CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO COMPUTER GRAPHICS

Computer Graphics is concerned with all aspects of producing pictures or images using a computer. Graphics provides one of the most natural means of communicating within a computer, since our highly developed 2D and 3D pattern-recognition abilities allow us to perceive and process pictorial data rapidly and effectively. Interactive computer graphics is the most important means of producing pictures since the invention
of photography and television.

Applications of Computer Graphics

1. Display of information
2. Design
3. Simulation and animation
4. User interfaces

The Graphics Architecture

Graphics Architecture can be made up of seven components:

1. Display processors
2. Pipeline architectures
3. The graphics pipeline
4. Vertex processing
5. Clipping and primitive assembly
6. Rasterization
7. Fragment processing

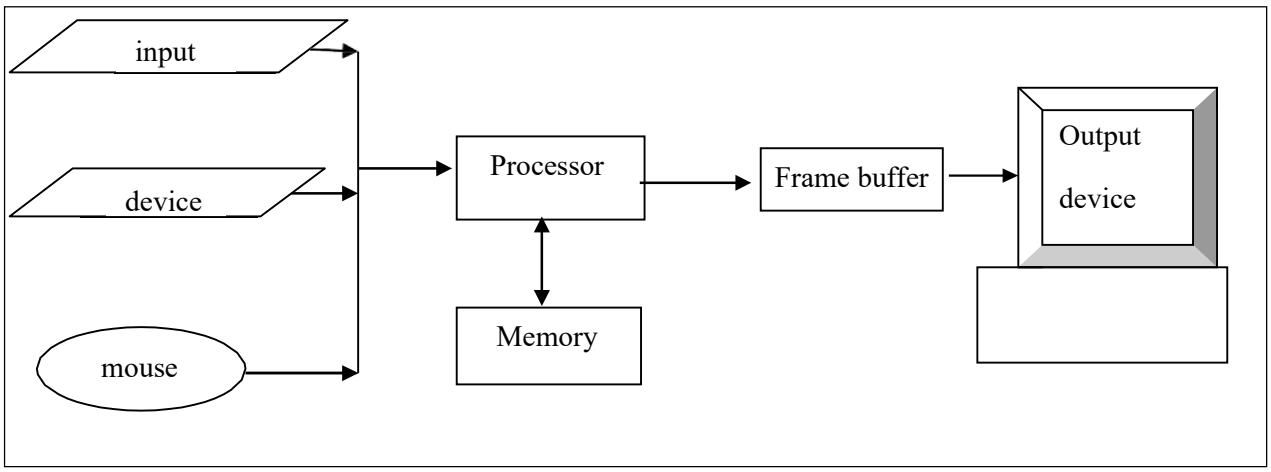


Figure 1.1: Components of Graphics Architecture and their working

1.2 INTRODUCTION TO OPENGL

OpenGL is software used to implement computer graphics. The structure of OpenGL is similar to that of most modern APIs including Java 3D and DirectX. OpenGL is easy to learn, compared with other.

APIs are nevertheless powerful. It supports the simple 2D and 3D programs. It also supports the advanced rendering techniques. OpenGL API explains following 3 components

1. Graphics functions
2. Graphics pipeline and state machines
3. The OpenGL interfaces

There are so many polygon types in OpenGL like triangles, quadrilaterals, strips and fans. There are 2 control functions, which will explain OpenGL through,

1. Interaction with window system
2. Aspect ratio and view ports

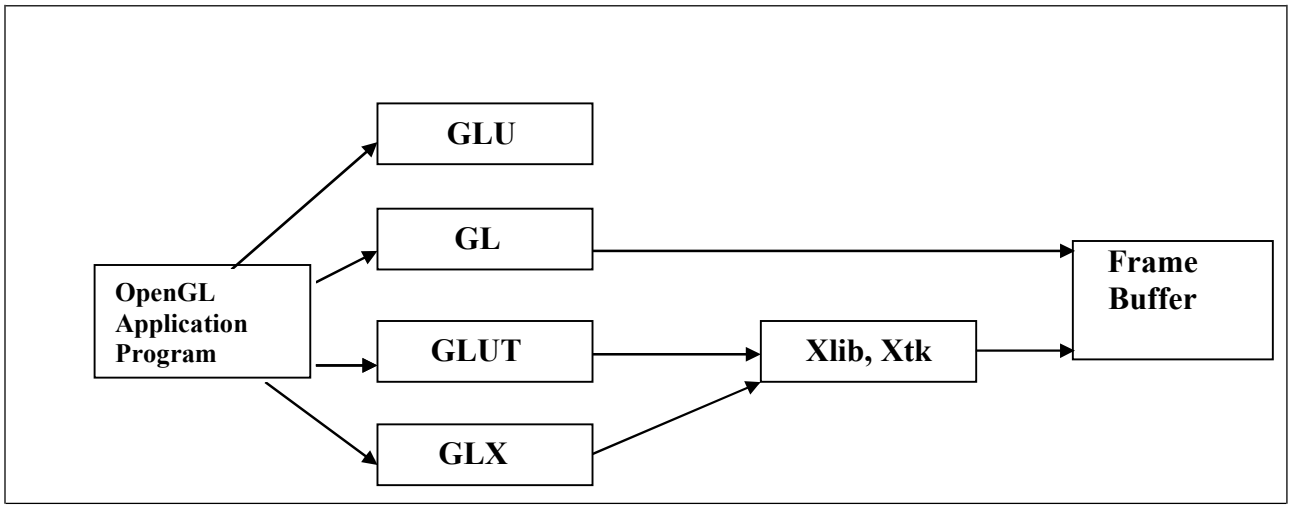


Figure 1.2: OpenGL Library organization

Most implementations of OpenGL have a similar order of operations, a series of processing stages called the OpenGL rendering pipeline. This ordering, as shown in Figure 1.2, is not a strict rule of how OpenGL is implemented but provides a reliable guide for predicting what OpenGL will do. The following diagram shows the assembly line approach, which OpenGL takes to process data. Geometric data (vertices, lines, and polygons) follow the path through the row of boxes that includes evaluators and per-vertex operations, while pixel data (pixels, images, and bitmaps) are treated differently for part of the process. Both types of data undergo the same final steps before the final pixel data is written into the frame buffer.

