHNOLOGY

BANGALORE – 560090

SRI KRISHNA INSTITUTE OF TECHNOLOGY

BANGALORE – 560090

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini project entitled "BUMP MAPPING AND 3D LIGHT ANIMATION OF DONUT SHAPE" as a part of 18CSL67 laboratory, is bonafide work carried out by a KIRTI TIWARI bearing USN:1KT20CS031 partial fulfillment for the award of degree in bachelor of Engineering in Computer Science Engineering fromVisvesvaraya Technological University, Belagavi during the academic year 2022-23. It is certified that all the corrections/suggestions indicated for internal assessment have been incorporated in reports submitted in the department Library. This mini project report has been approved as it satisfies the academic requirements in respect of the mini project report prescribed for award of said degree.

……	……	……..……
Signature of Internal Guide	Signature of the HOD	Signature of Principal
Mrs. KUSUMA H P	**Dr. SHANTHARAM NAYAK**	**Dr. MAHESHA K**
Assistant professor	Professor & head	Principal
Dept. of. CSE, SKIT	Dept. of. CSE, SKIT	SKIT, Bangalore

Name of the Examiner	**Signature with date**
1……	1……
2……	2……

ACKNOWLEDGMENT

Any achievement does not depend solely on individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities, have helped us in carrying out this mini project work. We would like to take this opportunity to thank them all.

This completion of mini project work brings a sense of satisfaction, but it is never complete without thanking the persons responsible for its successful completion.

I express my heartfelt sincere gratitude to **Dr SHANTHARAM NAYAK**, Professor and HOD, Department of Computer Science and Engineering, Sri Krishna Institute of Technology, Bengaluru, for his valuable suggestions and support.

I extend my special in-depth, heartfelt, sincere gratitude to my guide **Prof Kusuma H.P** Assistant Professor, Department of Computer Science and Engineering, Sri Krishna Institute of Technology, Bengaluru, for her constant support and valuable guidance for completion of the mini-project work.

I would like to thank all the teaching and non-teaching staff members in my Department Of Computer Science and Engineering, Sri Krishna Institute of Technology, Bengaluru, for their support.

Finally, we thank our Parents and Friends and Teacher for their moral support.

Kirti Tiwari (1KT20CS031)

DECLARATION

I, **KIRTI TIWARI**, student of 6th semester in Computer Science and Engineering, Sri Krishna Institute of Technology, Bengaluru, hereby declare that the mini project entitled, "**BUMP MAPPING AND 3D LIGHT ANIMATION OF DONUT SHAPE**" has been carried out by us under the supervision of **Prof KUSUMA H.P,** Assistant professor, Dept. of Computer Science and Engineering, Sri Krishna Institute of Technology, Bengaluru and submitted in partial fulfillment for the award of degree In Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological university, Belagavi during the academic year 2022 – 2023. We further declare that the report has not been submitted toany other University for the award of any other degree.

KIRTI TIWARI (1KT20CS031)

Place: Bangalore

Date:

ABSTRACT

A "**3D Donut Shape (Also known as Torus)**" is a surface of revolution generated by revolving a circle in three-dimensional space about an axis coplanar with the circle. If the axis of revolution does not touch the circle, the surface has a ring shape and is called a bump of revolution.

In this project, I designed the Bump mapping and 3D light animation of a Donut shape using OpenGL and the main theme of the project is to animate the Donut and I used few functions and also assigned the keys to make the Donut to rotate along x-axis, y-axis and z-axis and there is an option to increase and decrease the rotation speed and then to reset the Donut.

A function is written to switch the mode of the Donut from mesh to a polygon and also to increase and decrease the number of wraps. Coming to the Lighting of the Donut, I have placed two lights, one with Blue color and other with Yellow color on x-axis and y-axis respectively. Hence, we animate the Donut with lighting effects using OpenGL library in C programming language

TABLE OF CONTENTS

i. **Certificate**

ii. **Acknowledgement**

iii. **Abstract**

iv. **Table of Contents**

v. **Table of Figures**

TABLE OF CONTENTS

TABLE OF FIGURES

1. NTRODUCTION

The Computer Graphics is one of the most effective and commonly used methods to communicate the processed information to the user. It displays the information in the form of graphics objects such as pictures, charts, graphs and diagram instead of simple text.

In computer graphics, pictures or graphics objects are presented as a collection of discrete picture elements called **pixels**. The pixel is the smallest addressable screen element

Computer graphics today is largely interactive: The user controls the contents structure, and appearance of objects and their displayed images by using input devices, such as a keyboard, mouse, or touch-sensitive panel on the screen

Computer graphics concerns with the pictorial synthesis of real or imaginary objects from their computer-based models, whereas the related field of image processing treats the converse process, the analysis of scenes, or the reconstruction of models of 2D or 3D objects from their pictures.
The image processing can be classified as

- Image enhancement.
- Pattern detection and recognition
- Scene analysis and computer vision.

