# Introduction

## Computer Graphics

Computer graphics is concerned with all aspect of producing pictures or image using computer. Graphics is created using [computers and](http://en.wikipedia.org/wiki/Computer), more generally, the [representation and](http://en.wikipedia.org/wiki/Representation) [manipulation o](http://en.wikipedia.org/wiki/Manipulation)[f pictor](http://en.wikipedia.org/wiki/Pictorial)[ial data by](http://en.wikipedia.org/wiki/Data) a computer. The development of computer graphics has made computers easier to interact with and better for understanding and interpreting many types of data.

The graphics in openGL provides a wide variety of built-in function. It has become a common element in user interface, data visualization, TV commercials, motion picture and many other applications. The current trend of computer graphics is to incorporate more physics principles into 3D graphics algorithm to better simulate the complex interactions between objects and lighting environment.

## OpenGl (Open Graphics Library):

OpenGL has become a widely accepted standard for developing graphics application. The interface between the application program and the graphics system can be specified through that set of function that resides in graphics library. The specification is called the APPLICATION PROGRAM INTERFACE (API). The application program sees only the API and is thus shielded from the details both the hardware and software implementation of graphics library. The software driver is responsible for interpreting the output of an API and converting these data to a form that is understood by the particular hardware.

Most of our applications will be designed to access openGL directly through functions in three libraries. Function in the main GL library have name that begin with the letter gl and stored in the library. The second is the openGL utility Library (GLU). This library uses only GL function but contains codes for creating common object and viewing. Rather then using an different library for each system we used available library called openGL utility toolkit (GLUT). It used as #include<glut.h>

Graphics Editors can normally be classified as:

(a) 2D Graphics Editors.

(b) 3D Graphics Editors.

A 3D Graphics Editor is used to draw 3D primitives Rectangles, Circle, polygons, etc and alter those with operations like cut, copy, paste. These may also contain features like layers and object precision etc. 3D Graphics Editor should include the following features:

1) Facilities: Cursor Movement, Editing picture objects.

2) Good User Interface: GUI / Toolbars / Icon based User Interface.

Computer Graphics is concerned with all aspects of producing pictures or images using a computer. A particular graphics software system called OpenGL, which has become a widely accepted standard for developing graphics applications .

The applications of computer graphics in some of the major areas are as follows

1) Display of information.

2) Design.

3) Simulation and Animation.

4) User interfaces.

## Problem Section Statement

I was assigned to design a flowing fountain that contain below requirements :

1. A colorful fountain with many beautiful colors.

2. The fountain will flow it’s water.

3. Water speed will be controlled.

4. A beautiful view will be presented of this flowing fountain.

## Objectives of The Project

1. Developing a package using computer graphics with OpenGL.

2. Migration from text editor to OpenGL.

3. To show that implementation of Translation is easier with OpenGL.

4. Implementing certain technical concept like Translation, motion, and use of Idle Function.

5. How to use Lightning effects used to produce computer animation.

**Description of The Project**

**1)** The "FLOWING FOUNTAIN MODEL". It depicts a 3 Dimensional model of a fountain through which water is continuously flowing out in its idle state. The water gets stored into a small reservoir. The water flows out through different levels in the fountain, giving it a realistic look.

**2)** User can specify these levels as three, four or five at the beginning.

**3)** The program starts with a menu on the screen giving you the options : Proceed, Help, Exit.

**4)** The user is provided with an option to change the color of the fountain using the RIGHT MOUSE BUTTON. The user can view the fountain from different angles including a Top-view and can also zoom in or zoom out. This can be controlled using a set of specified keys on keyboard such as ‘N’ and ‘A’ for ZOOM IN and ZOOM OUT, buttons ‘T’ and ‘F’ for TOP and FRONT VIEW etc. Clicking on the RIGHT MOUSE BUTTON shows a sub menu 'Help' which displays the keyboard shortcuts for various controls. The third option ‘Exit’ pops out of the program.