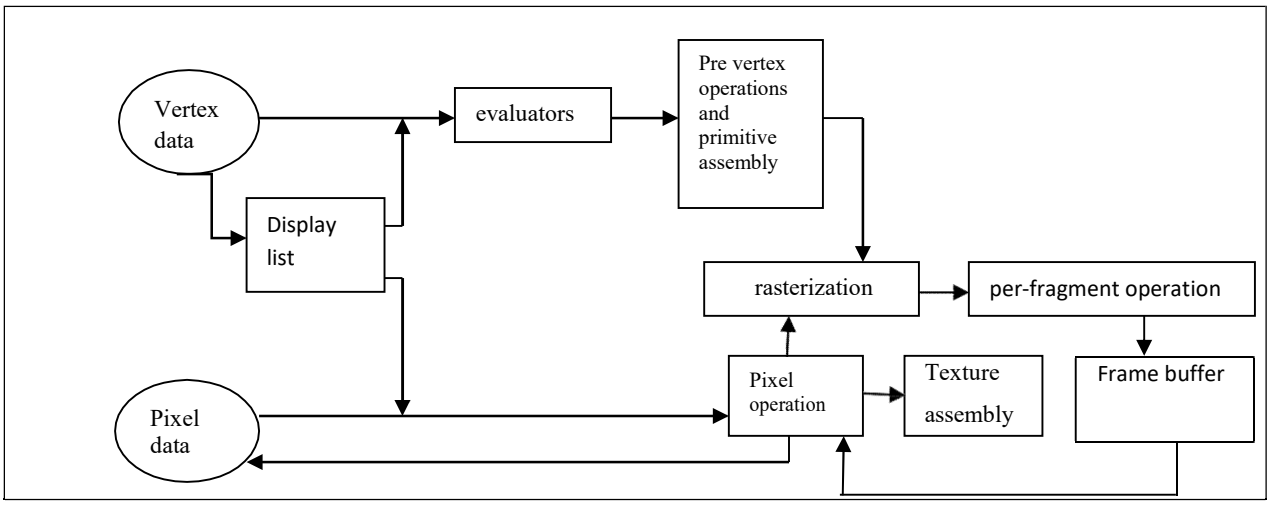


Figure 1.3: OpenGL Order of Operations

CHAPTER 2

REQUIREMENTS SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

- Operating system – Windows 10/linux
- Code::Blocks 17.12
- OPENGL library files – GL, GLU, GLUT
- Language used is C/C++

2.2 HARDWARE REQUIREMENTS

- Processor – Intel i5 7^{th} Gen
- Memory – 8GB RAM
- 1TB Hard Disk Drive
- Mouse or other pointing device
- Keyboard
- Display device

CHAPTER 3

SYSTEM DEFINITION

3.1 PROJECT DESCRIPTION

This project aims to demonstrate human anatomy such as digestive system, respiratory system and excretory system using openGL. Digestive system shows the movement of food in the body. Respiratory system shows the effect of inhalation and exhalation on lungs (i.e. breathing). Excretory system shows the main organs involved in excretion.
These different anatomies are put in a menu and one can select the menu option for look into a particular anatomy.

The main aim is to understand the key ideas and Implementation of Computer Graphics using OpenGL for representing "HUMAN ANATOMY". It is necessary to display the object in the user's point of view.

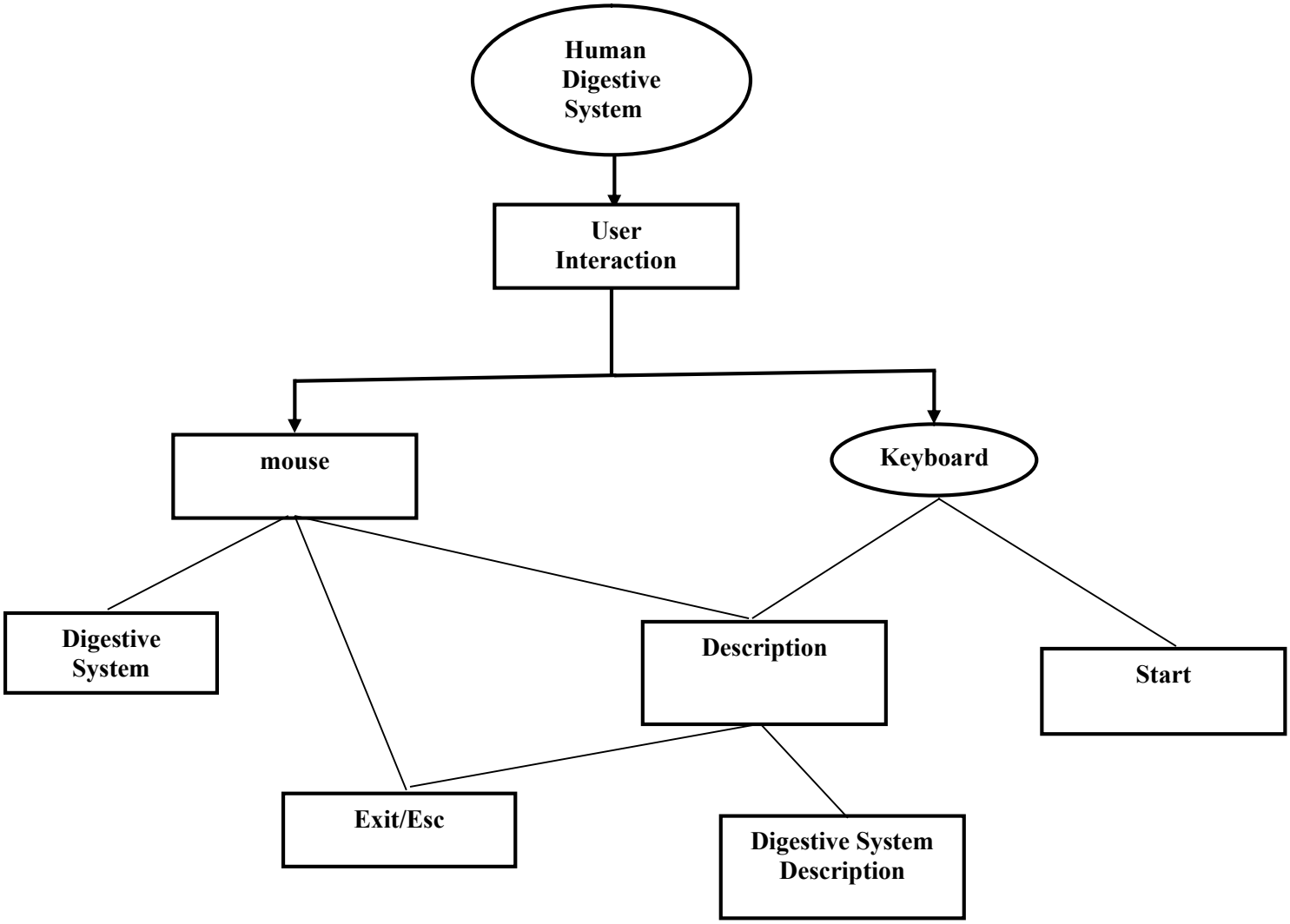
- The aim is to draw attention of users toward computer graphics.
- The aim is to create dominant project which is simple in use.