The image enhancement deals with the improvement in the image quality by eliminating noise or by increasing image contrast. Pattern detection and recognition deals with the detection and clarification of standard patterns and finding deviations from these patterns. The optical character recognition (OCR) technology is an practical example for pattern detection & recognition. Scene analysis deals with the recognition and reconstruction of 3D model of scene from several 2D images.

1.1 INTRODUCTION TO OPENGL

The OpenGL specification describes an abstract for drawing 2D and 3D graphics. Although it is possible for the API to be implemented entirely in software, it is designed to be implemented mostly or entirely in hardware.

The API is defined as a number of functions which may be called by the client program, alongside a number of named integer constants. Although the function definition is superficially similar to those of the C programming language, they are independent. As such, OpenGL has many language bindings, some of the most noteworthy being the Java Script binding WebGL (API based on OpenGL ES2.0); the C bindings WGL, GLX, CGL; the C binding provided by iOS and the Java and C binding provided by the Android.

In addition to being language-independent, OpenGL is also platform independent. The specification says nothing on the subject of obtaining, managing an OpenGL context, leaving this as a detail of the underlying windowing system. OpenGL is purely concerned with the rendering, providing no APIs related to input, audio or windowing.

1.2 GRAPHICS PIPELINE ARCHITECTURE

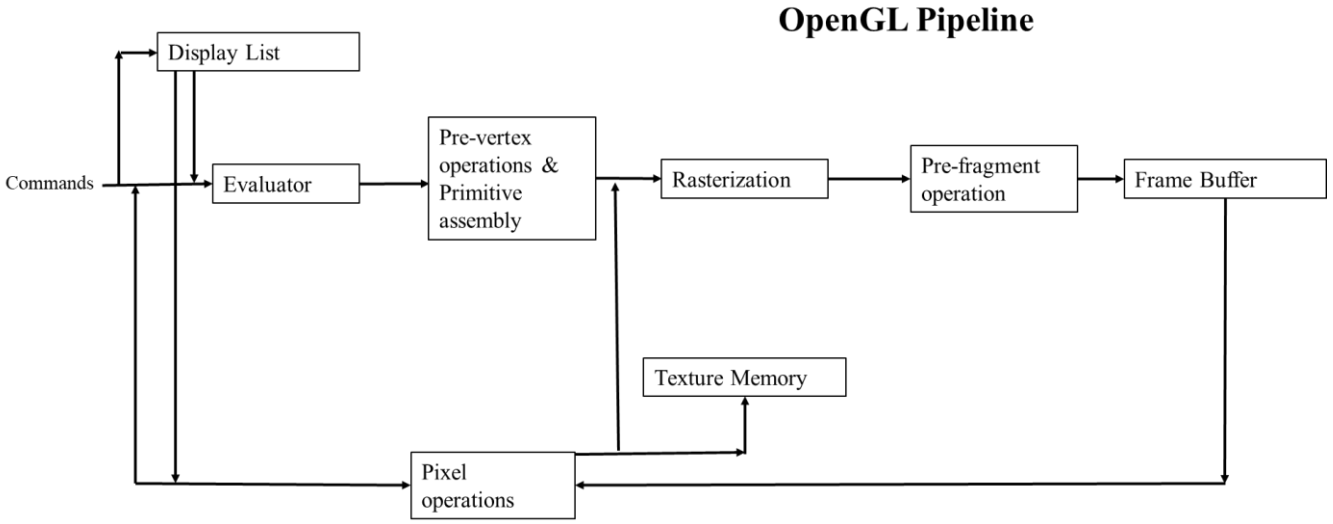

Fig. 1.2.1: Graphics Pipeline Architecture

1.3 PROCEDURE FOR CREATING PROJECT IN CODEBLOCKS

- Run Codeblocks (any available versions, used 16.01 in this project).
- Go to File → New → Project…
- A dialogue box will show up with different types of project option available to be created using Codeblocks 16.01.
- Select GLUT project. A new box pops up. Click on Next.
- Here, specify the project title and the destination folder of where your project is supposed to be saved in. Click on Next.
- Here, specify the location of GLUT stored in your computer. Generally it is stored in C:\Program Files (x86)\CodeBlocks\MinGW. Click on Next.
- Further Click on Next and then click on Finish to open the project in the console.
- Click on the project package and further open the src. Inside src, we can find the main cpp environment, where the Open GL, C program can be typed and executed.

