**CHAPTER 1**

**INTRODUCTION**

**1.1 Computer Graphics**

Computer Graphics is concerned with all the aspects of producing pictures or images using a computer. The field began humbly almost fifty years ago with the display of few lines on a cathode ray tube (CRT), Now we can create images with computer that are indistinguishable from photographs of real objects.

The development of Computer Graphics has been driven by both the needs of the user community and by the advances in hardware and software. The applications of Computer Graphics are many and varied. However, we can divide them into 7 major areas -

* Display of Information.
* Design.
* Simulation and Animation.
* User Interfaces.
* Graphs and Charts, CAD, Data Visualizations
* Image Processing, Education and Training, Entertainment Etc.

**1.2 Introduction to OpenGL**

It is a software interface to graphics hardware. OpenGL is an industry standard portable 3-D Computer Graphics API. OpenGL is a premiere environment for developing portable interactive 2D or 3D applications.

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 three-dimensional 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, here it competes with Direct3D on Microsoft Windows platforms (see Direct3D vs. OpenGL). OpenGL is managed by the non-profit technology consortium, the Khronos Group.

Functions in the main GL library have names that begin with the letters gl and are stored in a library usually referred to as GL. The second is the Open GL Utility Library(GLU). This library uses only GL functions but contains code for creating common objects and simplifying viewing.

**1.2.1 OpenGL for a Developer**

* OpenGL is not a PL but it contains pre-defined functionality.
* Provides functions to set or get or change the state variables.
* Provides functions to render scene onto a buffer which can then be shown in a window.
* Platform to create simple objects and animate them using various functions.
* Provides functions to develop more artistic options for a certain approach.
* Provides functions to explore multiple graphics related concepts.
* Platform Independent
* Open Graphics Library is a cross-language, cross-platform application programming interface for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit, to achieve hardware-accelerated rendering.

**CHAPTER 2**

**System Analysis**

**2.1 Existing System**

**2.2 Proposed System**

**2.2.1 Aim of the Project**

**2.2.2 Project Modules**

**CHAPTER 3**

#### REQUIREMENTS SPECIFICATIONS

**2.1Hardware Requirements**

1. Processor: 1.5 GHz or faster processor.

2. Processor Speed: Intel Pentium 4, Penium M, Pentium D processor or better, or AMD K-8 (Athlon) or better.

3. RAM: 256 MB internal RAM.

4. Graphics Tools: OpenGL 2.0 and later.

5. Windows 7, Windows Vista\*\*, Windows XP Professional or Windows XP Home.

6. NVIDIA GeForce FX 5200 or better graphics card.

7. Monitor resolution: A color monitor with a minimum resolution of 640\*480

## **2.2 Software Requirements**

1. Operating system : MS-DOS based operating system like Windows 98.

2. Other Software : C/C++.

3. Graphics Tool : OpenGL.

4. Alternative Graphics Tools: JAVA 3D, PHIGS, GKS etc

**CHAPTER 4**

**System Design & System Implementation**

**SOURCE CODE: CODE FOR EACH KEY**

#include<stdlib.h>

#include<GL/glut.h>

#include<stdio.h>

#include<string.h>

#include<math.h>

#define SCENE 10

int request[9]={0},counter[3]={0},pages[3]={0}, fault[9]={0}, colour[9]={1,1,1,1,1,1,1,1,1}, pagecolour[3]={1,1,1};

float assign[9]={-5.5,-5.5,-5.5,-5.5,-5.5,-5.5,-5.5,-5.5,-5.5};

int dest=0, showresult=0;

int step=-1, startani=0, faults=0;

char res[]="No. of page faults = ";

float bgcolor[][3]={{1,0,0},{0,1,0},{0,0,1}};

int bgpointer=0;

float tilecolor[][3]={{1,1,0},{1,0.7,0.7},{0,1,1}};

int tilepointer=0;

void \*fonts[]=

{

GLUT\_BITMAP\_9\_BY\_15,

GLUT\_BITMAP\_TIMES\_ROMAN\_10,

GLUT\_BITMAP\_TIMES\_ROMAN\_24,

GLUT\_BITMAP\_HELVETICA\_12,

GLUT\_BITMAP\_HELVETICA\_10,

GLUT\_BITMAP\_HELVETICA\_18,

};

void output(int x,int y,char \*string,int j)