3.2 USER DEFINED FUNCTIONS

- **frontscreen():** used to display the strings of the project title and our names in the flickers screen as well as the other strings.
- **Init():** used for shading purposes.
- **drawstring():** used to print text strings.
- **drawstring1():** used to print text strings.
- **body():** used to draw human body.
- **digesdraw():** used to draw various organs in digestive system with labeling.
- **liver():** used to draw liver in digestive system.
- **largeintes():** used to draw large intestine of digestive system.
- **food() :**used to draw food particle for digestive system.
- **foodp():** used to draw small food particle for digestive system.
- **digest():** used to draw digestive system.
- **animate():** used to movement to food particle.

- **descDigest():** used to display description about digestive system.
- **mydisplay():** used for displaying the entire scene.
- **myKeyboardFunc():** used for giving the keyboard inputs.
- **desc():** instruction window is displayed for necessary key events.
- **mymenu():** used for setting up the menu to be displayed.
- **Reshape():** used for setting up the viewport.

3.3 DATA FLOW DIAGRAM



CHAPTER 4

IMPLEMENTATION

4.1 SOURCE CODE

```
#include<GL/glut.h>
#include<string.h>
#include<math.h>
#include<stdlib.h>
#define fi 4
void desc();
int flag=0;
GLfloat v[][3]={{-35,30,-2.5},{-20,30,-2.5},{-20,40,-2.5},{-35,40,-2.5},{-35,30,2.5},{-
20,30,2.5},{-20,40,2.5},{-35,40,2.5}};
GLfloat dx=0,dy=0,tx=0,ty=0,mx=0,my=0,rx=0,ry=0;
GLfloat a,b;
unsigned char keyf=0;
int strflag=0;

void drawstring(float x,float y,float z,const char *str)
{
unsigned int c;
glRasterPos3f(x,y,z);
for(c=0;c<strlen(str);c++)
glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,str[c]);
}

void drawstring1(float x,float y,float z,const char *str)
{
unsigned int c;
glRasterPos3f(x,y,z);
if(strflag==0)
for(c=0;c<strlen(str);c++)
glutBitmapCharacter(GLUT_BITMAP_HELVETICA_12,str[c]);
if(strflag==1)
for(c=0;c<strlen(str);c++)
glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24,str[c]);
if(strflag==2)
for(c=0;c<strlen(str);c++)
glutBitmapCharacter(GLUT_BITMAP_9_BY_15,str[c]);
}


void drawline(GLfloat x1,GLfloat y1,GLfloat x2,GLfloat y2)
{
```

```
glBegin(GL_LINES);
glVertex3f(x1,y1,0);
glVertex3f(x2,y2,0);
glEnd();
}

void drawcircle(GLfloat m, GLfloat n)
{
GLfloat i;
glBegin(GL_LINE_LOOP);
for(i=0;i<=2*3.14;i+=0.0005)
glVertex3f(m+cos(i)*3,n+sin(i)*3,0);
glEnd();
}

void frontscreen(void)
{
 glClear(GL_COLOR_BUFFER_BIT);
 glColor3f(0,0,1);
 drawstring(-15,40,0.0,"GLOBAL ACADEMEY OF TECHNOLOGY");
 glColor3f(0.7,0,1);
 drawstring(-30,35,0.0,"DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING");
 glColor3f(1,0.5,0);
 drawstring(-15,30,0.0,"A MINI PROJECT ON ");
 glColor3f(1,0,0);
 drawstring(-25,20,0.0,"HUMAN DIGESTIVE SYSTEM");
 glColor3f(1,0.5,0);
 drawstring(-35,0,0.0,"BY:");
 glColor3f(0.5,0,0.5);
 drawstring(-35,-10,0.0,"SAURAV (1GA17CS139)");
 strflag=0;
 drawstring(-35,-14,0.0,"6th Semester,CSE");
 glColor3f(1,0.1,1);
 drawstring(-13,-35,0.0,"PRESS ENTER TO START");
 glFlush();
}

void body()
{
GLfloat i,x,y;
glColor3f(0,0,0);
glBegin(GL_POLYGON);
glVertex3f(x=0+cos(1)*7,y=35+sin(1)*7,0);
for(i=1;i<=3.81;i+=0.05)
glVertex3f(0+cos(i)*7,36.3+sin(i)*7,0);
glVertex3f(-7,35,0);
glVertex3f(-6,27,0);
```

```
glEnd();
//face
glColor3ub(255,191,128);
glBegin(GL_POLYGON);
glVertex3f(x,y,0);
glVertex3f(x,y-5,0);
glVertex3f(x,y-5.5,0);
glVertex3f(x,y-6,0);
glVertex3f(x+1.5,y-9,0);
glVertex3f(x,y-9,0);
glVertex3f(x,y-11,0);
glVertex3f(x-0.5,y-12,0);
glVertex3f(x-0.5,y-13,0);
glVertex3f(x-1,y-14,0);
glVertex3f(x-3,y-16,0);
glVertex3f(x-6,y-16,0);
glVertex3f(-6,27,0);
glVertex3f(-5,30,0);
glVertex3f(-4,32.5,0);
glVertex3f(-3.7,35,0);
//glVertex3f(-3.5,37,0);
//glVertex3f(-3,36,0);
glVertex3f(-2,35,0);
glEnd();
glBegin(GL_POLYGON);//neck
glVertex3f(1,27,0);
glVertex3f(-4,27,0);
glVertex3f(-4,22,0);
glVertex3f(1,22,0);
