**VISVESVARAYA TECHNOLOGICAL UNIVERSITY BELGAUM-590014**



**A Computer Graphics and Visualization**

**Mini-Project Report**

**On**

***“DEPICTION OF SEASONS USING LEAVES”***

*A Mini-project report submitted in partial fulfillment of the requirements for the award of the degree of* ***Bachelor of Engineering in Computer Science and Engineering*** *of Visvesvaraya Technological University, Belgaum.*

Submitted by:

**A.APARNA (1DT14CS001)**

**AND**

**ABHILASH SINGH (1DT14CS003)**

Under the Guidance of:

**Mr. RAGHU.M.T**

**(Asst. Prof. Dept. of CSE)**



**Department of Computer Science and Engineering**

**DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND**

**MANAGEMENT**

Kanakapura Road, Udayapura, Bangalore

2016-2017



**DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND**

**MANAGEMENT**

Kanakapura Road, Udayapura, Bangalore

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CERTIFICATE**

This is to certify that the Mini-Project on Computer Graphics and Visualization work entitled **“DEPICTION OF SEASONS USING LEAVES”** has been successfully carried out by **A.APARNA (1DT14CS001) and ABHILASH SINGH (1DT14CS003) ,** bonafide students of **Dayananda Sagar Academy of Technology and Management** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year 2016-2017. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

**GUIDES:**

**Mr.RAGHU.M.T**

**(Asst. Prof. Dept of CSE)**

***Dr. C. NANDINI (Vice Principal &HOD, Dept. of CSE)***

**Examiners: Signature with Date**

**1:**

**2:**

**ACKNOWLEDGEMENT**

We are very grateful to our institution, **Dayananda Sagar Academy** **of Technology And Management** with its ideals and inspirations for having provided with facilities, which has made this mini project a success.

We are grateful to **Dr. C. NANDINI**, Vice Principal and H.O.D, Department of CSE, Dayananda Sagar Academy of Technology and Management, for her unfailing encouragement and suggestions given to us in course of our mini project work.

We owe wholehearted thanks to **Mr.RAGHU.M.T**, Department of CSE, Dayananda Sagar Academy of Technology and Management, for the valuable guidance, encouragement and suggestions throughout the mini project work and for immense help in making the mini project a successful one.

**A.APARNA(1DT14CS001)**

**ABHILASH SINGH(1DT14CS003)**

**ABSTRACT**

Most modern day calendars around the world divide the year into 4 seasons: spring, summer, fall (autumn), and winter.

The project enables the user to choose one of these four seasons. The seasons are portrayed using a leaf. The different shades of nature and the leaf will show us how each season has its unique characteristics.

The project requires the usage of menus, Therefore the menu function will be implemented.

We can choose from a variety of seasons using the menus option and the leaves appear based on the particular season that has been opted and with their respective characteristic that may include colour, texture, etc of that season.