{

int len,i;

glColor3f(1.0f,0.0f,0.0f);

glRasterPos2f(x,y);

len=(int) strlen(string);

for(i=0;i<len;i++)

glutBitmapCharacter(fonts[j],string[i]);

}

void front\_page()

{

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glMatrixMode(GL\_MODELVIEW);

glPushMatrix();

output(200,500,"Hirasugar Institute Of Technology",2);

output(320,450," Nidasoshi ",2);

output(240,400,"Subject: Computer Graphics & Visualization Lab (18CSL67)",2);

output(280,300,"MINI PROJECT ON",2);

output(220,250,"Simulation of LRU using OpenGL",2);

output(100,150,"Guide:",3);

output(135,130,"Prof.S.G.Gollagi",5);

output(450,150,"By:",3);

output(475,130,"Girish G. Hiremath[2hn18cs008]",5);

output(475,100,"Geetanjali V. Patil[2hn18cs007]",5);

output(475,100,"Ashwini A. Dodabhangi[2hn18cs004]",5);

output(275,50,"-> PRESS S to START",2);

glFlush();

glCallList(SCENE);

glPopMatrix();

glutSwapBuffers();

}

void init()

{

glColor3f(0,0,0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0,800,0,600);

glMatrixMode(GL\_MODELVIEW);

}

int getLRU()

{

if(counter[0]>=counter[1] && counter[0]>=counter[2]) return 0;

if(counter[1]>=counter[0] && counter[1]>=counter[2]) return 1;

//if(counter[2]>=counter[1] && counter[2]>=counter[3]) return 2;

return 2;

}

void tile(float x, float y, char n)

{

float size=20;

//glColor3f(1,1,0);

glBegin(GL\_POLYGON);

glVertex2f(x-size, y-size);

glVertex2f(x+size, y-size);

glVertex2f(x+size, y+size);

glVertex2f(x-size, y+size);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_LINE\_LOOP);

glVertex2f(x-size, y-size);

glVertex2f(x+size, y-size);

glVertex2f(x+size, y+size);

glVertex2f(x-size, y+size);

glEnd();

glRasterPos2f(x-size/2, y);

glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15, n);

}

void draw\_flag()

{

glColor3fv(bgcolor[bgpointer]);

glBegin(GL\_POLYGON);

glVertex2f(-10,-10);

glVertex2f(10,-10);

glVertex2f(10,10);

glVertex2f(-10,10);

glEnd();

}

void setup\_request()

{

int i;

glColor3fv(bgcolor[bgpointer]);

glBegin(GL\_POLYGON);

glVertex2f(0,0);

glVertex2f(700,0);

glVertex2f(700,100);

glVertex2f(0,100);

glEnd();

glPushMatrix();

glTranslatef(-10, 40, 0);

for(i=0; i<9; i++)

{

glColor3fv(tilecolor[tilepointer]);

glTranslatef(70, 0, 0);

glPushMatrix();

if(assign[i]>-4.5)

{

glTranslatef(70\*step-70\*i,0,0);

}

glTranslatef(0, -250-45\*assign[i], 0);

if((int)assign[i]==dest && assign[i]>=0)

glColor3f(1,0.3,0.3);

else

//glColor3f(colour[i],1,0);

glColor3fv(tilecolor[tilepointer]);

tile(10,10,request[i]+'0');

glPopMatrix();

if(fault[i])

{

glPushMatrix();

glTranslatef(0, -385, 0);

draw\_flag();

glPopMatrix();

}

}

glPopMatrix();

}

void drawText(char \*string,float x,float y,float z)

{

char \*c;

glRasterPos3f(x, y,z);

for (c=string; \*c != '\0'; c++)