SNAPSHOTS FOR CREATING A PROJECT IN CODEBLOCKS

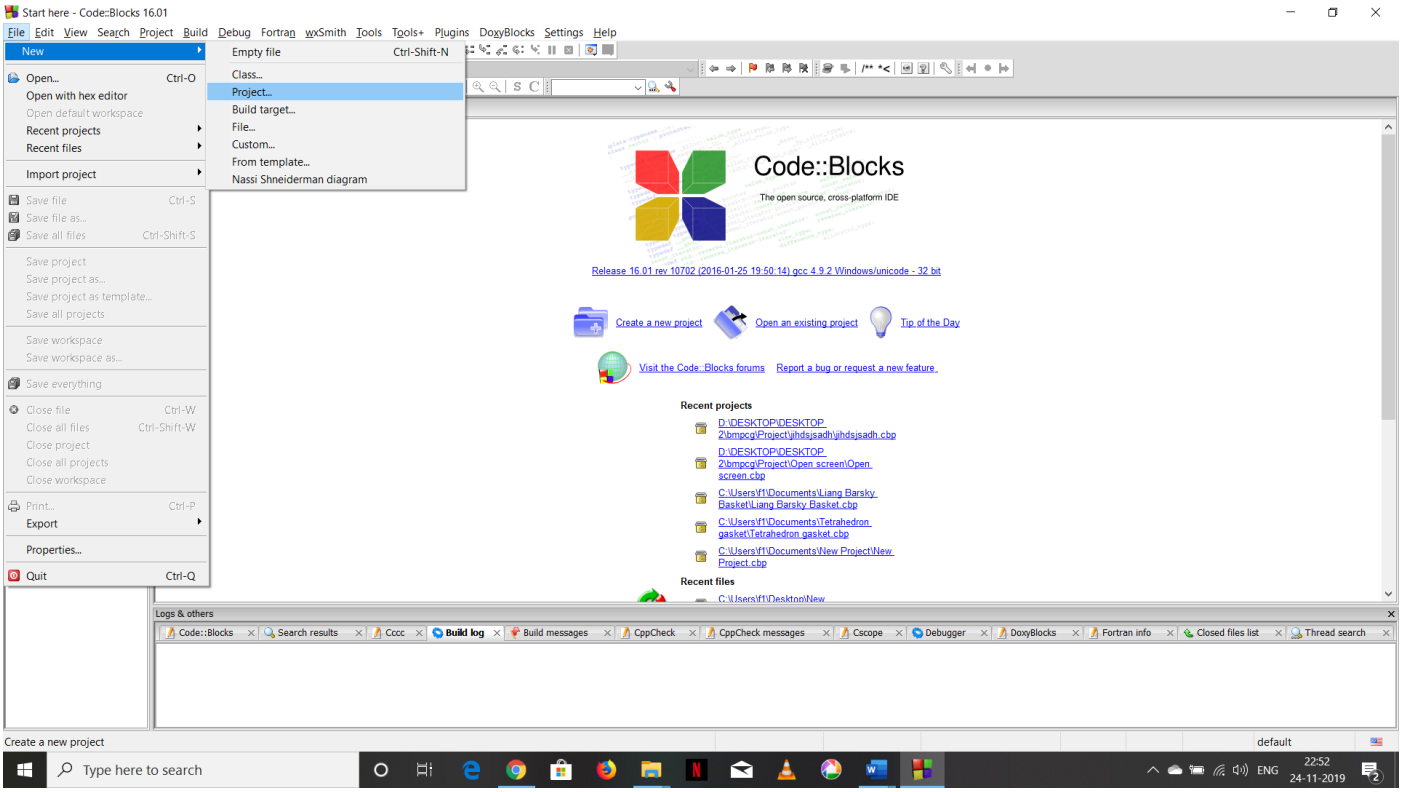

Fig. 1.3.1: Creating project (File->New->Project)