glEnd();
glBegin(GL_POLYGON);
for(i=3.14;i>(3.14/2.0);i-=0.05)
glVertex3f(-9+cos(i)*5,15+sin(i)*7,0);
glVertex3f(-9,22,0);
glVertex3f(5,22,0);
for(i=(3.14/2.0);i>0;i-=0.05)
glVertex3f(5+cos(i)*7,15+sin(i)*7,0);
glEnd();
glBegin(GL_POLYGON);
glVertex3f(-9,15.1,0);
glVertex3f(7,15.1,0);
glVertex3f(6,0,0);
glVertex3f(-8,0,0);
glEnd();
glBegin(GL_POLYGON);//left hand
glVertex3f(-13.99,15.011,0);
glVertex3f(-9,15.011,0);
```

```
glVertex3f(-10,2.011,0);
glVertex3f(-13.59,2.011,0);
glEnd();
glBegin(GL_POLYGON);//Right hand
glVertex3f(12,15.011,0);
glVertex3f(6.7,15.011,0);
glVertex3f(8,1.011,0);
glVertex3f(11.5,1.011,0);
glEnd();
glBegin(GL_POLYGON);
glVertex3f(6,0,0);
glVertex3f(-8,0,0);
glVertex3f(-9,-18,0);
glVertex3f(7,-18,0);
glEnd();
dx=x-1;dy=y-11;
a=x;b=y;
}


GLfloat aa,bb;
void digesdraw()
{
GLfloat i;
body();
glLineWidth(15);
glColor3ub(255,100,50);
glBegin(GL_LINE_STRIP);
glVertex3f(a,b-11,0);
for(i=(3.14/2.0);i<=3.14;i+=0.0005)
glVertex3f(2+cos(i)*3.5,26+sin(i)*3.5,0);//oesophagus1
glVertex3f(-1.5,27,0);
glVertex3f(-1.5,4.5,0);
glEnd();
glBegin(GL_LINE_STRIP);
glVertex3f(a,b-11,0);
for(i=(3.14/2.0);i<=3.14;i+=0.0005)
glVertex3f(2.5+cos(i)*3.5,26.2+sin(i)*3.5,0);//oesophagus2
glVertex3f(-1.2,25,0);
glVertex3f(-1.2,4,0);
glEnd();
glBegin(GL_POLYGON);//stomach upper
for(i=(5*3.14/2.0);i>=(3*3.14/2);i-=0.005)
glVertex3f(aa=(1+cos(i)*4),bb=(0+sin(i)*4),0);
glVertex3f(-1.5,5,0);
glEnd();
glBegin(GL_POLYGON);//stomach lower
```

```
glVertex3f(aa,bb,0);
for(i=(3.14/2);i<(3*3.14/2);i+=0.005)
glVertex3f(0+cos(i)*2.5,-2+sin(i)*2.5,0);
glEnd();
glBegin(GL_LINE_STRIP);
glVertex3f(-2.5,-2,0);
for(i=(3.14/2.0);i<=3.14;i+=0.0005)
glVertex3f(aa=-1.4+cos(i)*3,bb=-4.5+sin(i)*3,0);
for(i=(3.14/2.0);i<=3.0;i+=0.0005)
glVertex3f(-1.3+cos(i)*3,-5+sin(i)*3,0);
//glVertex3f(-3,-3.2,0);
//glVertex3f(-3,-3.5,0);
glEnd();
double freq=0.3,amp=4.5;
glBegin(GL_LINE_STRIP);
glVertex3f(aa,bb,0);
glVertex3f(aa,bb-3,0);
glEnd();
glBegin(GL_LINE_STRIP);//small interestine
for(i=-5;i>=-14;i-=0.005)
glVertex3f((amp*sin(2*3.14*freq*i)),i,0);
glEnd();

glColor3f(0,0,0);//border
glLineWidth(2.5);
glBegin(GL_LINE_LOOP);
glVertex3f(41,51,0);
glVertex3f(-41,51,0);
glVertex3f(-41,-21,0);
glVertex3f(41,-21,0);
glEnd();
glLineWidth(1);
glBegin(GL_LINE_LOOP);
glVertex3f(40,50,0);
glVertex3f(-40,50,0);
glVertex3f(-40,-20,0);
glVertex3f(40,-20,0);
glEnd();
strflag=1;
glColor3ub(0, 207, 119);
drawstring1(-20,-30,0,"HUMAN DIGESTIVE SYSTEM");
strflag=0;
glColor3f(0,0,0);
//drawline(a+1,b-9,a+5.6,b-9);
//drawstring1(a+6,b-9,0,"Nose");
drawline(a,b-12,a+6.5,b-12);
drawstring1(a+7,b-12,0,"Mouth");
```

```
drawline(-10,28,-1,28);
drawstring1(-16,28,0,"Pharynx");
drawline(-12,22,-1.5,22);
drawstring1(-20,22,0,"Oesophagus");
drawline(10,0,3,0);
drawstring1(10.5,0,0,"Stomach");
drawline(-14,-2,-2,-2);
drawstring1(-22,-2,0,"Duodenum");

//glLoadIdentity();
}


void liver()
{
GLfloat i;
glColor3ub(200,100,50);//liver
glBegin(GL_POLYGON);
glVertex3f(-1.2,4,0);
glVertex3f(-1.5,5,0);
glVertex3f(-1,6,0);
for(i=(3.14/2.0);i<=3.5;i+=0.005)
glVertex3f(0+cos(i)*6,0+sin(i)*7,0);
glEnd();
glLineWidth(1);
strflag=0;
glColor3f(0,0,0);
drawline(-15,4,-4,4);
drawstring1(-20,4,0,"Liver");
}
void largeinstes()
{
glLineWidth(15);
GLfloat i;
glColor3ub(240,150,50);//large interstine
glBegin(GL_LINE_LOOP);
glVertex3f(-1,-17.5,0);
glVertex3f(-1,-15,0);
glVertex3f(4,-15,0);
glVertex3f(4,-6,0);
glVertex3f(-5.3,-6,0);
glVertex3f(-5.3,-14,0);
glVertex3f(-2,-13,0);
glVertex3f(-2,-14,0);
glVertex3f(-6,-14,0);
glVertex3f(-6,-5,0);
glVertex3f(4.7,-5,0);
```

```
glVertex3f(4.7,-16,0);
glVertex3f(3,-16,0);
glVertex3f(1,-16,0);
glVertex3f(-1,-16,0);
glEnd();
glLineWidth(1);
strflag=0;
glColor3f(0,0,0);
drawline(-14,-7,-5,-7);
drawstring1(-24,-7,0,"Large intestine");
drawline(9.5,-9,1,-9);
drawstring1(10,-9,0,"Small intestine");
drawline(9.5,-15,-1,-15);
drawstring1(10,-15,0,"Rectum");
drawline(9.5,-17.5,-1,-17.5);
drawstring1(10,-17.5,0,"Anus");