**5)** The project is based on Simple window coordinates and using recursive techniques in OpenGL.

# Software Design

## Movement of a drop:

1. The movement of a drop contains two factors.
2. The direction, how the drop gets out of the fountain and the gravity. The position of a drop is pretty easy to compute if we know, how much time has passed since the drop has leaved the fountain.
3. We have to multiply the vector of the constant moving (how the drop leaves the fountain) with the time and then subtract the squared time multiplied with an acceleration factor. This acceleration factor contains the weight of a drop and the power of gravity. We now have to know the direction, how the drop comes out of the fountain, but this is just a bit calculating with sine and cosine.
4. Blending means that a pixel on the screen isn't replaced by another one, but they are "mixed". Therefore you can use the alpha value of colors, it indicates how much of the color of the consisting pixel is used for the new color - for antialiasing of points, OpenGL computes this alpha value.
5. After calling *glEnable(GL\_BLEND);* you have to tell OpenGL how to use the alpha values. It isn't specified, that a higher alpha-value means more transparency or something like that. You can use them as you want. To tell OpenGL \_what\_ you want, you must use *glBlendFunc().* It takes two parameters, one for the source factor and the second for the destination factor. I used
6. GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA as parameters. This is quite an often used combination and affects, that the higher the alpha value, the less transparency of the incoming fragment .

## Implementation

**Step 1**: [To create a fountain] : Declare a class called CDrop. In GetNewPosition (), I calculate the position and delay of each drop with respect to the coordinate axes.

**Step 2**: [function createlist ()] : Dynamically allocate memory for the required vertices. Function glGenLists used to generate a contiguous set of empty display lists. Then specify a series of ‘for ‘loops to construct the top and bottom of the stone. Then create a qaudrilateral to represent the ground.

To create water, use the following functions: GlTranslatef () – is to calculate water and stone height rand () function is used to generate random unique numbers, for each time it is executed.

**Step 3**: [function InitFountain ()] : Create fountain drops and vertices. Declare StepAngle, the angle which the ray gets out of the fountain and RayAngle, the angle you see when you look down on the fountain. Use sine () and cosine () functions inside for loops, to calculate the speed of each step in the fountain, how many steps are required, that a drop comes out and falls down again.

**Step 4**: [Displaying] : [Keyboard function] - Manages operations by various keys pressed on the key board [Display function] : Renders the program on to the screen.

**Functions:**

GlClear (GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT) : Indicates the buffers currently enabled for color writing and also indicates the depth buffer.

GlPushMatrix (), glPopMatrix () : to push and pop the current matrix stack.

DrawTextXY () : used to set the text of the program.

glFlush () — force execution of GL commands in finite time.

GlutSwapBuffers ()-Swap the buffers ->make the result of rendering visible.

[reshape function] : glMatrixMode (GL\_PROJECTION) -applies subsequent matrix operations to the

projection matrix stack.

**Step 5**: [main function] : Here we specify the initial display mode, window size and position. Create a new window where the output is rendered. Create menus to move near, move away, move down, move up and sub-menus for color, flow, level, and help.