{

if(\*c=='\n')

glRasterPos3f(x, y-0.05,z);

else

glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15, \*c);

}

}

void setup\_pages()

{

glPushMatrix();

//glColor3f(fabs(sin(ycolour[1])), 1, 0);

// glColor3f(1, pagecolour[0], 0);

tile(0,0,pages[0]==0?' ':pages[0]+'0');

glTranslatef(0, -45, 0);

// glColor3f(fabs(sin(ycolour[2])), 1, 0);

//glColor3f(1, pagecolour[1], 0);

tile(0,0,pages[1]==0?' ':pages[1]+'0');

glTranslatef(0, -45, 0);

// glColor3f(fabs(sin(ycolour[3])), 1, 0);

//glColor3f(1, pagecolour[2], 0);

tile(0,0,pages[2]==0?' ':pages[2]+'0');

glPopMatrix();

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

glPushMatrix();

glTranslatef(120+70\*step,200,0);

setup\_pages();

glPopMatrix();

glColor3f(1,1,0);

glPushMatrix();

glTranslatef(50, 400, 0);

setup\_request();

glPopMatrix();

glEnd();

if(showresult && step==8)

{

glColor3f(0,0,0);

res[21]=faults+'0';

drawText(res, 50, 20, 0);

}

drawText("LRU Page Replacement Algorithm", 250, 550, 0);

glFlush();

glutSwapBuffers();

}

void idle()

{

if(!startani)

return;

if(dest>assign[step])

assign[step]+=0.01;

if(dest<=assign[step])

{

if(fault[step])

pages[dest]=request[step];

startani=0;

dest=-10;

showresult=1;

}

display();

}

void mouse(int btn,int state,int x, int y)

{

int n,i,j;

if(startani==1)

return;

if(btn==GLUT\_LEFT\_BUTTON && state==GLUT\_DOWN &&step<8)

{

if(step<9)

step++;

for(i=0;i<3;i++)

{

if(request[step]==pages[i])

{

fault[step]=0;

counter[i]=0;

for(j=0;j<3;j++)

if(j!=i) counter[j]++;

//dest=i;

dest=-5;

startani=1;

colour[step]=0;

glutPostRedisplay();

return;

}

}

n=getLRU();

counter[n]=0;

for(i=0;i<3;i++)

if(i!=n)

counter[i]++;

//pagecolour[n]=0;

//assign[step]=n;

dest=n;

startani=1;

fault[step]=1;

faults++;

}

glutPostRedisplay();

}

void handle\_bg\_colour(int action)

{

bgpointer=action;

glutPostRedisplay();

}

void handle\_tile\_colour(int action)

{

tilepointer=action;

glutPostRedisplay();

}

void menu(int action)

{

if(action==0)

exit(0);

}

void addMenu()

{

int submenu1, submenu2;

submenu1 = glutCreateMenu(handle\_bg\_colour);

glutAddMenuEntry("Red",0);

glutAddMenuEntry("Green",1);

glutAddMenuEntry("Blue",2);

submenu2 = glutCreateMenu(handle\_tile\_colour);

glutAddMenuEntry("Yellow",0);

glutAddMenuEntry("Light Red",1);

glutAddMenuEntry("Light Blue",2);

glutCreateMenu(menu);

glutAddSubMenu("Background Colour",submenu1);

glutAddSubMenu("Tile Colour",submenu2);

glutAddMenuEntry("Quit",0);

glutAttachMenu(GLUT\_RIGHT\_BUTTON);

}

void display1()

{

glClearColor(1.0,1.0,1.0,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

front\_page();

glFlush();

glutSwapBuffers();

}

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

{

switch(key)

{

case 'S':

case 's':glutDisplayFunc(display);

break;

case 'Q':

case 'q':exit(0);

}

glutPostRedisplay();

}

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

{

glutInit(&argc,argv);

int i;

printf("Enter a sequence of 9 numbers for page request\n");

for(i=0; i<9; i++)

scanf("%d", &request[i]);

glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB);