}
int countd=0;

void food()
{
glPointSize(5);//food
glColor3ub(20,190,50);
//glColor3f(0,0,0);
glBegin(GL_POINTS);
glVertex3f(0.9,27,0);
glEnd();
glBegin(GL_POINTS);
glVertex3f(0.65,27.2,0);
glEnd();
}

void foodp()
{
glPointSize(3);//food
glColor3ub(20,190,50);
//glColor3f(0,0,0);
glBegin(GL_POINTS);
glVertex3f(-4.4,6.5,0);
glVertex3f(-3.9,6.5,0);
glVertex3f(-4.1,7.2,0);
glEnd();
}


void digest()
```

```
{
GLfloat i,dy=-1;
digesdraw();
glPushMatrix();
//glColor3f(1.0,1.0,0.0);
glTranslatef(dx+tx,dy+ty,0.0);
food();
glPopMatrix();
glPushMatrix();
glTranslatef(dx+mx,dy+my,0.0);
foodp();
glPopMatrix();
liver();
largeinstes();
if(countd==1)
{
glColor3ub(115, 183, 0);
glLineWidth(3);
drawline(a+9,b-9,a+9.5,b-10);
drawline(a+9,b-9,a+9.5,b-8);
drawline(a+9,b-9,a+13,b-9);
drawstring(a+14,b-9,0,"Food");
}
glutSwapBuffers();
}

//for(i=(3.14/2.0);i<=3.0;i+=0.0005)
//glVertex3f(-1.3+cos(i)*3,-5+sin(i)*3
GLfloat theta1=5*3.14/2.0,theta=3.14/2.0,theta2=3.14/2.0;
void animate()
{
if(flag==1)
{
if(tx<20 && ty<20 && theta<=3.14)
{
tx=-1.2+cos(theta)*3.5;
ty=sin(theta)*3.5;
countd=1;
//food
theta+=0.005;
}
if(theta>=3.14 && ty>=-20.5)
{
ty=ty-0.01;
countd=0;
}
```

```
if(theta>=3.14 && theta1>= 3*3.14/2.0 && ty<=-20.5 && ty>=-30)
{
mx=-1.5+cos(theta1)*4;
my=-4.8+sin(theta1)*4;
theta1-=0.005;

}

else if(theta1<= 3*3.14/2.0 && my<=-5.79 && my>=-11 && theta2<=3.0)
{
mx=mx-0.01;
my=my-0.01;

//theta2+=0.005;
}
glutPostRedisplay();
}
}

void descDigest()
{
//glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
glColor3ub(253, 123, 119);//border
glBegin(GL_LINE_LOOP);
glVertex3f(47,50,0);
glVertex3f(-45,50,0);
glVertex3f(-45,-40,0);
glVertex3f(47,-40,0);
glEnd();
glLineWidth(2.5);
glBegin(GL_LINE_LOOP);
glVertex3f(48,51,0);
glVertex3f(-46,51,0);
glVertex3f(-46,-41,0);
glVertex3f(48,-41,0);
glEnd();

glColor3f(0.7,0,1);
strflag=1;
drawstring1(-20,45,0.0,"Human Digestive System");
glColor3ub(255, 159, 0);
drawstring(-35,30,0.0,"Here are the major stages of the Digestive system:");
strflag=2;
glColor3ub(0, 207, 119);
drawstring1(-35,25,0.0,"1. Chewing");
glColor3f(0,0,0);
strflag=0;
```

```
drawstring1(-24,25,0.0,"is the first stage of the digestive system. When you chew your food it
breaks up big pieces ");
drawstring1(-26,22.5,0.0,"into little pieces that are easier to digest and swallow.Also, the saliva is
more than just water.");
drawstring1(-26,20,0.0," It has special enzymes in it that start to breakdown starchy food while
chewing.");
strflag=2;
glColor3ub(0, 207, 119);
drawstring1(-35,17,0.0,"2. Swallowing");
strflag=0;
glColor3f(0,0,0);
drawstring1(-17,17,0.0,"may seem like a simple process to us. But food doesn't just fall down our
throats ");
drawstring1(-26,14.5,0.0,"into the stomach.");
strflag=2;
glColor3ub(0, 207, 119);
drawstring1(-35,9,0.0,"3. Stomach");
strflag=0;
glColor3f(0,0,0);
drawstring1(-21,9,0.0,"The next stage is the stomach. Food hangs out in the stomach for around
four hours. ");
drawstring1(-26,6.5,0.0,"While the food sits there, more enzymes go to work on it, breaking down
things like proteins ");
drawstring1(-26,4,0.0,"that our bodies can use. The stomach kills a lot of bad bacteria as well, so we
don't get sick.");
strflag=2;
glColor3ub(0, 207, 119);
drawstring1(-35,-2,0.0,"4. Small Intestine");
strflag=0;
glColor3f(0,0,0);
drawstring1(-15,-2,0.0,"The first part of the small intestine works with juices from the liver and
pancreas");
drawstring1(-26,-5,0.0," to continue to break down our food. The second part is where the food gets
absorbed from the");
drawstring1(-26,-7.5,0.0,"intestine and into our body through the blood.");
strflag=2;
glColor3ub(0, 207, 119);
drawstring1(-35,-14,0.0,"5. Large Intestine");
strflag=0;
glColor3f(0,0,0);
drawstring1(-15,-14,0.0,"The last stage is the large intestine. Any food that the body doesn't need or
can't");
drawstring1(-26,-17,0.0," use is sent to the large intestine and later leaves the body as waste. ");