**TABLE OF CONTENTS**

**CHAPTER1**: **INTRODUCTION**

1.1 INTRODUCTION TO COMPUTER GRAPHICS

1.2 OVERVIEW

1.3 PROBLEM STATEMENT

1.4 OBJECTIVES

1.5 MOTIVATION

1.6 COMPUTER GRAPHICS

1.7 OPENGL

1.8 APPLICATIONS OF COMPUTER GRAPHICS

**CHAPTER 2**: **LITERATURE SURVEY**

2.1 HISTORY OF COMPUTER GRAPHICS

2.2 RELATED WORK

2.3 RELATED DATA STRUCTURE

**CHAPTER 3**: **SYSTEM REQUIREMENTS**

3.1 DESIGN CONSTRAINTS

**CHAPTER 4**: **DESIGN**

4.1 PROPOSED SYSTEM

4.2 DATA FLOW DIAGRAM

4.3 ALGORITHM

4.4 FLOW DIAGRAM

**CHAPTER 5**: **IMPLEMENTATION**

5.1 MODULE DESCRIPTION

5.2 USER DEFINED FUNCTIONS

**CHAPTER 6**: **SNAPSHOTS**

**CHAPTER 7**: **CONCLUSION AND FUTURE ENHANCEMENT**

**BIBILOGRAPHY**

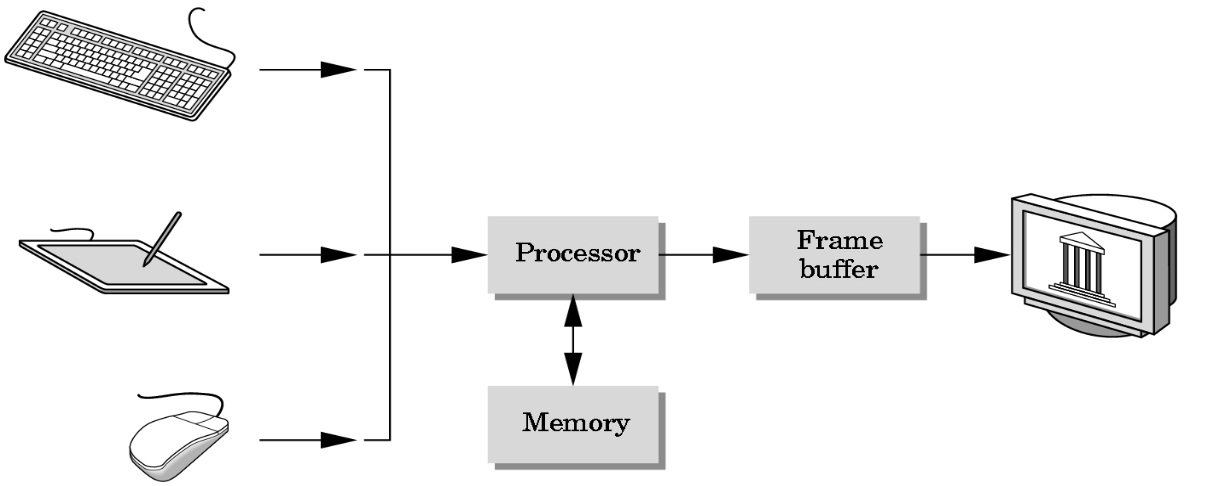
**CHAPTER 1**

**INTRODUCTION**

* 1. **Introduction to Computer Graphics**

Before the invention of computer graphics people used to display the information manually either by drawing or creating model which resembles real environment. For example the Greeks were able to convey their architectural ideas graphically using drawing. Today the same type of information is generated by architects, mechanical engineers and draftsman using computer based graphics system.

Our interaction with computers has become dominated by a visual paradigm that includes windows, icons, menus and a pointing device, such as a mouse. From a user’s Perspective, windowing System such as the X windows system, Microsoft’s Windows. Although we are familiar with the style of graphical user interface used on most workstations, advances in computer graphics have made possible other forms of interfaces.



**Fig 1.1 A Graphics System**

**1.2 OVERVIEW**

Nature truly displays its color palette throughout the 4 seasons (spring, summer, winter and fall). For years, scientists have worked to understand the changes that happen to trees and shrubs in the. Although we don't know all the details, we do know enough to explain the basics and help enjoy more fully Nature's multicolored display. Three factors influence leaf color-leaf pigments, length of night, and weather, but not quite in the way we think. The timing of color change and leaf fall are primarily regulated by the calendar, that is, the increasing length of night. None of the other environmental influences-temperature, rainfall, food supply, and so on-are as unvarying as the steadily increasing length of night during autumn. As days grow shorter, and nights grow longer and cooler, biochemical processes in the leaf begin to paint the landscape with Nature's palette.

**1.3 PROBLEM STATEMENT**

The aim of this project is to develop a 2D Display which supports operations such as Movement, Color change, and also transformation operations like translation, rotation, scaling etc. on objects. The package must also have a user friendly interface.The 2-D graphics package designed here provides an interface for the users for handling the display and manipulation of the color of the leaves . The Mouse is the main input device used.

**1.4 OBJECTIVES**

The objective of the project is to demonstrate the different library functions and effects of graphics using opengl.

* The aim of the project is to demonstrate the change in color of the leaves through different seasons.
* It should be designed in such a way that it provides a very useful graphics implementation interface.
* It should be easy to understand, user interactive interface.
* Creation of primitives, i.e. polygons(leaves).
* Providing human interaction through Mouse .

**1.5 MOTIVATION**

The concept of openGL primitive and openGL transformation functions have motivated the creation of this project. The lab programs were a motivation to bring some animation effects which includes all the functions learnt in computer graphics subject and learn more effects .

**1.6 COMPUTER GRAPHICS**

Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D and 3D pattern recognition abilities allow us to perceive and process pictorial data rapidly and efficiently. Interactive computer graphics is the most important means of producing pictures since the invention of photography and television. It has the added advantage that, with the computer, we can make pictures not only of concrete real world objects but also of abstract, synthetic objects, such as mathematical surfaces and of data that have no inherent geometry, such as survey results.