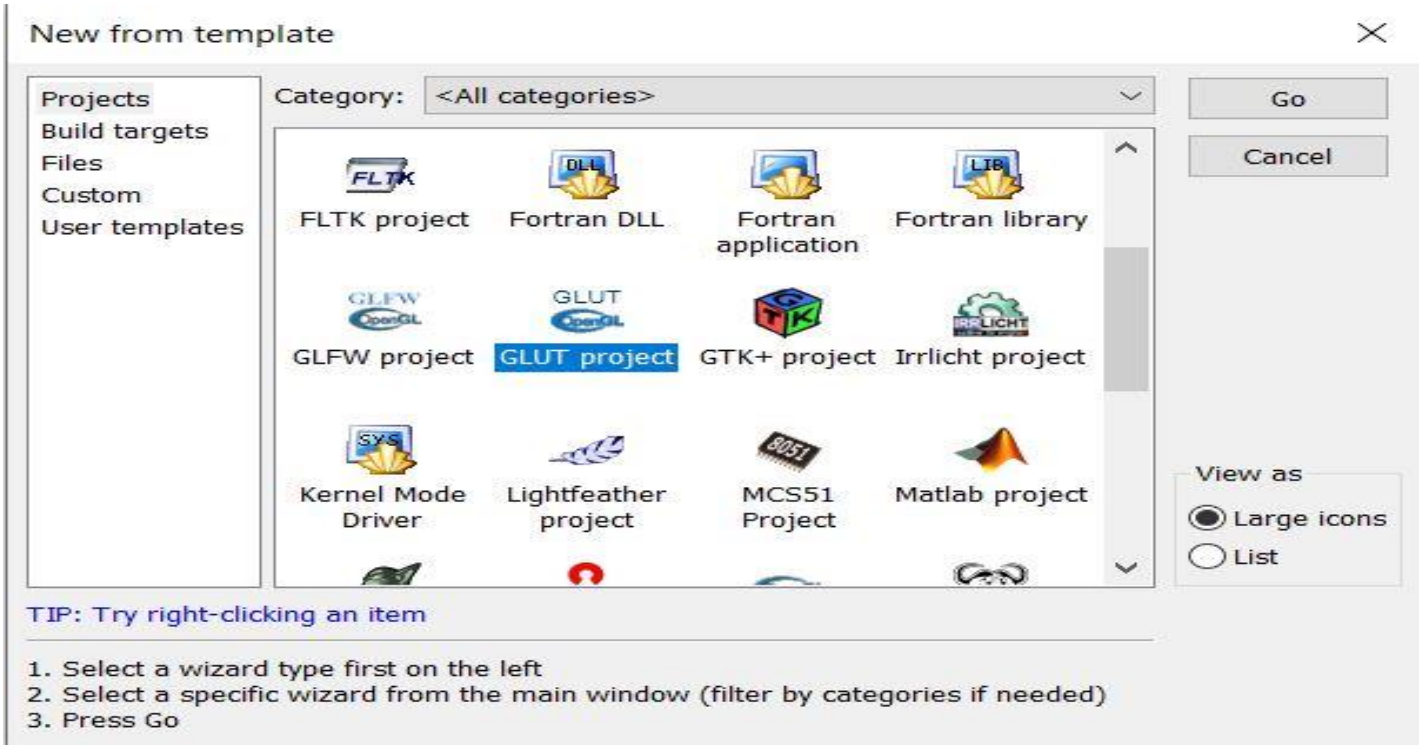

Fig. 1.3.2: Selecting GLUT Project

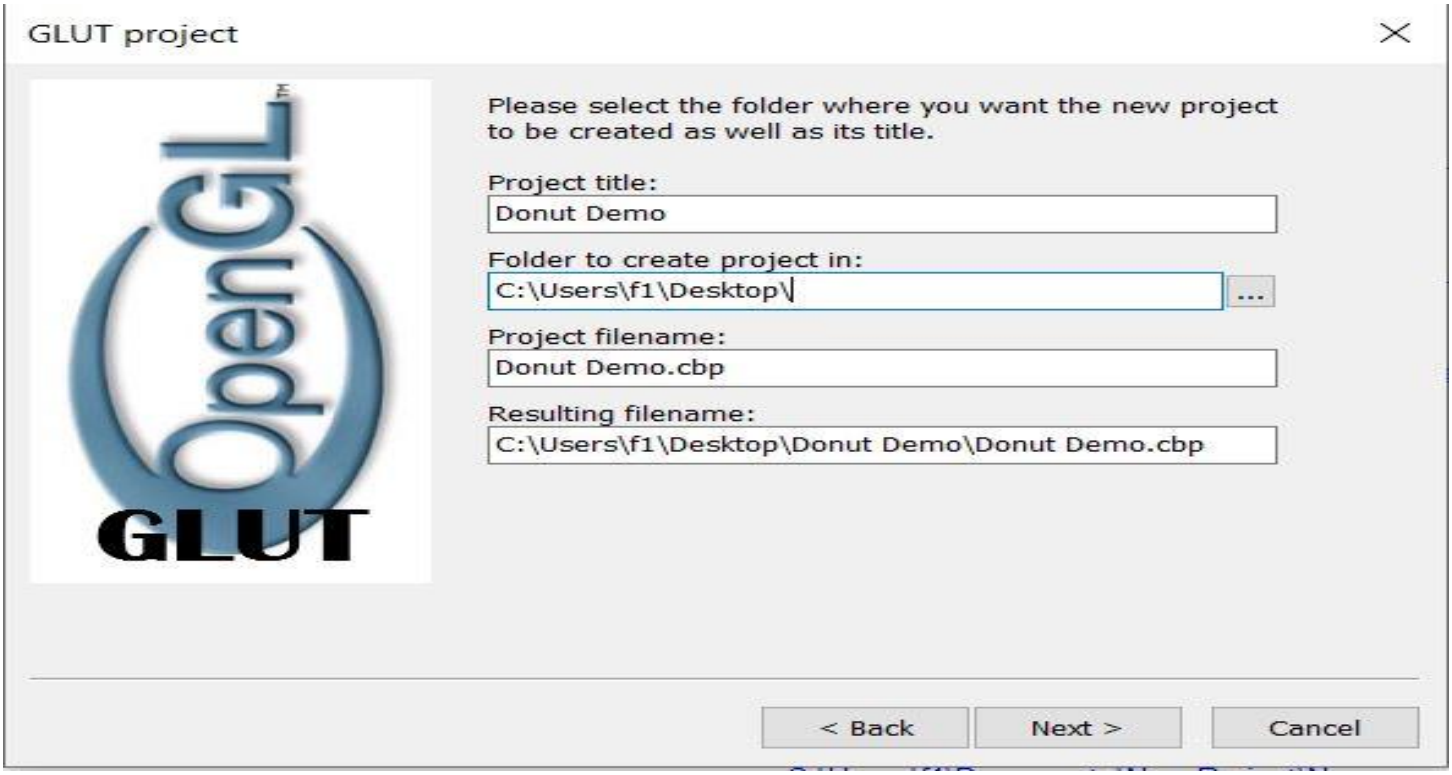

Fig. 1.3.3: Providing Project Credentials

1.4 ADVANTAGES OF INTERACTIVE GRAPHICS

- It provides a tool for producing pictures not only of concrete, "real-world" objects but also of abstract, synthesis of objects.
- It as an ability to show moving pictures, and thus it is possible to produce animations.
- With the use of interactive graphics, we can control the movement of an object. The interactive graphics provides tool called motion dynamics. With this tool user can move and tumble objects with respect to stationary observer, or he can make object stationary and the viewer moving around them.
- Interactive graphics provides facility called update dynamics.
- With the recent development of digital signal processing (DSP) and audio synthesis chip the interactive graphics can now provide audio feed back along with the graphical feed backs to make the simulated environment even more realistic.

1.5 AREAS OF APPLICATION OF COMPUTER GRAPHICS

- User interfaces
- Plotting of graphs and charts
- Office automation and Desktop publishing
- Computer aided Drafting and designs
- Process control
- Cartography

1.6 OPENGL PROGRAMMING LANGUAGE:

As a software interface for graphics hardware, OpenGL's main purpose is to render two- and three-dimensional objects into a frame buffer. These objects are described as sequences of vertices (which define geometric objects) or pixels (which define images). OpenGL performs several processing steps on this data to convert it to pixels to form the final desired image in the frame buffer.

2. REQUIREMENT SPECIFICATION

SOFTWARE REQUIREMENTS

Operating System: Windows 98/XP or Higher
Programming Language: C, C++
IDE: Dev C++ / CodeBlocks (Any version)
Toolkit: GLUT Toolkit

HARDWARE REQUIREMENTS

The package has been developed on:
Processor: Pentium Processor
Processor Speed: 333MHz
RAM: 32 MB or Higher
Graphics Card: 512MB
Monitor: Color
Keyboard: Low Profile, Dispatchable Type
I/O Parts: Mouse, Monitor

3. IMPLEMENTATION

3.1 Mini Project on Bump Mapping and 3D Light Animation of Donut Shape

3.1.1 Declaration and initialization of variables:

Here in this section we declare the basic variables that have been used in the project these variables help in controlling the animation, controlling the fineness of the polygonal mesh, controlling the size of the Donut, Mode flags, Lighting values, Material values etc and few of them are listed below

GL enum shadeModel = GL_FLAT========>Toggles between GL_FLAT and GL_SMOOTH
GL enum polygonMode = GL_LINE=======>Toggles between GL_LINE and GL_FILL
