



NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)

Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC

☎: 08258 - 281039 - 281263, Fax: 08258 - 281265

Report on Mini Project

“Simulation of Railway Station”

Course Code: 18CS607

Course Name: Computer Graphics Lab

Semester: 6

Section: B

Submitted To:

Mr. Pradeep Kanchan

Assistant Professor Gd-III

Department of Computer Science and Engineering

NMAM Institute of Technology, Nitte

Submitted By:

Name: Nidhi Rai

USN: 4NM18CS103

Name: Mahima Biswas

USN: 4NM18CS087

Date of submission: 13/05/2021

Signature of Course Instructor

ABSTRACT

Computer Graphics is a core technology in digital photography, film, video games, computer displays, and many specialized applications. Modelling, Geometric processing, rendering geometric transformation, visibility, simulation of light Animation of Lifelike characters, natural phenomena, their interactions and surrounding environments are some of the applications of Computer Graphics.

Simulation of Railway Station is a newly designed computer graphics mini project. This mini project begins with displaying a railway station and passengers waiting on the platform. When the signal turns red the train stops in the station and receives the passengers and departures when the signal is green. We also have night mode and day mode setup.

The project also consists of the train compartment which displays the interior along with the passengers. There are variety of scenic views like hills and rivers which can be changed as desired by the user. We have added music which enhances the overall project. This project is developed in C/C++ in Code Blocks. Keyboard and Mouse are the two input devices used which will also allow the user to interact with the application.

TABLE OF CONTENTS

TITLE	PAGE NO
1. ABSTRACT	2
2. INTRODUCTION	4
3. IMPLEMENTATION	5 - 54
4. CONCLUSION	55
5. REFERENCE	56
6. APPENDIX	57 - 61

INTRODUCTION

Computer 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.

OpenGL, short for “Open Graphics Library” is an application programming interface (API) designed for rendering 2D and 3D graphics. 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. It provides a common set of commands that can be used to manage graphics in different applications and on multiple platforms.

OpenGL serves for two main purposes firstly, to hide the complexities of interfacing with different 3D accelerators, by presenting programmer with a single, uniform API. Secondly, 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.

IMPLEMENTATION

The project “Simulation of railway station” is created by using a set of functions for the respective components of the railway station.

1: void display() function displays the names in the introductory window:

```
void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0,0.0,0.0);
    drawstring(400,515,0.0,"Simulation of Railway Station");
    glColor3f(1.0,0.0,0.0);
    drawstring(250,280,0.0,"Mahima Biswas");
    drawstring(250,250,0.0,"Nidhi Rai");
    glColor3f(1.0,0.0,0.0);
    drawstring(600,280,0.0,"4NM18CS087");
    drawstring(600,250,0.0,"4NM18CS103");
    drawstring(380,150,0.0,"*Click on this page and Press enter*");
    glFlush();
}
```

2: void man() has the code for the men standing in platform and compartment.

```
void man()
{
    face(320,271);
    glColor3ub(0,0,0);
    glBegin(GL_LINES);
        glVertex2i(540-220,195+76);
        glVertex2i(540-220,190+76); //nose
        glVertex2i(531-220,200+76);
```

```

        glVertex2i(537-220,200+76);//eyebrow
        glVertex2i(543-220,200+76);
        glVertex2i(549-220,200+76);//eyebrow

    glEnd();

//ear right
    glBegin(GL_POLYGON);
    glColor3ub(255,191,128);

    glVertex2i(540-14-220,194+1+76);
    glVertex2i(540-14-220,190+1+76);

    glVertex2i(538-14-220,189+1+76);
    glVertex2i(538-14-220,195+1+76);

    glEnd();

//ear left
    glBegin(GL_POLYGON);
    glColor3ub(255,191,128);

    glVertex2i(554-220,195+76);
    glVertex2i(556-220,196+76);
    glVertex2i(556-220,191+76);
    glVertex2i(554-220,190+76);

    glEnd();

//hair
    glBegin(GL_POLYGON);
    glColor3ub(0,0,0);

    glVertex2i(527-220,203+76);
    glVertex2i(553-220,203+76);
    glVertex2i(547-220,210+76);
    glVertex2i(533-220,210+76);

    glEnd();

// eyes

```

```

        glBegin(GL_POLYGON);

            glVertex2i(533-220,198+76);
            glVertex2i(535-220,198+76);
            glVertex2i(535-220,196+76);
            glVertex2i(533-220,196+76);
            glEnd();

glBegin(GL_POLYGON);

    glVertex2i(545-220,198+76);
    glVertex2i(547-220,198+76);
    glVertex2i(547-220,196+76);
    glVertex2i(545-220,196+76);

glEnd();

    // mouth
glBegin(GL_POLYGON);

    glVertex2i(535-220,187+76);
    glVertex2i(540-220,185+76);
    glVertex2i(545-220,187+76);
    glVertex2i(540-220,187+76);

glEnd();

//beard
glBegin(GL_POLYGON);

    glColor3ub(0,0,0);
    glVertex2i(538-220,180+76);
        glVertex2i(542-220,180+76);
        glVertex2i(542-220,184+76);
        glVertex2i(538-220,184+76);
glEnd();