**1.7 OPENGL**

OpenGL (Open Graphics Library) is a standard specification defining a cross language cross platform API for writing applications that produce 2D and 3D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex 3D scenes from simple primitives. OpenGL was developed by Silicon Graphics Inc. (SGI) in 1992 and is widely used in CAD, virtual reality, scientific visualization, information visualization and flight simulation. It is also used in video games, where it competes with direct 3D on Microsoft Windows Platforms. OpenGL is managed by the nonprofit technology consortium, the Khronos group Inc.

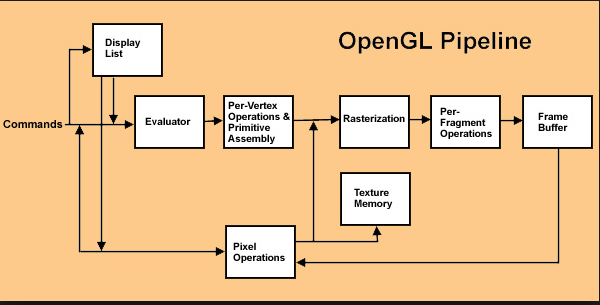
OpenGL serves two main purposes:

* To hide the complexities of interfacing with different 3D accelerators, by presenting programmer with a single, uniform API
* To hide the differing capabilities of hardware platforms, by requiring that all implementations support the full OpenGL feature set.

OpenGL has historically been influential on the development of 3D accelerator, promoting a base level of functionality that is now common in consumer level hardware:

* Rasterized points, lines and polygons are basic primitives.
* A transform and lighting pipeline.
* Z buffering.
* Texture Mapping.
* Alpha Blending.

**1.7.1 OPENGL GRAPHICS ARCHITECTURE:**



**FIG 1.1 OPENGLGRAPHICS ARCHITECTURE**

The OpenGL architecture:

* **Display Lists :**

All data, whether it describes geometry or pixels, can be saved in a display list for current or later use. When a display list is executed, the retained data is sent from the display list just as if it were sent by the application in immediate mode.

* **Evaluators :**

All geometric primitives are eventually described by vertices. Parametric curves and surfaces may be initially described by control points and polynomial functions called basis functions.

* **Per Vertex Operations :**

For vertex data, next is the “per-vertex operations” stage, which converts the vertices into primitives. Some vertex data are transformed by 4 x 4 floating-point matrices. Spatial coordinates are projected from a position in the 3D world to a position on your screen.

* **Primitive Assembly :**

Clipping, a major part of primitive assembly, is the elimination of portions of geometry which fall outside a half space, defined by a plane.

* **Pixel Operation:**

While geometric data takes one path through the OpenGL rendering pipeline, pixel data takes a different route. Pixels from an array in system memory are first unpacked from one of a variety of formats into the proper number of components. Next the data is scaled, biased, and processed by a pixel map. The results are clamped and then either written into texture memory or sent to the rasterization step.

* **Rasterization:**

Rasterization is the conversion of both geometric and pixel data into fragments. Each fragment square corresponds to a pixel in the frame buffer. Color and depth values are assigned for each fragment square.

* **Fragment Operations :**

Before values are actually stored into the frame buffer, a series of operations are performed that may alter or even throw out fragments. All these operations can be enabled or disabled.

**1.8 APPLICATIONS OF COMPUTER GRAPHICS:**

* **USER INTERFACE:**

It is now a well-established fact that graphical interfaces provide an alternative and easy inter action between users and computers the built in graphics provided with user interfaces use the control items.

In industry, business government and education organization’s computer graphics ismost commonly used to create 2D and 3D graphs of mathematical, physical and economicfunctions in the form of histograms, bars and pie charts which are very useful in decisionmaking.

* **COMPUTER AIDED DRAFTING AND DESIGN:**

The computer aided drafting uses the graphics components and systems. Electrical, electronic and mechanical devices such as automobile bodies, structure of airplanes, ships and buildings use this concept.

* **SIMULATION AND ANIMATION FOR SCIENTIFIC VISULIZATION AND ENVIRONMENT:**

Use of graphics in simulation makes mathematical models and mechanical systems more realistic and easy to study. The interactive graphics supported by animation software proved their use in production of animated movies and cartoon films.