//glutInitWindowSize(640,480);

glutInitWindowSize(1000,800);

glutCreateWindow("COH");

glutKeyboardFunc(key);

glutDisplayFunc(display1);

//glutDisplayFunc(display);

glutMouseFunc(mouse);

glutIdleFunc(idle);

glClearColor(1,1,1,1);

init();

addMenu();

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0,800,0,600);

glMatrixMode(GL\_MODELVIEW);

glutMainLoop();

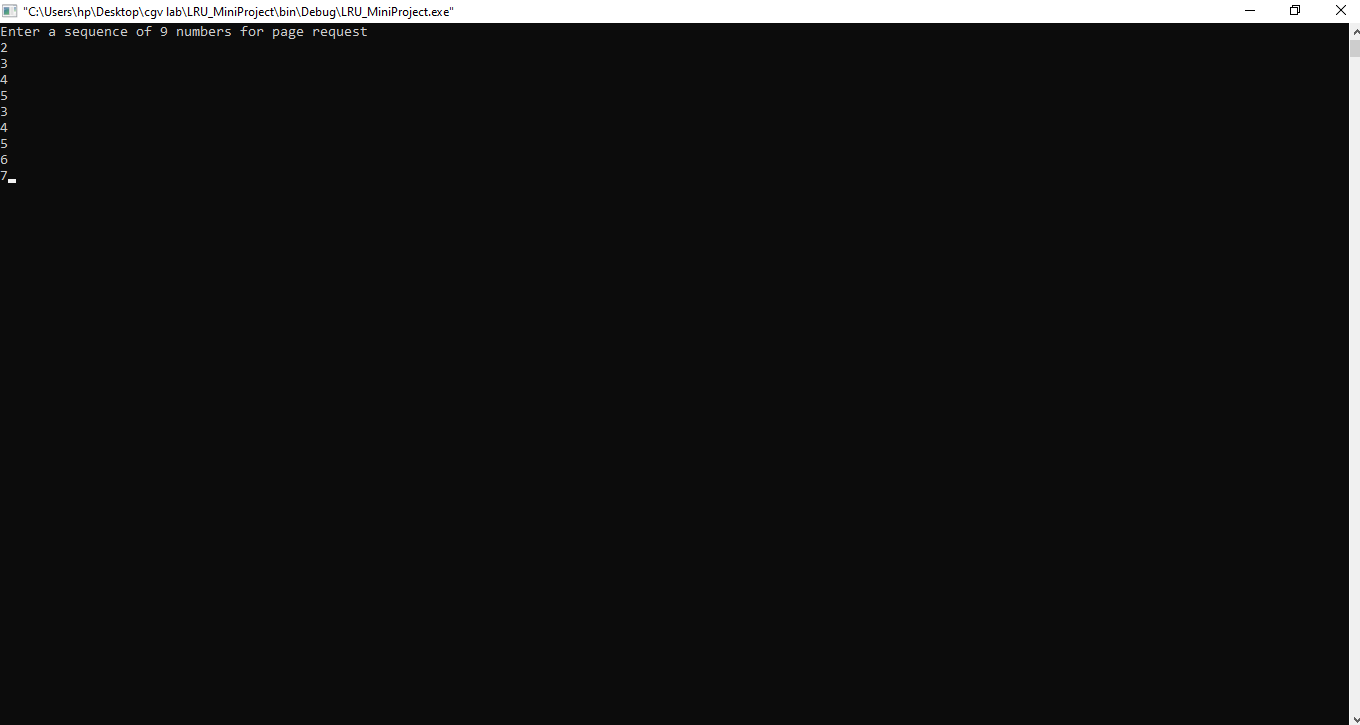
return 0;

