**Chapter 1**

**INTRODUCTION**

**1.1 About Computer Graphics**

Computer Graphics is concerned with all aspects of producing pictures or images using a computer. We can create images by computers that are indistinguishable from photographs of real objects. There is a common saying that “a picture is worth a thousand words”. We can picturise all the real world objects using computer graphics. We can model all real world objects, render them, give various material properties for them, and create animations using computer graphics. Computer graphics today is largely interactive: the user controls the contents, structure and appearance of the objects and their displayed images by using input devices such as keyboard, mouse. User can select command by pull down menu, pop-up screen etc. There are many graphics software and APIs and one such API is OpenGL.

**1.2 About OpenGL**

OpenGL is graphics software system which has become widely accepted for developing graphics application. Open GL enables top down approach. It provides ease to the programmer as most of the functions are inbuilt. OpenGL has most widely used and supported 2D and 3D graphics application programming interface (API). OpenGL is easy to learn and it possesses most of the characteristics of other popular graphical systems. The libraries provided by OpenGL interface are OpenGL core, OpenGL utility library, OpenGL utility toolkit. We want to start writing, as quickly as possible, application program that will generate graphical output.

OpenGL can be accessed using GL library utility library(GLU),OpenGL utility toolkit(GLUT).GL library have names that begin with the letters gl. The GLU library is available in all OpenGL implementations. Functions in the GLU library begin with the letters glut. GLUT will use GLX and the X libraries. The application program can use only GLUT functions.

OpenGL offers a rich and highly usable API for 2D and 3D graphics and image manipulations .OpenGL makes heavy use of macros to increase code readability and avoid the use of magic numbers. The OpenGL way of creating graphics is used widely in both academics and industries.

**1.3 About Project**

This project is designed and implemented using OpenGL interactive application that basically deals with providing the graphical interfaces between user and system. The mini project “Life Cycle of Frog” is the course of stages of the frog. It involves several stages such as, the eggs hatch in 6 days, the tadpoles grow hind legs after 5 weeks, the tail starts to disappear as the legs grow, after 11 weeks the frog is fully grown and its life span is around 10-12 years, the adult frog lay eggs. These are the stages involved in our project.

**Chapter 2**

**SYSTEM REQUIREMENT AND SPECIFICATIONS**

The system requirement and specification of our project is as follows:

**2.1 Hardware Requirements**

Processor : Intel Pentium 4

Monitor

Main memory : 1 GB RAM

CPU Speed : 2.0 GHz

Key Board : Standard

Mouse : Standard

**2.2 Software Requirements**

Operating system : Ubuntu 14.04

Vi editor

OpenGL library

Glut Library

**2.3 Details of the Software**

Here, the coding of our project is done in OpenGL (Open Graphics Library) which is a standard specification to produce 2D and 3D computer graphics. We use, the OpenGL Utility Toolkit called GLUT which is a library of utilities for OpenGL programs.

**2.3.1 OpenGL and GLUT**

OpenGL (Open Graphics Library) is a standard specification defining a cross language, cross-platform API for writing applications that produce 2D and 3Dcomputer graphics, describing a set of functions and the precise behavior that they must perform. From this specification, hardware vendors create implementations-libraries of functions created to match the functions stated in the OpenGL specification, making use of hardware acceleration where possible. Hardware vendors have meet specific tests to be able to qualify their implementation as an OpenGL implementation.

GLUT is the OpenGL Utility Toolkit, a window system independent toolkit for writing OpenGL programs. It implements a simple windowing application programming interface (API) for OpenGL. GLUT makes it considerably easier to learn about and explore OpenGL programming. GLUT provides a portable API so you can write a single OpenGL program that works across all PC and workstation OS platforms.

**Chapter 3**

**DESIGN**

The design of any software depends on the architecture of the machine on which that software runs, for which the designer needs to know the system architecture. Design process involves design of suitable algorithms, modules, sub systems, interfaces etc.

**3.1 Algorithm**

An Algorithm is a finite sequence of instructions, an explicit step by step procedure for solving a problem, often used for calculation and data processing.