```

```

glBegin(GL_POLYGON);

    glVertex2i(575-220,153+76);
    glVertex2i(556-220,138+76);
    glVertex2i(556-220,145+76);
    glVertex2i(567-220,154+76);

glEnd();

glBegin(GL_POLYGON);

    glVertex2i(515-220,168+76);
    glVertex2i(505-220,153+76);
    glVertex2i(513-220,154+76);
    glVertex2i(518-220,162+76);

glEnd();

glBegin(GL_POLYGON);

    glVertex2i(505-220,153+76);
    glVertex2i(524-220,138+76);
    glVertex2i(524-220,145+76);
    glVertex2i(513-220,154+76);

glEnd();

// belt

glBegin(GL_POLYGON);

    glColor3ub(150,12,30);
    glVertex2i(556-220,145+76);
    glVertex2i(524-220,145+76);
    glVertex2i(524-220,140+76);
    glVertex2i(524-220,140+76);
    glVertex2i(556-220,140+76);

glEnd();

```



```
// collar

    glBegin(GL_POLYGON);

        glColor3ub(200,140,110+76);

        glVertex2i(529-220,180+76);

        glVertex2i(551-220,180+76);

        glVertex2i(546-220,170+76);

        glVertex2i(534-220,170+76);

glEnd();


    glBegin(GL_TRIANGLES);

        glColor3ub(20,140,110);

        glVertex2i(540-220,177+76);

        glVertex2i(545-220,170+76);

        glVertex2i(535-220,170+76);

glEnd();

//buttons

    glColor3ub(0,0,0);

    glPushMatrix();

    glTranslate(540-220,165+76,0);

    glutSolidTorus(1,1,100,90);
```

```
        glPopMatrix();

    glPushMatrix();

    glTranslatef(540-220,158+76,0);

    glutSolidTorus(1,1,100,90);

    glPopMatrix();

    glPushMatrix();

    glTranslatef(540-220,151+76,0);

    glutSolidTorus(1,1,100,90);

    glPopMatrix();
```

```

    // pant
glBegin(GL_POLYGON);
    glColor3ub(80,80,230);
glVertex2i(555-220,140+76);
glVertex2i(525-220,140+76);
glVertex2i(520-220,105+76);
glVertex2i(530-220,105+76);
glVertex2i(533-220,138+76);
glVertex2i(550-220,105+76);
glVertex2i(560-220,105+76);
    glEnd();
    //shoe left
glBegin(GL_POLYGON);
    glColor3ub(100,10,10);
glVertex2i(530-220,105+76);
glVertex2i(530-220,96+76);
glVertex2i(512-220,96+76);
glVertex2i(520-220,105+76);
    glEnd();
    //shoe right
glBegin(GL_POLYGON);
    glColor3ub(100,10,10);
glVertex2i(550-220,105+76);
glVertex2i(550-220,96+76);
glVertex2i(568-220,96+76);
glVertex2i(560-220,105+76);
    glEnd();
}

```

3: void women() function has the code of the women seen waiting on the platform. They are also present in the train compartment.

```
void women()
{
    face(540,295);
        glColor3ub(0,0,0);
        glBegin(GL_LINES);
            glVertex2i(540,294);
            glVertex2i(540,290); //nose
            glVertex2i(531,298);
            glVertex2i(532,299);
            glVertex2i(532,299);
            glVertex2i(537,298); //eyebrow
            glVertex2i(549,298);
            glVertex2i(548,299);
            glVertex2i(548,299);
            glVertex2i(543,298); //eyebrow
        glEnd();
    //ear right
        glBegin(GL_POLYGON);
        glColor3ub(255,191,128);
        glVertex2i(540-14,294+1);
        glVertex2i(540-14,290+1);
        glVertex2i(538-14,289+1);
        glVertex2i(538-14,295+1);
        glEnd();
    //ear left
        glBegin(GL_POLYGON);
```

```

        glColor3ub(255,191,128);
        glVertex2i(554,295);
        glVertex2i(556,296);
        glVertex2i(556,291);
        glVertex2i(554,290);
        glEnd();
        //ear ring right
        glBegin(GL_POLYGON);
        glColor3ub(255,85,90);
        glVertex2i(539-14,292);
        glVertex2i(542-14,285);
        glVertex2i(536-14,285);
        glEnd();
        //ear ring left
        glBegin(GL_POLYGON);
        glColor3ub(255,85,90);
        glVertex2i(551,285);
        glVertex2i(555,292);
        glVertex2i(558,285);
        glEnd();
//hair
        glBegin(GL_POLYGON);
        glColor3ub(0,0,0);
        glVertex2i(525,299);
        glVertex2i(549,309);
        glVertex2i(540,312);
        glVertex2i(528,307);
        glEnd();
        glBegin(GL_POLYGON);

