**3. API’S USED**

1. **glBegin (GLenum mode);**

It defines the type of primitives that the vertices define. Each subsequent execution og glVertex3f pecifies the x,y,z coordinates of a location in space.

1. **glEnd (void);**

It ends the list of vertices.

1. **glClear (GLbitfield mask);**

It is used to make the screen solid and white.

1. **glClearColor (GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha);**

The background color is set with this API where the last argument specifies a degree of transparency.

1. **glColor3f (GLfloat red, GLfloat green, GLfloat blue);**

It sets color to current drawing.

1. **glEnable (GLenum cap);**

We enable different algorithms by this function call.

1. **glDisable (GLenum cap);**

We disable different algorithms by this function call.

1. **glLoadIdentity (void);**

To initialize the current transform matrix to the identity transform.

1. **glMatrixMode (GLenum mode);**

It switches matrix mode between the two matrices –

MODEL\_VIEW (GL\_MODELVIEW) & PROJECTION (GL\_PROJECTION)

1. **glPushMatrix (void);**

It is used to push matrix on the stack.

1. **glPopMatrix (void);**

It is used to pop matrix from the stack.

1. **glRasterPos3f (GLfloat x, GLfloat y, GLfloat z);**

It is used to alter the shape, size, position of pixels.

1. **glTranslatef (GLfloat x, GLfloat y, GLfloat z);**

It is used to alter the model view or projection matrices.

1. **glVertex3f (GLfloat x, GLfloat y, GLfloat z);**

It is used to represent vertex.

1. **glViewport (GLint x, GLint y, GLsizei width, GLsizei height);**

It specifies that the viewport will have lower left corner(x,y) in screen co-ordinates and will be width pixels wide and height pixels high.

1. **glutBitmapCharacter(void \*font, int character);**

The character is placed at the present raster position on the display, is measured in pixels and can be altered by the various forms of the function glRasterPos\*.

1. **glutCreateWindow(const char \*title);**

It creates and opens OpenGL window.

1. **glutDisplayFunc(void (GLUTCALLBACK \*func)(void));**

It sends graphics to screen.

1. **glutInit(int \*argcp, char \*\*argv);**

It initiates interaction between windowing system and OpenGL.

1. **glutInitDisplayMode(unsigned int mode);**

This function specifies how the display should be initialized. The constants GLUT\_SINGLE and GLUT\_RGB, which are ORed together, indicate that a single display buffer should be allocated and the colors are specified using desired amount of red, green and blue.

1. **glutInitWindowSize(int width, int height);**

It sets the size of created window.

1. **glutKeyboardFunc(void (GLUTCALLBACK \*func)(unsigned char key, int x, int y));**

The keyboard event is generated when the mouse is in the window and one of the key is pressed or released. This GLUT function is the call bach for event generated by pressing a key.

1. **glutMainLoop(void);**

It causes the program to begin an event-processing loop.

1. **glutReshapeFunc(void (GLUTCALLBACK \*func)(int width, int height));**

The reshape event is generated whenever the window is resized, such as by a user interaction.

1. **glutSwapBuffers(void);**

We can swap the front and back buffers at will from the application programs.

1. **glBindTexture(GLenum target, GLuint texture);**

We can bind a named texture to a texturing target.

1. **glTexParameteri(GLenum target, GLenum pname, GLint param);**

We can assign a value or a set of values to the texture parameter.

1. **glScalef(GLfloat x,GLfloat y,GLfloat z);**

We can produce a non-uniform scaling along the x, y, and z axes. The three parameters indicate the desired scale factor along each of the three axes.

1. **glutMouseFunc(void (\*func)(int button, int state,int x, int y));**

We can set the mouse callback for the current window. When a user presses and releases mouse buttons in the window, each press and each release generates a mouse callback.

1. **gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top);**

We can set up a two-dimensional orthographic viewing region.