Step1: Begin

Step2: Display the first page

Step3: Accept the key from user

Step4: Repeat the following steps until the last page

Step5: If the key is n

Goto next page

Step6: Accept key from user

Step7: If key is r

Goto step4

Else if key is q

Exit

**3.2 Flow Chart**

Yes

Yes

A

Developed embryopage

If key pressed is n

Frog span page

If key pressed is n

First page

If key pressed is n

Yes

Tadpole page

Froglet page

Yes

If key pressed is n

Tadpoles with legs page

Developed tadpole page

If key pressed is n

If key pressed is n

**A**

**Figure 3.2** Flowchart

If key pressed is r

No

Yes

Yes

Yes

If key pressed is q

Life Cycle Page

If key pressed is n

Developed Frog page

Yes

If key pressed is n

**Chapter 4**

**IMPLEMENTATION**

**4.1 Functions**

The functions that are used in the program are discussed below. This section contains brief description of all the headers and functions. These functions are as follows:

**4.1.1 Headers Defined**

The in-built are defined in the OpenGL library. Some of the headers that are used

as follows

* **#include<stdio.h>** : to take input from standard input and write to standard output
* **#include<stdlib.h>** : to include standard library functions
* **#include<GL/glut.h>** :to include glut library files

**4.1.2 Inbuilt Functions**

OpenGL functions used in the code are as follows:

* **glClear Color()**

Specifies clear values to the colour buffers and clears the display before redrawing it. It specifies the red, green, blue and alpha values used by glClear to clear the buffers. Values specified is in the range of [0, 1].

* **glClear()**

Clears buffer to preset values .Specifies BITWISE OR of masks that indicate the buffers to be cleared.

* **glBegin()**

Specifies the primitive or primitives that will be created from vertices presented between glBegin() and glEnd().

* **glTranslate()**

It produces a translation by (x,y,z). The current matrix is multiplied by this translation matrix, with the product replacing the current matrix.

* **glutKeyboardFunc()**

It is a user interactive function which displays snapshots on hitting the

appropriate keyboard keys onto the display.

* **glutDisplayFunc()**

It is the function that displays the primitives onto the screen by calling the user defined functions.

* **glVertex()**

Function commands are used within the glBegin/glEnd to specify point, line and polygon vertices.

* **glColor3f()**

Set the color

* **glutInitWindowSize()**

Specifies the Initial height and width of the window pixel

* **glutInitWindowPosition()**

Specifies the initial position of the top-left corner of the window pixels

* **glutCreateWindow()**

Creates a window on the display

* **gluOrtho2D()**

Multiply the current matrix with an orthographic matrix

* **glutMainLoop()**

It enters GLUT event proceesing Loop.This routine shoulb be called at least once in a GLUT program

**4.1.3 User Defined Functions**

* **void display()**

This is the display function used to display the objects on the screen.

* **void hindlegbigtad()**

This function is used to draw the tadpoles which have hind legs.

* **void lastpage()**

This function is used to display the page which displays the entire life cycle

* **void sectext()**

This function displays the text on the second page.

* **void thirdtext()**

This displays the text in the third page.

* **void fourthtext()**

This displays the text in the fourth page.

* **void fifthtext()**

This displays the text in the fifth page.

* **void sixthtext()**

This displays the text in the sixth page.

* **void seventhtext()**

This displays the text in the seventh page.

* **void firstpagetext()**

This displays the details of the student and college.

* **void key(unsigned char k,int x,int y)**

This is the keyboard function which accepts a character from the keyboard.

* **void embcluster()**

This function draws a cluster of embryos.

* **void bigtad()**

Draws elder tadpoles.

* **void frog(int n)**

This function draws an elder frog.

* **void justeggs()**

This function draws eggs without embryo.

* **void swamp()**

This function draws a background scene of a swamp.

* **void embtail()**

This function draws an embryo with tail.

* **void firstpage()**

This function is used to draw the background of the first page.

* **void egg()**

This function is used to draw the eggs.

* **void arrow()**

This function is used to draw the arrow marks in the last page.

* **void throat()**

This function is used to draw the air filled throat of the frog.

* **void embryo(float n)**

This function is used to draw the rotating embryos.

