**3. Detailed design**

**3.1 Window design:**

Graphics editor uses only one window for all its purposes. The part of the window is used as canvas for drawing , container for holding many tools like color palette, writing tools like pencil, brush etc…

One of the most important things is that window is portable and flexible. The positioning of controls inside the window is generalized. The controls are placed with respect to the relative values of the window width and height. This means that the editor is portable to any system having different resolutions.

**3.2 Main Algorithm:**

The main algorithm for the graphics editor is the one shown below.

1. Initiate the Opengl graphics mode.

2. Initiate the mouse interface and keyboard interfaces.

3. Draw the toolbar icons, menus, and the main window.

4. Perform any drawing or editing on the canvas using any of the tools.

5. Close the application after releasing any dynamically used resources are released.

As we see above specific functions are registered for mouse and keyboard events. These functions are generally call back functions which are called upon receiving specific events from mouse and keyboard. Upon receiving inputs from these functions , specific actions are performed based on the mouse position.

**Data Structures Used:**

There are no other data structures explicitly used. This is mainly because, the editor is designed for area based operations and not object-based operations. So it is not necessary maintain information about objects drawn on the drawing area.

**3.3 Other Algorithms used**

Various algorithms have been used in this editor to provide the functionalities it boasts of. Few of them have been explained here.

**3.3.1 Free- hand Drawing:**

Free- hand drawing is a special case of polyline drawing where we have a large number of points obtained by continuously poling the mouse position during the time the mouse' left button is clicked.

**3.3.2 Translation:**

Translation is done by adding the required amount of translation quantities to each of the points of the objects in the selected area. If P(x,y) be the a point and (tx, ty) translation quantities then the translated point is given by

P'(x,y) = p(x+tx,y+ty)

**3.3.3 Scaling:**

The scaling operation on an object can be carried out for an object by multiplying each of the points (x,y) by the scaling factors sx, sy.

Newx = oldx\*sx

Newy = oldy\*sy