const float RotIncFactor = 1.5==========>Factor change in rot rate per key stroke
int Light0Flag = 1==================>Is light #0 on?

3.1.2 Keyboard Functions:

In this section we use few keys and assign them the functions such as Escape key to exit from the project, Letter ‘r’ to reset the animation, Letter ‘f’ to Shade mode toggles from flat to smooth, Letter ‘p’ to switch from mesh type of Donut to the polygon mode, Letter ‘w’ to decrease number of wraps around the Donut, Letter ‘W’ to increase number of wraps around the Donut, Letter ‘l’ to toggle between the local and non-local viewer, Letter ‘q’ to toggle between the triangle and the quadrilateral, Key ‘0’ and ‘1’ to toggle on and off the two lights etc. and few other keys used are listed below

GLUT_KEY_UP:
Either increase upward rotation, or slow downward rotation
KeyUp();
GLUT_KEY_DOWN:
Either increase downward rotation, or slow upward rotation
KeyDown();
GLUT_KEY_LEFT:
Either increase left rotation, or slow down rightward rotation.
KeyLeft();
GLUT_KEY_RIGHT:
Either increase right rotation, or slow down leftward rotation.
KeyRight();

3.1.3 Screen Update Function for the program

This is the function used to refresh the screen every time and update the Donut. The function updateScene is defined below:

```
updateScene
{
    Clear the rendering window
    Set the shading to flat or smooth.
    Set to be "wire" or "solid"

        Set up lights- two lights are set up
      if ( Light0Flag==1 || Light1Flag==1 )
        glMaterialfv->GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE
      glMaterialfv -> GL_FRONT_AND_BACK, GL_SPECULAR

    if ( Light0Flag==1 )
        Push Matrix
        Translate->Lt1pos[0], Lt1pos[1], Lt1pos[2]
        Material->GL_FRONT_AND_BACK, GL_EMISSION
        Create Solid Sphere
        Enable->GL_LIGHT0
    else
        Disable->GL_LIGHT0

    if ( Light1Flag==1 )
        Push Matrix
        Translate->Lt1pos[0], Lt1pos[1], Lt1pos[2]
        Material->GL_FRONT_AND_BACK, GL_EMISSION
        Create Solid Sphere
        Pop Matrix
        Enable->GL_LIGHT1

    else
        Disable->GL_LIGHT1

        Donut Materials
      Material->GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE
      Material->GL_FRONT_AND_BACK, GL_SPECULAR
      Material->GL_FRONT_AND_BACK, GL_SHININESS
      Material>GL_FRONT_AND_BACK, GL_EMISSION

    PushMatrix is use to save values to use again next time.
    Update the orientation of the Donut, if the animation is running.
```

```
        Set the orientation. Draw the Donut
        Restore to original matrix as set in resizeWindow.
        Flush the pipeline, swap the buffers
}
```