* **void plant()**

This function is used to draw the plants.

* **void ground()**

This function is used to draw theground.

* **void stones()**

This function is used to draw the stones on the river bed.

* **void bubbles()**

This function is used to draw the bubbles.

* **void tadlegbig()**

This function is used to draw the elder tadpoles with legs.

* **void bigtadclust()**

Draws a cluster of big tadpoles.

* **void tadlegs(float legspd)**

Draws the moving legs of the tadpoles.

* **voidtadlegclust()**

Draws the cluster of tadpoles with legs.

**Chapter 5**

**TESTING**

**5.1 Introduction to Testing**

Verification and validation is a generic name given to checking processors, which ensures that the software confirms to it specifications and meets the demand of users.

**Validation**

Are we building the right product?

Validation involves checking at that the program has implanted meets the requirement of the users.

**Verification**

Are we building the product right?

Verification involves checking that the program confirms to its specification.

**5.2 Stages in the Implementation of Testing**

**Unit Testing**

Each individual unit is tested for correctness. These individual components will be tested to ensure that they operate correctly.

**Integration Testing**

A module is a collection of dependent components such as a function. A module encapsulates related components so can test without other system modules.

**System Testing**

The sub-system are integrated to make up the entire system. The errors that result from an anticipated interaction between sub-systems and system components are removed.

**User Acceptance Testing**

This is the final stage in the testing process before the system is tested for operational use. Any requirement problem or requirement definition problem revealed from acceptance testing are considered and made error free.

**Test Plan**

Careful planning is needed to the most of testing and controlled testing cost.

**5.3 Test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TC ID** | **Testcase Description** | **Input** | **Actual Output** | **Expected Output** | **Remark** |
| 1 | About Frog Spawn | Pressing key ‘ n’ in the keyboard | It is displaying the frog spawn page, Refer Figure 6.1 | Frog spawn page should appear | Pass |
| 2 | Development of embryo | Pressing key ‘ n’ in the keyboard | It is displaying the developed embryo page, Refer Figure 6.2 | Developed embryo page should appear | Pass |
| 3 | About the tadpole | Pressing key ‘ n’ in the keyboard | It is displaying the tadpole page. Refer Figure 6.3 | Tadpole page should appear | Pass |
| 4 | Development of tadpole | Pressing key ‘ n’ in the keyboard | It is displaying the developed tadpole page, Refer Figure 6.4 | Developed tadpole page should appear | Pass |
| 5 | Development of legs in tadpole | Pressing key ‘ n’ in the keyboard | It is displaying the tadpoles with legs page, Refer Figure 6.5 | Tadpoles with legs page should appear | Pass |
| 6 | About the froglet | Pressing key ‘ n’ in the keyboard | It is displaying the froglet page, Refer Figure 6.6 | Froglet page should appear | Pass |
| 7 | Complete development of frog | Pressing key ‘ n’ in the keyboard | It is displaying the developed frog page, Refer Figure 6.7 | Developed frog page should appear | Pass |
| 8 | About the life cycle and to view again all the phases or to exit | Pressing key ‘ n’ in the keyboard | It is displaying the life cycle page, Refer Figure 6.8 | Life cycle page should appear | Pass |

**Figure 5.1** Test case table

**Chapter 6**

**SNAPSHOTS**

The chapter involves snapshots of the project, which provides an idea of the project.

These snapshots depict different situations of the output.



**Figure 6.1** Frog spawn page

The above Figure 6.1 illustrates the frog spawn page.



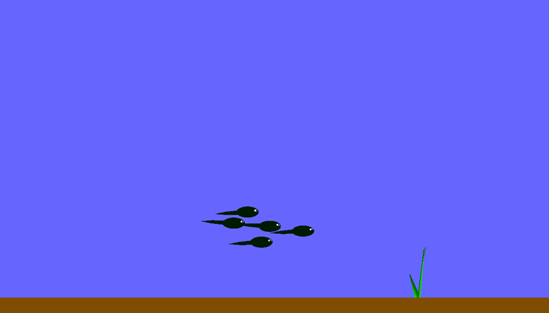
**Figure 6.2** Developed embryo page

The above Figure 6.2 illustrates the developed embryo page.