}

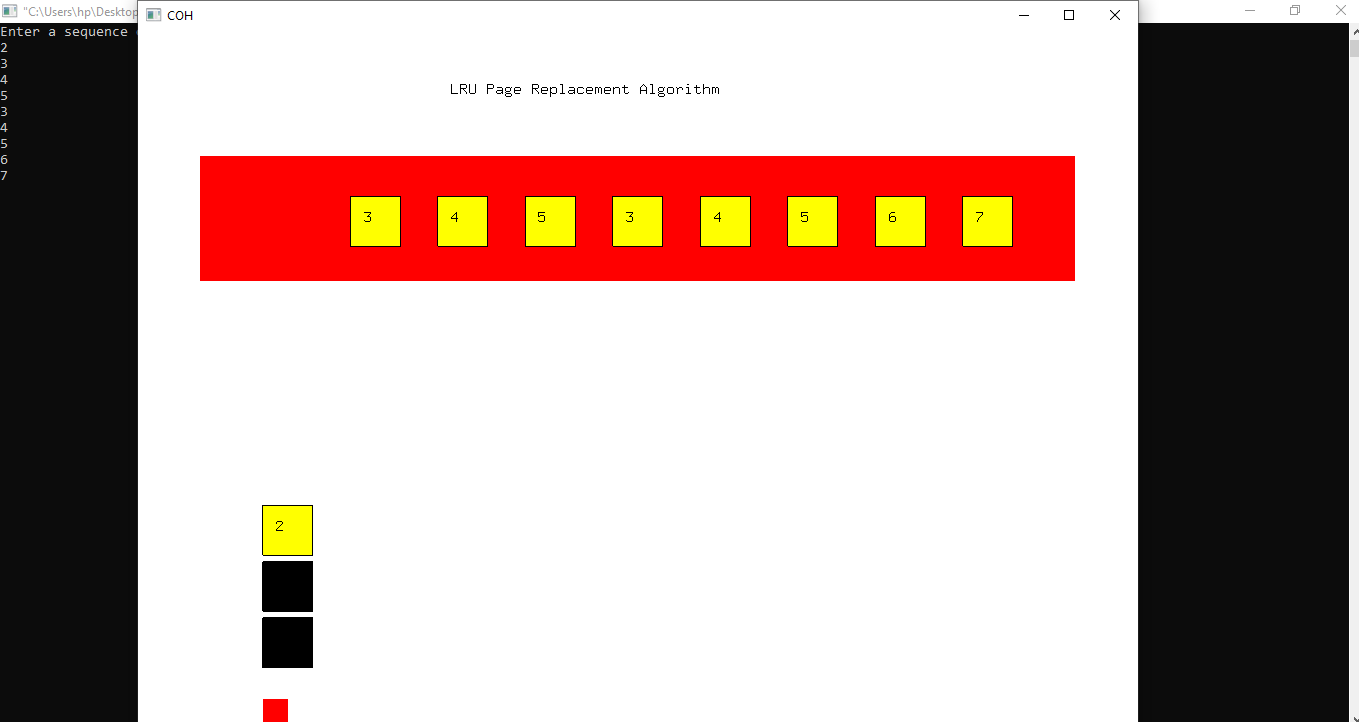
4.1 Modular Description

**CHAPTER 5**

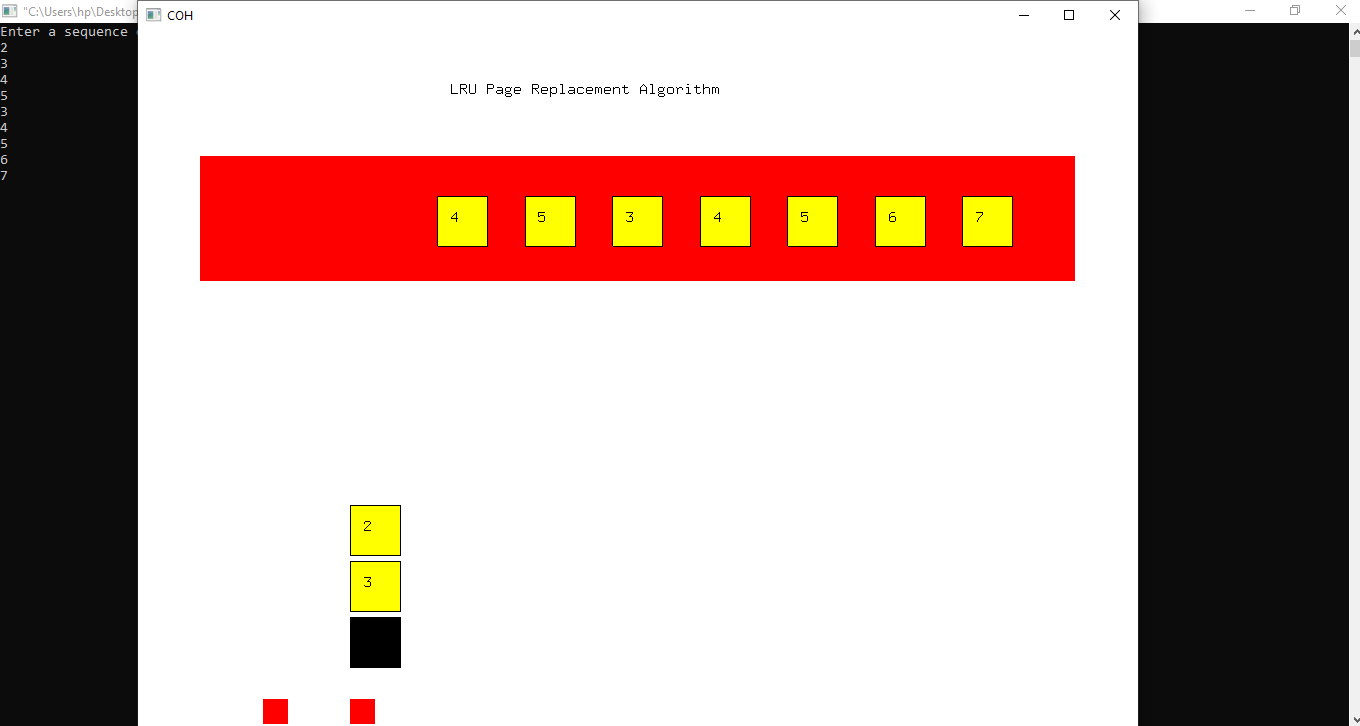
**Sample output**

****

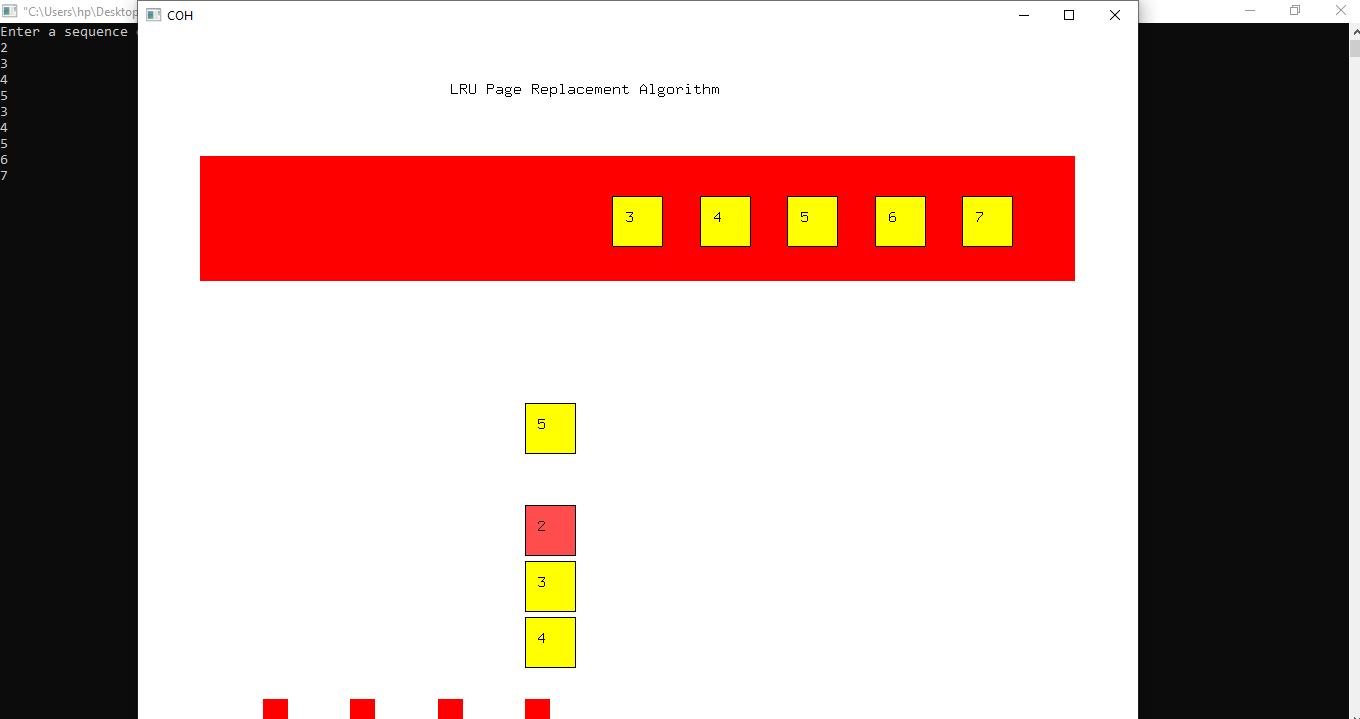
**5.1**

****

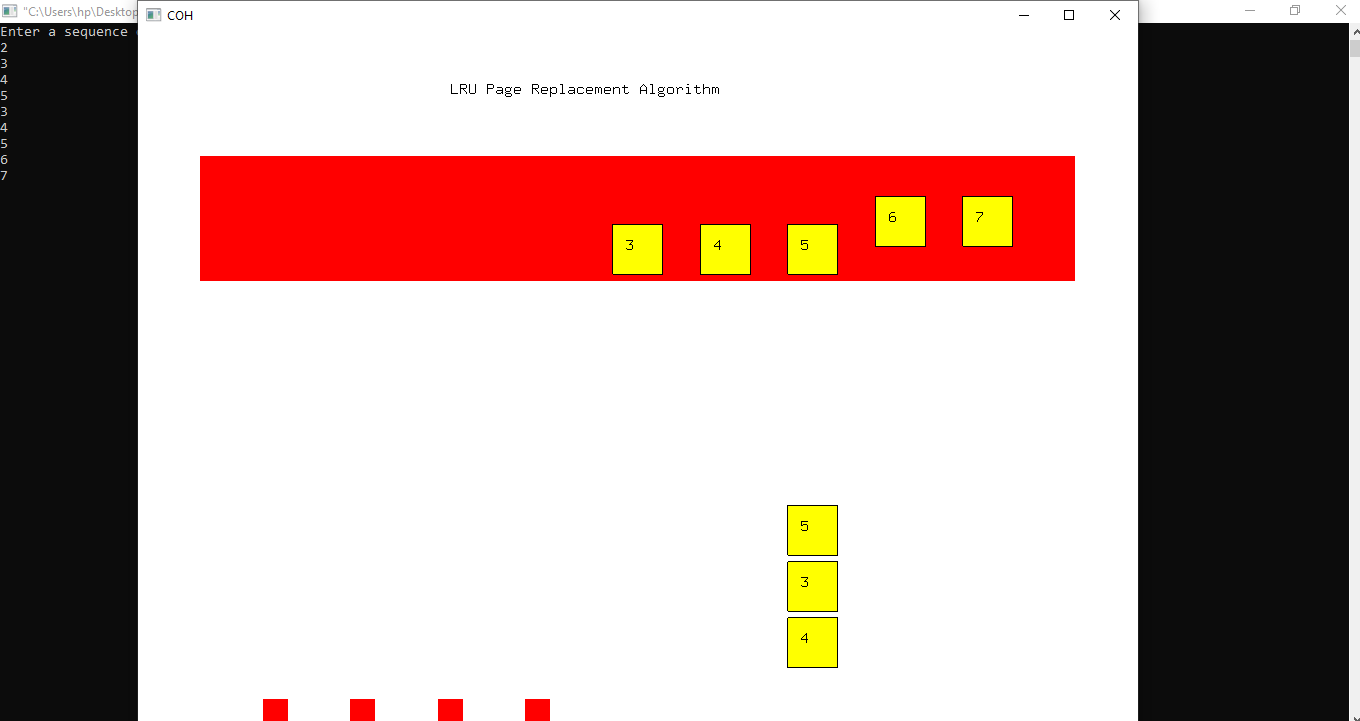
**5.2**

****

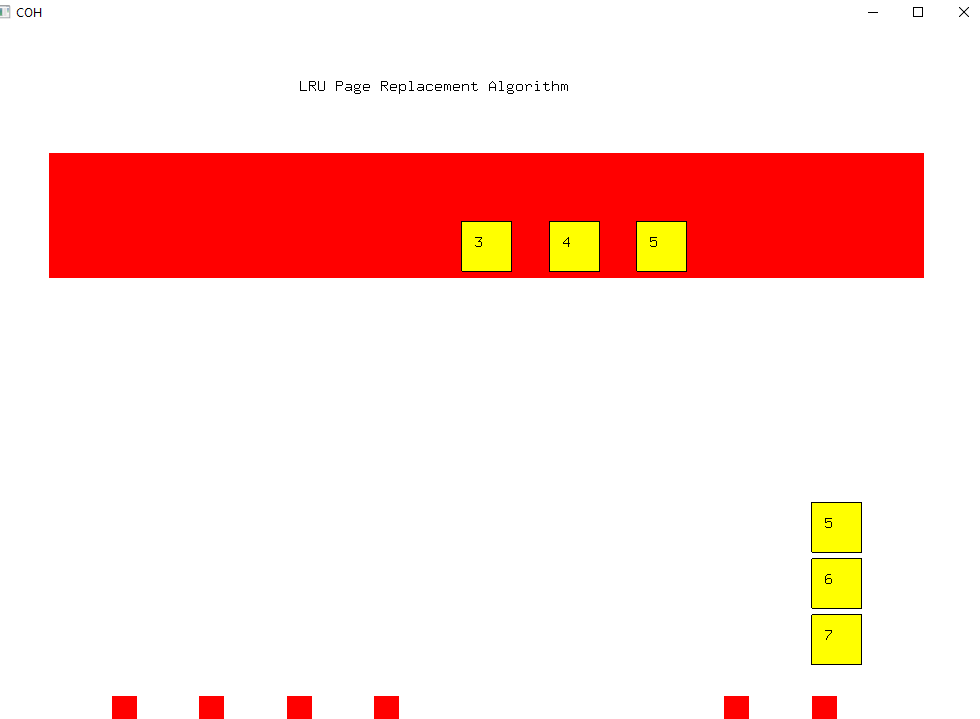
**5.3**

****

**5.4**

****

**5.5**



**5.6**

**Chapter 6**

**CONCLUSION**

As a part of our academics’ mini project we have successfully implemented the concept of ray casting using OpenGL graphics package. We have tried to use the many of the graphics concepts using OpenGL such as translation, rotation, scaling,viewer motions etc.We would like to end by saying that, this project has helped us a lot to learn more about OpenGL ,OpenGL APIs and its functions. It was a very good experience in developing visual of Objects In Multimirror which helped us in improving our knowledge about OpenGL.

**Chapter 7**

**Bibliography**

[1] Edward Angel, “Interactive Computer Graphics A top-down approach using OpenGL”, 5th editon

[2] Donald Hearn and Pauline Baker: “Computer Graphics- OpenGL Version”, 3rd edition using OpenGL” 3rd edition

[3]F.S.Hill Jr.:”Computer Graphics

**WEBSITES**

[1] www.OpenGL.org Redbook.

[2][www.OpenGL.org](http://www.OpenGL.org) simple example

[3]<http://www.files32.com/Types-Of-Clipping-In-Computer-Graphics.asp>

[4]<http://www.opengl.org/code>