3.1.4 Initialize OpenGL Function:

This is initial rendering function defined with the following properties.

- Enable lighting calculations
- Turn on lights
- Ambient light
- Set position of lights

3.1.5 Resize Window Function

This function is written to resize the animation when the window size is altered. This function helps in smooth animation and viewing it on a large or small screen, as considered necessary.

- Use Viewport function
- Set up the projection view matrix
- Move system 10 units away to be able to view from the origin
- Tilt system 15 degrees downward in order to view from above the xy-plane

3.1.6 Navigating Function

This function is written to navigate to all the integrated sub-programs. Once the execution of the sub-program is complete the control comes back to the master function.

3.1.7 Main Function

Here we set up OpenGL, hook up callbacks, and start the main loop.

- Set the window position (from top corner), and size (width and height).
- Initialize OpenGL rendering modes.

- Set up callback functions for key presses
- Set up the callback function for resizing windows
- Call this for background processing
- Call this whenever window needs redrawing
- Start the main loop - glutMainLoop

3.2 USER CONTROLS

CONTROLLING RESOLUTION OF THE DONUT MESH
Press "W" to increase the number wraps.
Press "w" to decrease the number wraps.
Press "N" to increase the number of segments per wrap.
Press "n" to decrease the number of segments per wrap.
Press "q" to toggle between quadrangles and triangles.

CONTROLLING THE ANIMATION:
Press the "a" key to toggle the animation off and on.
Press the "s" key to perform a single step of the animation.
The left and right arrow keys controls the
rate of rotation around the y-axis.
The up and down arrow keys increase and decrease the rate of rotation around the x-axis. In order to reverse rotational direction you must zero or reset the Donut ("0" or "r").
Press the "r" key to reset the Donut back to initial
position, with no rotation.
Press "0" (zero) to zero the rotation rates.

CONTROLLING LIGHTS
Press '1' or '2' to toggle the first or second light off and on.
Press 'f' to toggle between flat and smooth shading.

COMMANDS SHOWING OPENGL FEATURES:
Pressing "p" toggles between wireframe and polygon mode.
Pressing "f" key toggles between flat and smooth shading.

4. RESULTS:

4.1 MINI PROJECT

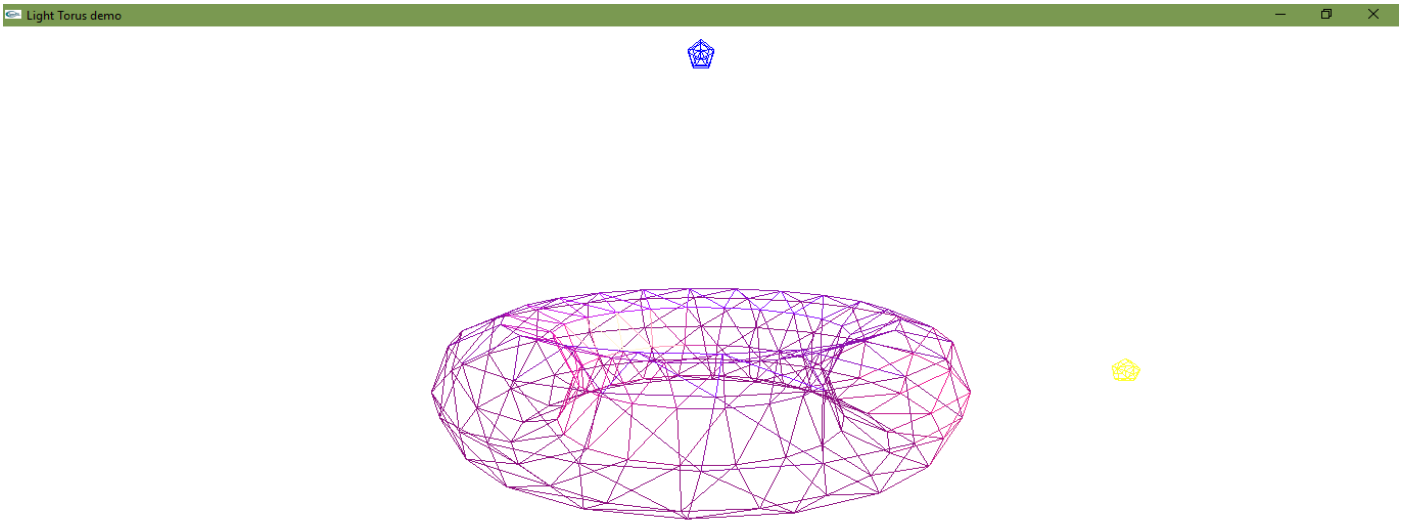

Fig 4.1.1: Initial Configuration

This is the initial configuration of the light Donut. It starts with a 15 degree inclination on the x-axis. And also, both the lights are ‘ON’.