**CHAPTER 2**

**LITERATURE SURVEY**

During the spring and summer the leaves have served as factories where most of the foods necessary for the tree's growth are manufactured. This food-making process takes place in the leaf in numerous cells containing chlorophyll, which gives the leaf its green color. This extraordinary chemical absorbs from sunlight the energy that is used in transforming carbon dioxide and water to carbohydrates, such as sugars and starch.

Along with the green pigment are yellow to orange pigments, carotenes and xanthophyll pigments which, for example, give the orange color to a carrot. Most of the year these colors are masked by great amounts of green coloring.

But in the fall, because of changes in the length of daylight and changes in temperature, the leaves stop their food-making process. The chlorophyll breaks down, the green color disappears, and the yellow to orange colors become visible and give the leaves part of their fall splendor.

At the same time other chemical changes may occur, which form additional colors through the development of red anthocyanin pigments. Some mixtures give rise to the reddish and purplish fall colors of trees such as dogwoods and sumacs, while others give the sugar maple its brilliant orange.

The autumn foliage of some trees show only yellow colors. Others, like many oaks, display mostly browns. All these colors are due to the mixing of varying amounts of the chlorophyll residue and other pigments in the leaf during the fall season.

A color palette needs pigments, and there are three types that are involved in leaf color:

* Chlorophyll, which gives leaves their basic green color. It is necessary for photosynthesis, the chemical reaction that enables plants to use sunlight to manufacture sugars for their food. Trees in the temperate zones store these sugars for their winter dormant period.
* Carotenoids, which produce yellow, orange, and brown colors in such things as corn, carrots, and daffodils, as well as rutabagas, buttercups, and bananas.
* Anthocyanins, which give color to such familiar things as cranberries, red apples, concord grapes, blueberries, cherries, strawberries, and plums. They are water soluble and appear in the watery liquid of leaf cells.

**2.1 HISTORY OF COMPUTER GRAPHICS**

In the late 1960’s the development of Free-form curves and surfaces for computer graphics begins. Free form curves and surfaces were developed to describe curved 3D objects without using polyhedral representations which are bulky and intractable. To get a precise curve with polygons might require thousands of faces, whereas curved surfaces requires much less calculations. The UNISURF CAD system was created for designing cars which utilized the curve theories.

The rapid increase in the performance of graphics hardware, coupled with recent improvements in its programmability, have made graphics hardware a compelling platform for computationally demanding tasks in a wide variety of application domains. In this report, we describe, summarize, and analyze the latest research in mapping general-purpose computation to graphics hardware.

We begin with the technical motivations that underlie general-purpose computation on graphics processors (GPGPU) and describe the hardware and software developments that have led to the recent interest in this field. We then aim the main body of this report at two separate audiences. First, we describe the techniques used in mapping general-purpose computation to graphics hardware. We believe these techniques will be generally useful for researchers who plan to develop the next generation of GPGPU algorithms and techniques. Second, we survey and categorize the latest developments in general-purpose application development on graphics hardware.

**2.2RELATED WORK**

The user controls the color of the leaves by selecting a season from the options in the displayed menu. According to the season the falling leaves change into the assigned color.

The colors assigned for the seasons are:

* Spring: Green
* Summer: Yellow
* Winter: Brown
* Fall: Red

If the package is being used from the designing end, then the user may also control the number of the leaves, the rotation speed, the speed at which the leaves are falling and other such functions.

**2.3 RELATED DATA STRUCTURE**

Package uses data structures like Stack for storing the co-ordinates of primitives and also uses arrays for storing the rgb values of the colors.

**CHAPTER 3**

**SYSTEM REQUIREMENTS**

**3.1 DESIGN CONSTRAINTS**

**SOFTWARE CONSTRAINTS**:

* **OS:** Windows XP/7/8.
* **Language Tool:** Win32 API for interface and Open GL for the functionalities.
* **Compiler:** GNU GCC Compiler/C++ Compiler.
* **Library Files:** glut32.h, opengl32.h & glu32.h.
* **Software Used** Visual studio 2015.

**HARDWARE CONSTRAINTS:**

* **Processor:** Intel Pentium 150MHz onwards Compatible.
* **RAM:** 1GB RAM.
* **HDD:** 1GB
* **Monitor:** 16bit color monitor.

**CHAPTER 4**

**DESIGN**

**4.1 PROPOSED SYSTEM**

To achieve three dimensional effects, OpenGL software is proposed. It is a software which provides a graphical interface. It is an interface between application program and graphics hardware. The advantages are:

* OpenGL is designed as a streamlined**.**
* It is a hardware independent interface, it can be implemented on

many different hardware platforms.