strflag=2;
glColor3ub(0, 207, 119);
drawstring1(-35,-24,0.0,"6. The Liver and Pancreas");
strflag=0;
```

```
glColor3f(0,0,0);
drawstring1(-6,-24,0.0,"The liver and pancreas do a lot to help the digestive system along.");
drawstring1(-26,-26.5,0.0,"Both work with the small intestine. The liver provides bile that helps
break up fat into");
drawstring1(-26,-29,0.0," smaller bits. The pancreas provides additional enzymes to help digest all
sorts of food. ");

void passive(int x,int y)
{

if(x>=450 && x>=490 && y>=150 && y<=170)
{
glutSetCursor(GLUT_CURSOR_INFO);
drawline(a+1,b-9,a+5.6,b-9);
drawstring1(a+6,b-9,0,"nose");
}
glutPostRedisplay();
}

void mydisplay(void)
{
glViewport(10,-50,980,700);
 glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
 if(flag==0)
  frontscreen ();
 if(flag==1)
 {
  digest();
  glutIdleFunc(animate);
 }

  desc();
glutSwapBuffers();
}

void myKeyboardFunc( unsigned char key, int x, int y )
{

 if(flag==0 && key==13) //Ascii of 'enter' key is 13
    flag=1;
 if(key=='d')
  keyf=key;
 if(key=='D')
  keyf=key;
 if(key=='r')
  keyf=key;
 if(key=='R')
```

```
  keyf=key;
 if(key=='e')
  keyf=key;
 if(key=='E')
  keyf=key;
 if(key=='Q' || key=='q' || key==27)
  exit(0);

mydisplay();
}
 void desc()
{
glClear(GL_COLOR_BUFFER_BIT);
glColor3ub(17, 40, 60);//border
glBegin(GL_LINE_LOOP);
glVertex3f(47,50,0);
glVertex3f(-45,50,0);
glVertex3f(-45,-40,0);
glVertex3f(47,-40,0);
glEnd();
glLineWidth(2.5);
glBegin(GL_LINE_LOOP);
glVertex3f(48,51,0);
glVertex3f(-46,51,0);
glVertex3f(-46,-41,0);
glVertex3f(48,-41,0);
glEnd();
glColor3ub(58, 62, 73);
strflag=1;

glColor3ub(159, 34, 70);
drawstring(-37,30,0.0," The human body is the entire structure of a human being. It is composed");
drawstring(-37,26,0.0,"of many different types of cells that together create tissues and subse-");
drawstring(-37,22,0.0,"quently organ systems. They ensure homeostasis and the viability of the ");
drawstring(-37,18,0.0,"human body.");
drawstring(-37,13,0.0," It comprises a head, neck, trunk (which includes the thorax and abdomen)");
drawstring(-37,9,0.0,"arms and hands, legs and feet.");
drawstring(-37,4,0.0," The study of the human body involves anatomy, physiology, histology and");
drawstring(-37,0,0.0,"embryology. The body varies anatomically in known ways.Physiology focus-
");
drawstring(-37,-4,0.0,"-es on the systems and organs of the human body and their functions.
Many");
drawstring(-37,-8,0.0,"systems and mechanisms interact in order to maintain homeostasis, with
safe");
drawstring(-37,-12,0.0,"levels of substances such as sugar and oxygen in the blood.");
strflag=2;
glColor3ub(159, 176, 167);
```

```
drawstring1(-35,-20,0.0,"Enter key 'd' or 'D' to see brief description on Digestive system");
drawstring1(-24,-35,0.0,"Enter 'ESC' or 'q' or 'Q' to exit");
if(keyf=='d' || keyf=='D')
{
glClear(GL_COLOR_BUFFER_BIT);
descDigest();
glutIdleFunc(descDigest);
}

//glutPostRedisplay();
}


void mymenu(int id)
{
switch(id)
{
case 1: frontscreen();
break;
case 2:flag=1;
glutIdleFunc(animate);
//glutPassiveMotionFunc(passive);
tx=0;ty=0;theta=3.14/2.0;theta1=5*3.14/2.0;theta2=3.14/2.0;mx=0;my=0;
break;
case 3: flag=2;
  desc();
  glutIdleFunc(desc);
  glutKeyboardFunc(myKeyboardFunc);
  keyf=0;
break;
case 6:exit(0);
}
}




void init()
{
glClearColor(1.0,1.0,1.0,1.0);
glMatrixMode(GL_PROJECTION);
glOrtho(-60,60,-60,60,-60,60);
}



int main(int argc,char **argv)
```

```
{
glutInit(&argc,argv);

glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB/*|GLUT_DEPTH*/);
glutInitWindowSize(1000,1000);
glutInitWindowPosition(0,0);
glutCreateWindow("Human Digestive System");
init();

glutDisplayFunc(mydisplay);
glutKeyboardFunc(myKeyboardFunc);
glutCreateMenu(mymenu);
glutAddMenuEntry("Digestive system",2);
glutAddMenuEntry("Description",5);
glutAddMenuEntry("EXIT",6);
glutAttachMenu(GLUT_RIGHT_BUTTON);
glutKeyboardFunc(myKeyboardFunc);
;
glutMainLoop();
return 0;
}
}
```