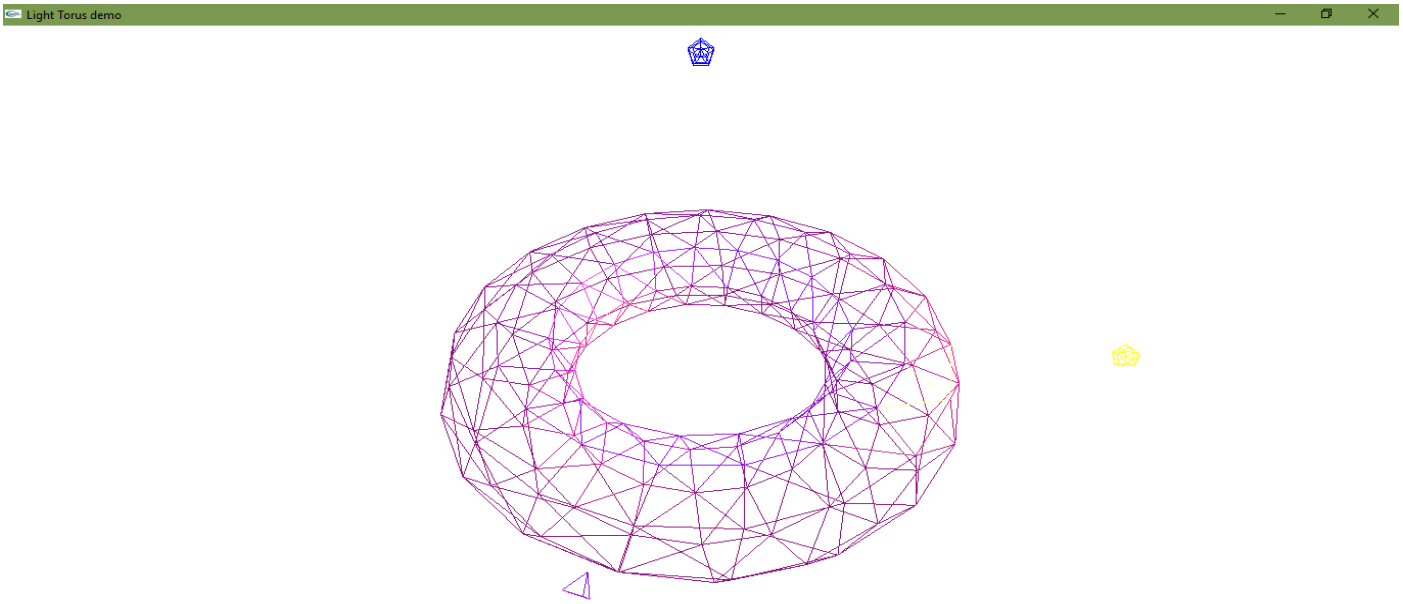

Fig 4.1.2: More vertices in Donut

Here we rotate the Donut along the x-axis and the y-axis with increased the number of vertices.