* With OpenGL, we can draw a small set of geometric primitives such as

Points, lines and polygons etc.

* It provides double buffering which is vital in providing transformations.
  1. **DATA FLOW DIAGRAM**

Transformed graphical output (color)

Input from user

**FIG 4.1 DATA FLOW DIAGRAM FOR INITIAL MODEL**

**4.3 ALGORITHM**

Step1: Initialize the graphics window and its size using GLUT functions.

Step 2: Register the menu and display call backs in main function.

Step3: When right key is pressed hierarchical menu appears on window.

Step 4: If Spring is selected then leaves change remain green as default color is green.

Step 5: If Summer is selected then leaves change to yellow.

Step 6: If Winter is selected then leaves change to brown.

Step 7: If Fall is selected then leaves change to red.

Step 8: If Quit option is selected program is terminated.

**4.4 FLOW DIAGRAM**

MENU( )

DISPLAY( )

LEAF( )

1. SPRING( )

2. SUMMER( )

3. WINTER( )

4. FALL( )

5. QUIT( )

MENU( )

**FIG 4.2 FLOW DIAGRAM**

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 MODULE DESCRIPTION**

The program is divided into four classes:

* **MODULE1: DISPLAY**

This function deals with the displaying of the coordinate axes and number of mentioned leaves. It also deals with the transformation of the current color to the required weather color until the specified weather color. This class also controls the speed at which the leaf falls which depends on its size.

* **MODULE2: LEAF**

This function uses the translate, rotate, and scale function to display the leaf at the mentioned position. It uses various vertex functions to display the stem , leaf and veins of the leaf.

* **MODULE3: MENU**

This class describes the menu to be displayed which provides options to choose or select a particular season. It also provides option to quit or terminate the program.

* **MODULE4:MAIN**

The main function uses all these functions and through the help of menu and mouse interactions lets the users change the color of the leaf. The display stops when the user selects the quit option.

**5.2 USER DEFINED FUNCTIONS**:

* **FUNCTION FOR DISPLAYING THE LEAF:**

display ( )

{

Both the x and y coordinate axes are drawn. The transformation of leaves’ color is controlled using a for loop. Speed at which leaf falls is monitored using the the y coordinate value and the size of the leaf

}

* **FUNCTION FOR DRAWING THE LEAF:**

Leaf( int x, int y, float size, float angle )

{

Here the translate, rotate and scaling functions are used to draw the leaf at the mentioned position, also the stem, leaf and the veins are drawn using vertex functions. Parameters passed are the values of x and y coordinates and the size and angle of rotation of the leaf.

}

* **FUNCTION FOR DISPLAYING THE MENU :**

menu( int op)

{

Here, the code for displaying the menu is described. Menu to select different seasons is described also code for quit option is written all using a switch statement and providing different cases for each season and quit option. Parameter passed is the option chosen.

}

Spring( )

{

Specifies weather color values for green color.

}

Summer( )

{

Specifies weather color values for yellow color.

}

Winter( )

{

Specifies weather color values for brown color.

}

Fall( )

{

Specifies weather color values for red color.

}

**5.2.1 IN BUILT FUNCTIONS USED:**

* **RAND**

**Syntax:**

rand( );

**Description:**

Generates a random value, in this case for the x coordinate values.

* **SLEEP**

**Syntax:**

Sleep( );

**Parameters:**

Milliseconds for which the program must sleep.

**Description**:

Specifies time for which program must sleep.

* **TRANSLATE FUNCTION**

**Syntax:**

glTranslatef( x,y,z);

**Parameters:**

x, y and z values

**Description:**

Accepts axis along which it must translate the object.

* **ROTATE FUNCTION**

**Syntax:**

glRotatef( angle,x,y,z);

**Parameters:**

Angle of rotation, x, y and z values.

**Description:**

Accepts the angle of rotation and x, y and z values.

* **SCALE FUNCTION**

**Syntax:**

glScalef(size,size,size);

**Parameters:**

Sizes of scaling in all three axes.

**Description:**

Acceptsscaling factor for all three axes.

* **POST REDISPLAY**

**Syntax:**

Void glutPostRedisplay( );

**Description**:

glutPostRedisplay marks the normal plane of current window as needing to be redisplayed. glutPostRedisplay may be called within a window's display or overlay display callback to re-mark that window for redisplay.