**SOURCE CODE:**

 **OUTPUT:**

Fig 1: Welcome page of the project



Fig 2: Menu page of the project

Fig 3: Help page of the project

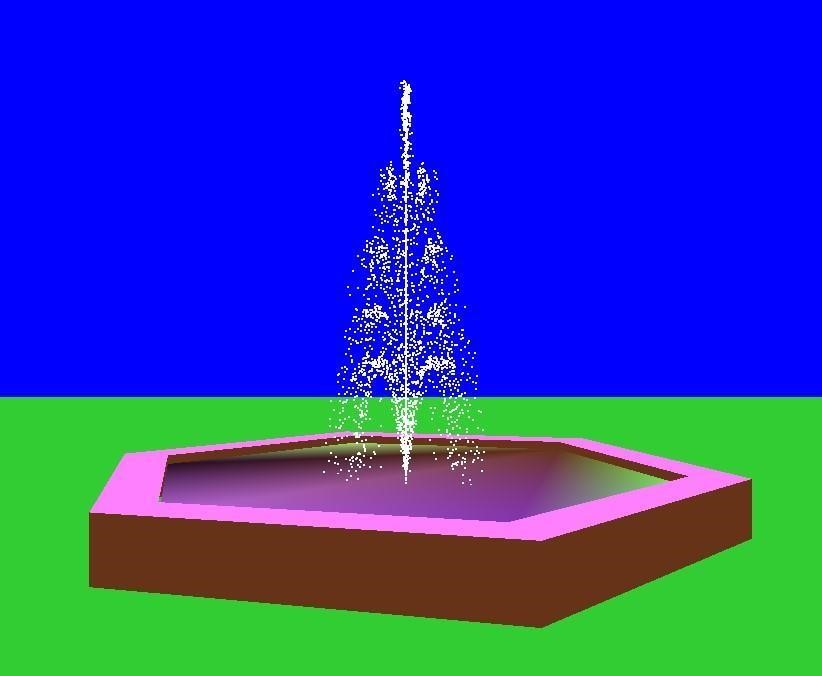


Fig 4: Flowing fountain

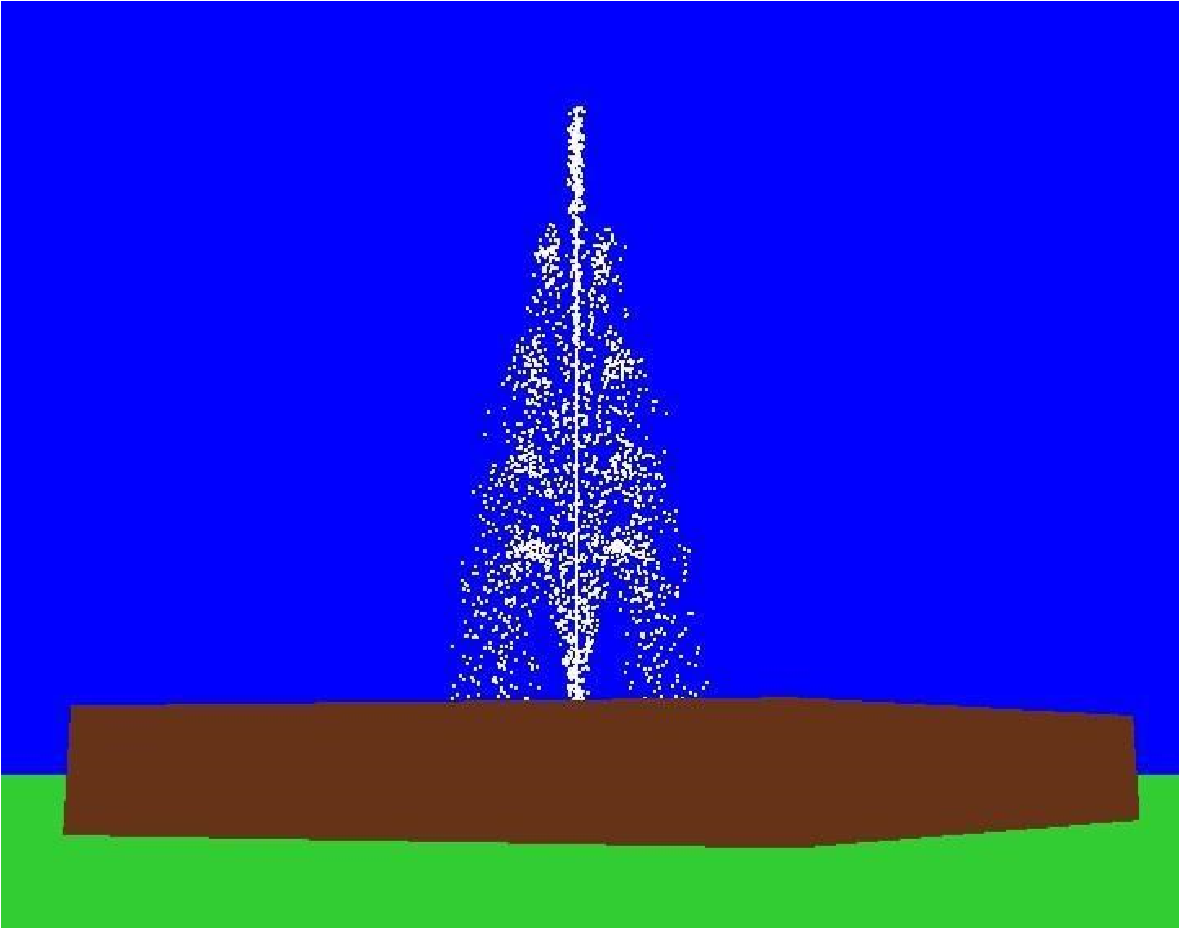


Fig 5: Top view of the flowing fountain Fig 6: Front view of the flowing fountain

**Conclusion**

An attempt has been made to develop an OpenGL package which meets necessary requirements of the user successfully. Since it is user friendly, it enables the user to interact efficiently and easily. The development of the mini project has given us a good exposure to OpenGL by which we have learnt some of the technique which help in development of animated pictures, gaming.

Hence it is helpful for us even to take up this field as our career too and develop some other features in OpenGL and provide as a token of contribution to the graphics world.