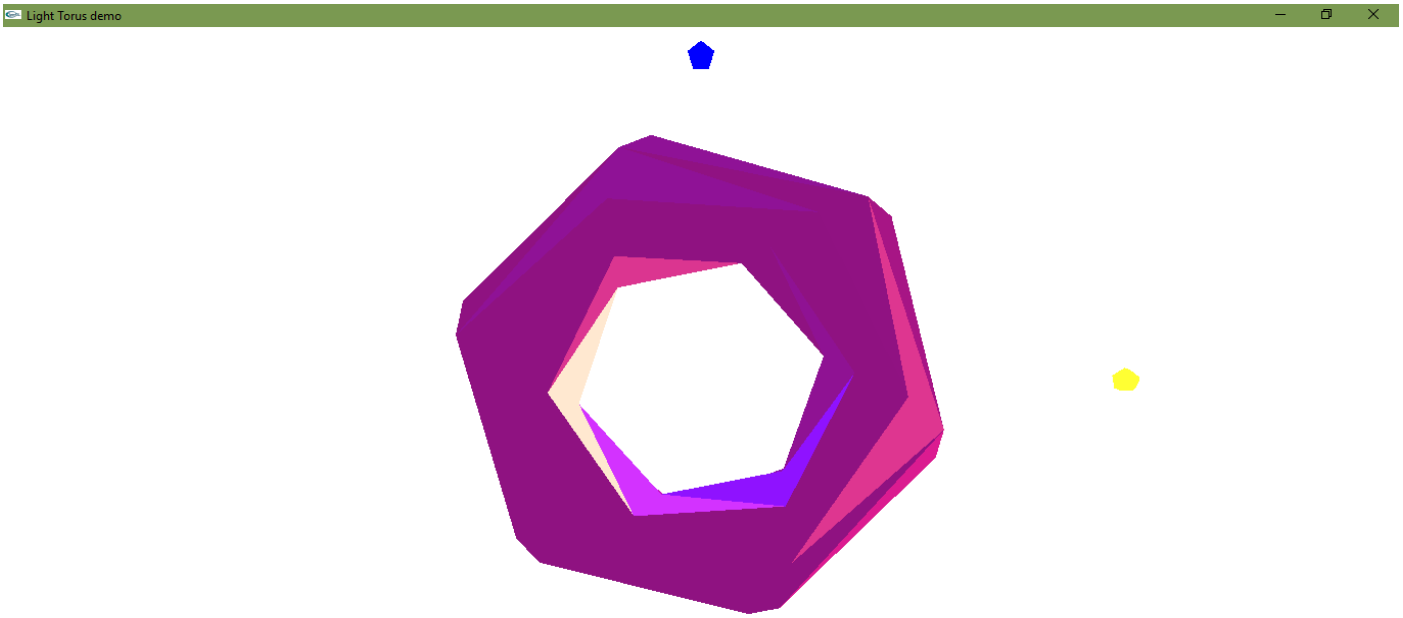

Fig 4.1.3: Less vertices and in Polygon Mode

This is the polygon mode of the light Donut with 6 vertices. It is kept parallel to XY-plane.

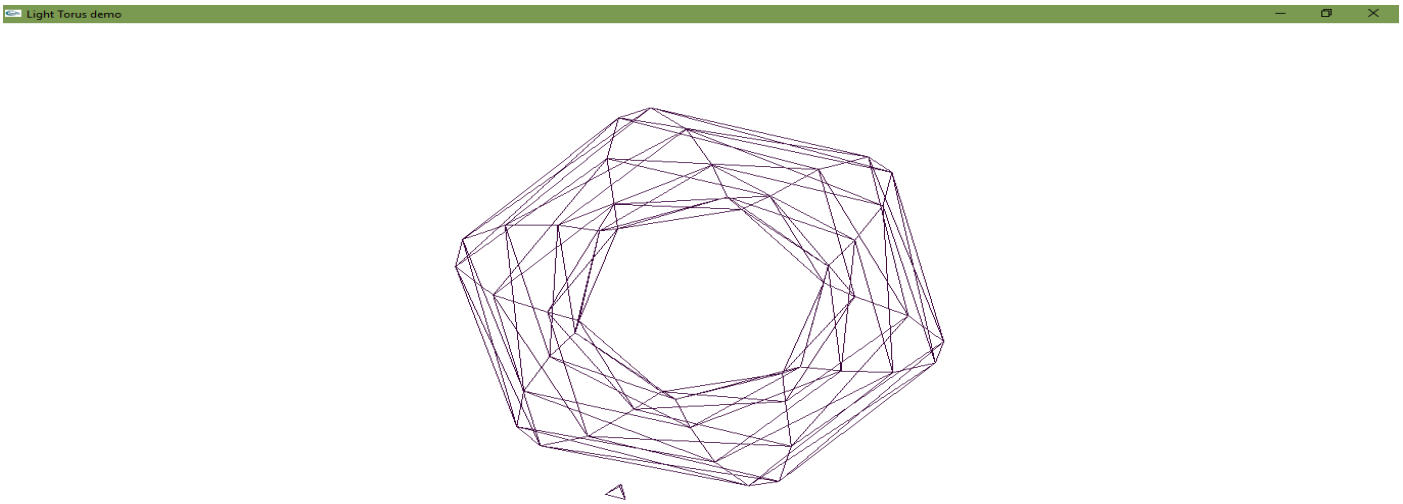

Fig 4.1.4: No lights

In this orientation both the lights are turned off and the Donut is parallel to the XY-plane. The number of vertices is set at 6.

5. CONCLUSION

This project was implemented using OpenGL and it provides us the information about Bump Mapping and 3D Light Animation of Donut Shape. This project also helps us to analyze how we can use the different standard functions for creating the animated objects. This package is very useful for the users since it provides the basic information about various OpenGL functions used for Light Animation of Donut Shape. This project consists one Mini-Project on Light Animation of Donut Shape.

A function is written to switch the mode of the Donut from mesh to a polygon and also to increase and decrease the number of wraps. Coming to the Lighting of the Donut, we have placed two lights - one with Blue color and other with Yellow color on x-axis and y-axis respectively. Hence we animate the Donut with lighting effects using OpenGL library in C programming language.

6. BIBILOGRAPHY

Books referred

- **Edward Angel,** "Interactive Computer graphics A Top Down Approach with OpenGL-,5th Edition", Addison Wesley,2008.

- "OpenGL programming guide". The official guide to learning OpenGL Version 2.1. 6th edition, 2007".