* **COLOR FUNCTION**

**Syntax :**

glColor3f(red, green , blue);

**Parameters :**

red *:* The new red value for the current color.

green *:*he new green value for the current color.

blue*:* The new blue value for the current color.

**Description:**

This function accepts different colors.

**CHAPTER 6**

**SNAPSHOTS**

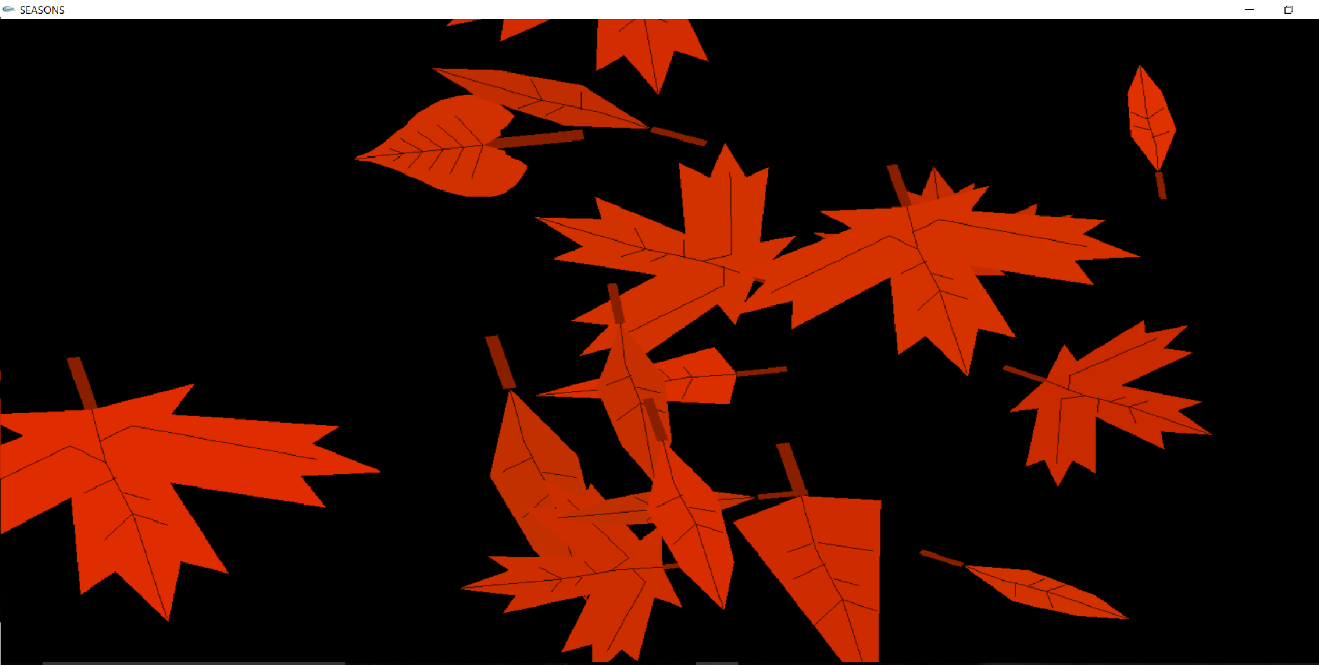
****

**FIG.6.1 SPRING**

**FIG 6.2 SUMMER**

****

**FIG 6.3 WINTER**

****

**FIG 6.4 FALL**

****

**FIG 6.5 MENU DISPLAY**

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENTS**

The project efficiently and correctly been built and the previously existing errors have been rectified and corrected. The required functions stated earlier have been implemented. After all testing process, the program is now ready to be used. We can use the menus efficiently to change the season with respect to our needs and adjust it. Further edits can also be done in case we want some changes.

In future the following enhancements could be done:

* Providing Background colors.
* Providing High Quality Graphics.

**BIBLIOGRAPHY**

[1] The Red Book-OpenGL programming Guide,6th edition

[2] Edward Angel Interactive Computer Graphics A Top-Down Approach with OpenGL, 5th edition, Addison and Wesley

[3] Donald Hearn and Pauline Baker: Computer Graphics- OpenGl version, 3rdedition, Pearson Education

[4] F.S. Hill Jr.: Computer Graphics Using OpenGL,3rdedition, PHI.

Websites:

* <http://www.opengl.org/registry/>
* [www.na.fs.fed.us](http://www.na.fs.fed.us)
* [www.openglforum.com](http://www.openglforum.com)
* <http://www.esf.edu>
* [www.vtucs.com](http://www.vtucs.com)