CHAPTER 5

TESTING AND RESULTS

5.1 DIFFERENT TYPES OF TESTING

1. Unit Testing

Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components.

2. Module Testing

A module is a collection of dependent components such as a object class, an abstract Data type or some looser collection of procedures and functions. A module related Components, so can be tested without other system modules.

3. System Testing

This is concerned with finding errors that result from unanticipated interaction between Sub-system interface problems.

4. Acceptance Testing

The system is tested with data supplied by the system customer rather than simulated test data.

5.2 TEST CASES

The test cases provided here test the most important features of the project.

Table 5.2.1: Test Case

Test case	Test-Case	Input	Actual	Expected	Remark
ID	description		output	output	
1	Display of digestive system with labeling	Right mouse button	digestive system displayed	digestive system should be displayed with labeling	Pass
2	Food particle movement in digestive system	Right mouse button	Food particle moves	Food particle movement	Pass
3	Display of respiratory system with labeling	Right mouse button	respiratory system displayed	respiratory system should be displayed with labeling	Pass
4	expansion and contraction of lungs	Right mouse button	Lungs size expands and then contracts accordingly "inhale" and "exhale" texts are displayed	expansion and contraction of lungs	Pass
5	Display of excretory system with labelling	Right mouse button	Excretory system displayed	excretory system should be displayed with labelling	Pass

