**12. APPENDIX**

**Glut functions**

**1 >>> glutDisplayFunc :-** glutDisplayFunc sets the display callback for the current window.

**Syntax :-**  void glutDisplayFunc(void(\*func)(void));

func:- The new display callback function.

**2 >>> glutReshapeFunc :-** glutReshapeFunc sets the reshape callback for the current window.

**Syntax :-** void glutReshapeFunc(void(\*func)(int width,int height));

func:- The new reshape callback function.

**3 >>> glutMouseFunc :-** glutMouseFunc sets the mouse callback for the current window.

**Syntax :-** void glutMouseFunc(void(\*func)(int button,int state,int x,int y));

func:- The new mouse callback function.

**4 >>> glutKeyboardFunc :-** glutKeyboardFunc sets the keyboard callback for the current window.

**Syntax :-** void glutKeyboardFunc(void(\*func)(unsigned char key,int x,int y));

func:- The new Keyboard callback function.

**5 >>> glutCreateWindow :-** glutCreateWindow creates a top-level window.

**Syntax :-** int glutCreateWindow(char \*name);

name:- ASCII character string for use as window name.

**6 >>> glutInit :-** glutInit is used to initalize the GLUT library.

**Syntax :-** void glutInit(int \*argc,char \*\*argv);

argc:- A pointer to the program's unmodified argc variable from main.Upon return,the value pointed to by argc will be updated because glutInit extracts any command line options intented for the GLUT library.

argv:- The program's unmodified argv variable from main.Like argc,the data for argv will be updated because glutInit extracts any command line options understood by the GLUT library.

**7 >>> glutInitWindowPosition ,glutInitWindowSize :-** glutInitWindowPosition and glutInitWindowSize set the initial window position and size respectively.

**Syntax :-** void glutInitWindowSize(int width,int height);

void glutInitWindowPosition(int x,int y);

width: Width in pixels.

height: Height in pixels.

x: Window x location in pixels.

y: Window y location in pixels

**8 >>> glutInitDisplayMode :-** glutInitDisplayMode sets the initial display mode.

**Syntax :-**  void glutinitDisplayMode(unsigned int mode);

mode:- Display mode,normally the bitwise OR-ing of GLUT display mode bit masks.

**9 >>> glutMainLoop :-** glutMainLoop enters the GLUT event processing loop.

**Syntax :-** void glutMainLoop(void);

**10 >>> glutPostRedisplay :-** glutPostRedisplay marks the current window as rendering to be redisplayed.

**Syntax :-** void glutPostRedisplay(void);

**11 >>> glutBitmapCharacter :-** glutBitmapCharacter renders a bitmap character using OpenGL.

**Syntax :-** void glutBitmapCharacter(void \*font,int charcter);

**12 >>> glEnable :-**

**Syntax :-** void glEnable(GLenum cap);

cap:- specifies a symbolic constant indicating a GL capability.

**13 >>> glBegin and glEnd :-** The glBegin and glEnd functions delimit the vertices of a primitive or a group of like primitives.

**Syntax :-** void glBegin(GLenum mode);

mode:- The primitives that will be created from vertices presented between glBegin and the subsequent glEnd.

**14 >>> glClear :-** The glClear function clears buffers to present values.

**Syntax :-** void glClear(GLbitfield mask);

mask:- Bitwise OR operators of masks that indicate the buffers to be cleared.

**15 >>> glClearColor :-** The glClearColor unction specifies clear values for the color buffers.

**Syntax :-** void glClearColor(red,green,blue,aplha);

red:- The red value that glClear uses to clear the color buffers.The default value is zero;

green:- The green value that glClear uses to clear the color buffers.The default value is zero;

blue:- The blue value that glClear uses to clear the color buffers.The default value is zero;

alpha:- The alpha value that glClear uses to clear the color buffers.The default value is zero;

**16 >>> glColor3i :-** Sets the color.

**Syntax :-** void glColor3i(GLint red,GLint green,GLint blue);

red:- The new red value for the current value .

green:- The new green value for the current value .

blue:- The new blue value for the current value .

**17 >>> glColor3fv :-** Sets the current color from an already existing array of colors.

**Syntax :-** void glColor3fv(const GLfloat \*v);

v:- A pointer to an array that contains red,green,blue values.

**18 >>> glFlush :-** The glFlush function forces execution of OpenGL functions in finite time.

**Syntax :-** void glFlush(void);

This function has no parameters.

**19 >>> glLoadIdentity :-** The glLoadIdentity function replaces the current matrix with the identity matrix.

**20 >>> glOrtho :-** The glOrtho function multiplies the current matrix by an orthographic matrix.

**Syntax :-** void glOrtho(GLDouble left, GLDouble right, GLDouble bottom, GLDouble top, GLDouble zNear, GLDouble zFar);

left:- The coordinates of the left vertical clipping plane.

right:- The coordinates of the right vertical clipping plane.

bottom:- The coordinates of the bottom horizontal clipping plane.

top:- The coordinates of the top horizontal clipping plane.

zNear:- The distances to the nearer depth clipping plane.This distance is negative if the plane is to be behind the viewer.

zFar:- The distances to the farther depth clipping plane.This distance is negative if the plane is to be behind the viewer.

**21 >>> glPointSize :-** The glPointSize function specifies the diameter of rasterized points.

**Syntax :-** void glPointSize(GLfloat size);

Size:-The diameter of rasterized points.The default is 1.0.

**22 >>> glPushMatrix and glPopMatrix :-** The glPushMatrix and glPopMatrix functions push and pop the current matrix stack.

**Syntax :-** void WINAPI glPopMatrix(void);

**23 >>> glRotatef :-** The glRotatef function multiplies the current matrix by a rotation matrix.

**Syntax :-** void glRotatef(GLfloat angle,GLfloat x,GLfloat y,GLfloat z);

angle:-The angle of rotation.

x:-The x coordinate of vector.

y:-The y coordinate of vector.

z:-The z coordinate of vector.

**24 >>> glScalef :-** The glScaled and glScalef functions multiplies the current matrix by a general scaling matrix.

**Syntax :-** void glScalef(GLfloat x,GLfloat y,GLfloat z);

x:-Scale factors along x axis.

y:-Scale factors along y axis.

z:-Scale factors along z axis.

**25 >>> glTranslatef :-** The glTranslatef function multiplies the current matrix by a translation matrix.

**Syntax :-** void glTranslatef(GLfloat x,GLfloat y,GLfloat z);

x:- the x coordinate of a translation vector.

y:- the y coordinate of a translation vector.

z:- the z coordinate of a translation vector.

**26 >>> glVertex2d :-** Specifies a vector.

**Syntax :-** void glVertex2d(GLdouble x,GLdouble y);

x:- Specifies the x-coordinate of a vertex.

y:- Specifies the y-coordinate of a vertex.