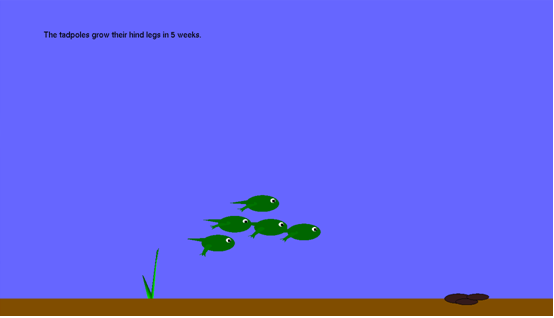
**Figure 6.3** Tadpole page

The above Figure 6.3 illustrates the tadpole page.



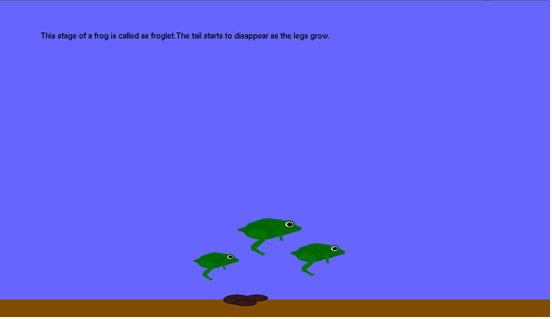
**Figure 6.4** Developed tadpole page

The above Figure 6.4 illustrates the developed tadpole page.



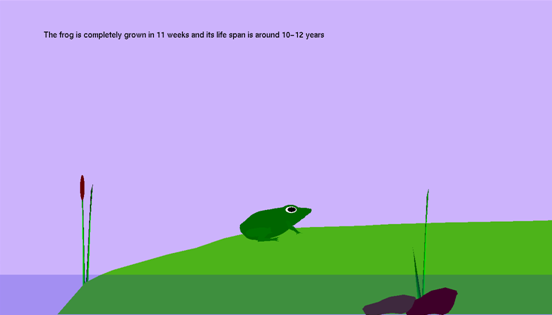
**Figure 6.5** Tadpoles with legs page

The above Figure 6.5 illustrates the tadpole with legs page.



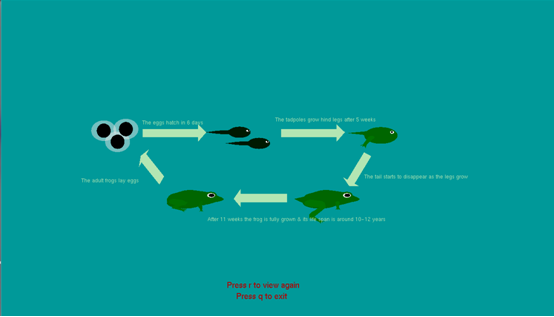
**Figure 6.6** Froglet page

The above Figure 6.6 illustrates the froglet page.



**Figure 6.7** Developed Frog page

The above Figure 6.7 illustrates the developed frog page.



**Figure 6.8** Life Cycle page

The above Figure 6.8 illustrates the life cycle page.

**CONCLUSION**

Computer graphics plays a major role in today’s world where visualization takes the upper hand as compared to textual interaction. This is largely true as we can see user interfaces becoming more and more attractive all thanks to the major leaps in the field of computer graphics. The project which we have designed is with an aim to implement an application in computer graphics in OpenGL. It was a great experience developing this project. We learnt many concepts of computer graphics while developing this project.

**FUTURE ENHANCEMENT**

We have implemented graphical functions such as, translation, rotation, push and pop for the development of frog in different stages. We can further add audio to the current project to make it more impactful. We look forward to implement this in future along with some additional functions and algorithms.

**BIBLIOGRAPHY**

**Books**

[1]. Edward Angel, 2009, Interactive Computer Graphics 5th edition, Pearson

Educational ltd

[2]. Donald Hearn and Pauline Baker: Computer Graphics-OpenGL version, 3rdEdition,

Pearson Education, 2004

**Websites**

[1]. http://www.allgraphicsdesign.com

[2]. http://redbook.com