CHAPTER 6

SNAPSHOTS

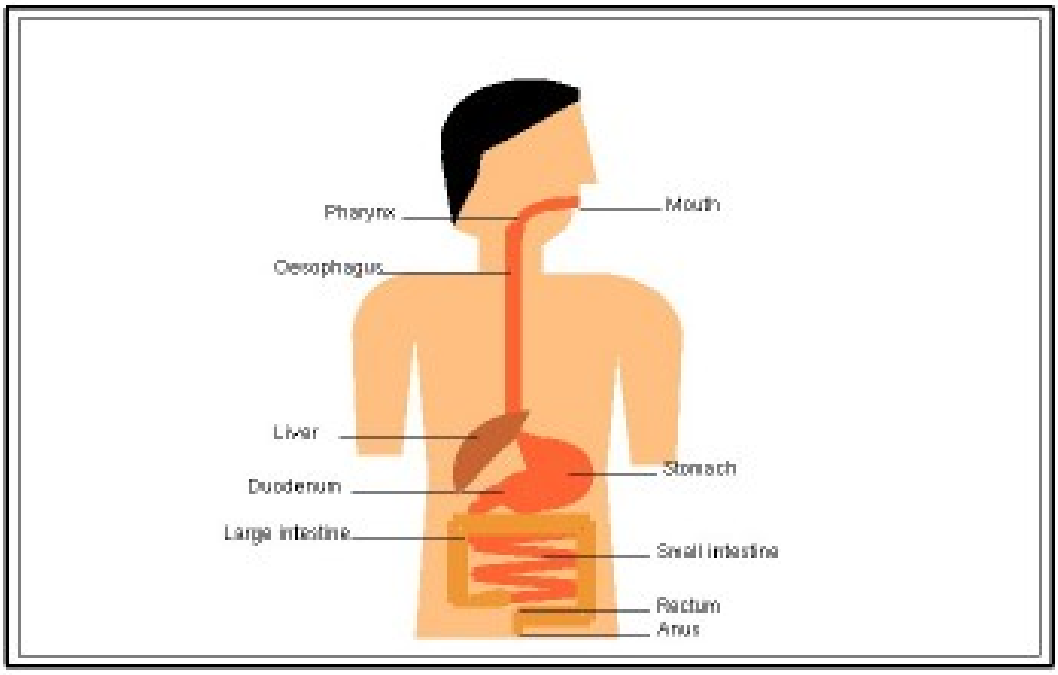


Figure 6.1

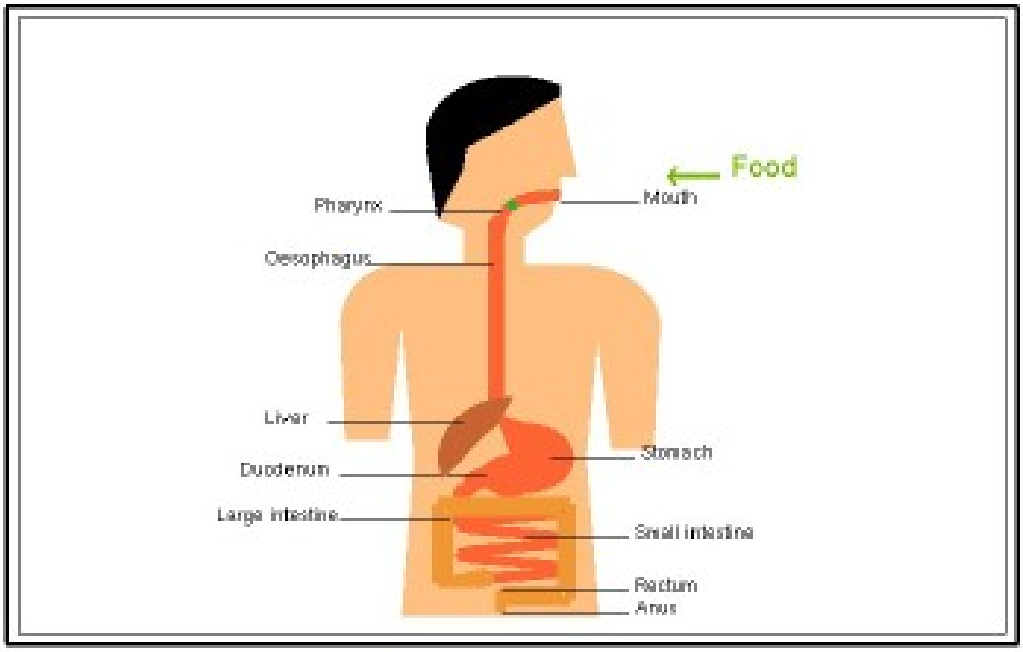


Figure 6.2

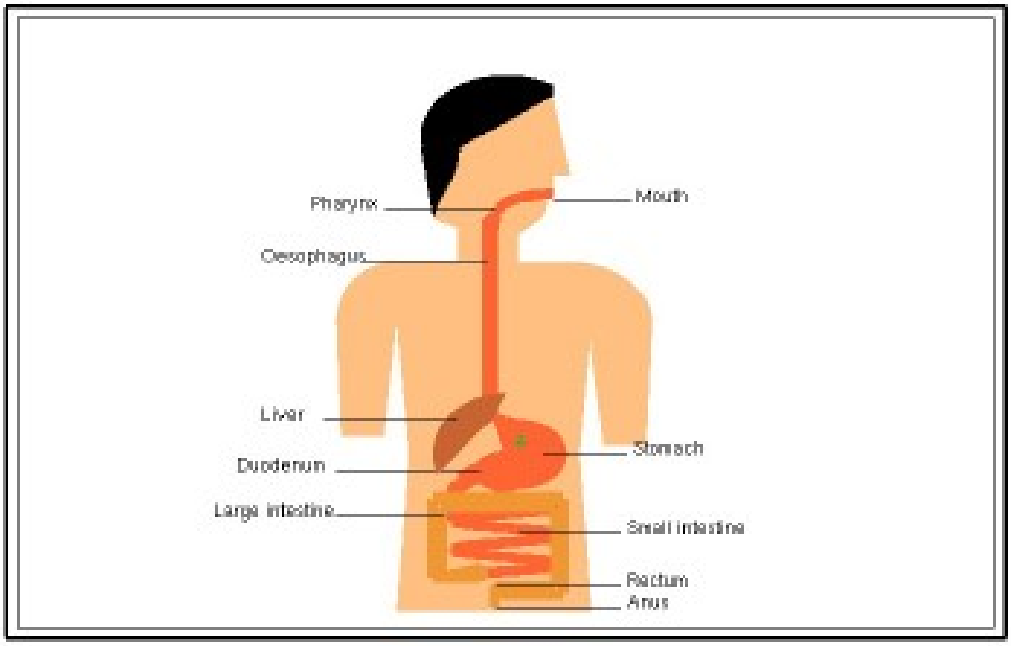


Figure 6.3

CONCLUSION

This project is an effort in the development of a Graphical Software package which is the building block of graphical application. During the development of this package effort lead to understanding the display of geometric primitives like rectangles, lines, line loops, polygon, modes of display, features like translation were also designed. Various functions and operations of the graphical library provide the learning platform to get the maximum performance of the OpenGL functions.

The project "HUMAN DIGESTIVE SYSTEM" has been successfully implemented using OpenGL. The illustration of graphical principles and OpenGL features are included and application program is efficiently developed.

This project enlightens the basic idea of scaling, rotation and translation of the objects. Since it uses interaction with both keyboard and mouse it is sufficiently easy for any kind of end user to run it.

However user requirement changes and provision for changes in design has been allowed. If no major requirements are demanded from the user the current design holds. Thus this project meets the basic requirements successfully and is flexible in all respects to one and all. On conclusion this mini project is implemented with the standard OpenGL functions.

BIBLIOGRAPHY

Books

- Edward Angel: Interactive Computer Science A Top-Down Approachwith OpenGL, 5th Edition, Pearson Education 2008.
- James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, Computer Graphics, Pearson Education 1997.
- Donald Hearn, Pauline Baker: Computer Graphics - OpenGL Version, 3rd Edition, Pearson Education 2004.
- Programming Guide, 5thEdition.

Websites & Links

- http://www.wikipedia.org/
- http://www.opengl.org/
- http://www.khronos.org/
- http://www.opengl-tutorial.org