```

```

        glColor3ub(0,0,0);
        glVertex2i(540,307);
        glVertex2i(549,309);
        glVertex2i(552,307);
        glVertex2i(555,299);
        glEnd();
        // eyes
        glBegin(GL_POLYGON);
            glVertex2i(533,296);
            glVertex2i(535,296);
            glVertex2i(535,294);
            glVertex2i(533,294);
        glEnd();
        glBegin(GL_POLYGON);
            glVertex2i(545,296);
            glVertex2i(547,296);
            glVertex2i(547,294);
            glVertex2i(545,294);
        glEnd();
        //mouth
        glBegin(GL_POLYGON);
            glColor3ub(150,50,50);
            glVertex2i(534,287);
            glVertex2i(540,284);
            glVertex2i(546,287);
            glVertex2i(540,285);
        glEnd();
        //shirt
        glBegin(GL_POLYGON);

```

```
        glColor3ub(204,0,204);
    glVertex2i(529,280);
        glVertex2i(551,280);
        glVertex2i(566,269);
        glVertex2i(561,260);
        glVertex2i(556,265);
    glVertex2i(556,245);
        glVertex2i(524,245);
        glVertex2i(524,265);
        glVertex2i(519,260);
        glVertex2i(514,269);
glEnd();
//neck
    glBegin(GL_POLYGON);
        glColor3ub(255,190,128);
        glVertex2i(533,280);
        glVertex2i(547,280);
        glVertex2i(545,271);
        glVertex2i(535,271);
    glEnd();
//hands
        glBegin(GL_POLYGON);
            glColor3ub(255,191,128);
            glVertex2i(565,268);
            glVertex2i(575,253);
            glVertex2i(567,254);
            glVertex2i(562,262);

        glEnd();
        glBegin(GL_POLYGON);
```

```

        glVertex2i(575,253);
        glVertex2i(556,238);
        glVertex2i(556,245);
        glVertex2i(567,254);
glEnd();
glBegin(GL_POLYGON);
        glVertex2i(515,268);
        glVertex2i(505,253);
        glVertex2i(513,254);
        glVertex2i(518,262);
glEnd();
glBegin(GL_POLYGON);
        glVertex2i(505,253);
        glVertex2i(524,238);
        glVertex2i(524,245);
        glVertex2i(513,254);
glEnd();
// belt
glBegin(GL_POLYGON);
        glColor3ub(0,0,0);
        glVertex2i(556,245);
        glVertex2i(524,245);
        glVertex2i(524,240);
        glVertex2i(556,240);
glEnd();
// leg
glBegin(GL_POLYGON);
glColor3ub(255,190,128);
glVertex2i(555,240);

```

```

glVertex2i(525,240);
glVertex2i(520,205);
glVertex2i(530,205);
glVertex2i(533,238);
glVertex2i(550,205);
glVertex2i(560,205);
    glEnd();
//skirt
glBegin(GL_POLYGON);
    glColor3ub(0,0,0);
        glVertex2i(524,240);
        glVertex2i(556,240);
        glVertex2i(566,210);
        glVertex2i(514,210);
glEnd();
//shoe left
glBegin(GL_POLYGON);
glColor3ub(180,0,0);
glVertex2i(530,205);
glVertex2i(530,196);
glVertex2i(528,196);
glVertex2i(528,204);
glVertex2i(522,196);
glVertex2i(512,196);
glVertex2i(520,205);
glEnd();
//shoe right
glBegin(GL_POLYGON);
glColor3ub(180,0,0);

```



```

        glVertex2i(550,205);
        glVertex2i(550,196);
        glVertex2i(552,196);
        glVertex2i(552,204);
        glVertex2i(558,196);
        glVertex2i(568,196);
        glVertex2i(560,205);
        glEnd();
    }

```

4: void draw_object() is used to represent the railway station and its surroundings including trees, sky, people, day mode and night mode.

```

void draw_object()
{
    int l;
    if(day==1)
    {
        //sky
        glColor3f(0.0,0.9,0.9);
        glBegin(GL_POLYGON);
        glVertex2f(0,380);
        glVertex2f(0,700);
        glVertex2f(1100,700);
        glVertex2f(1100,380);
        glEnd();

        //sun
        for(l=0;l<=35;l++)
        {
            glColor3f(1.0,0.9,0.0);
            draw_circle(100,625,l);
        }
    }
}

```

```

}

//plane
if(plane==1)
{
    glColor3f(1.0,1.0,1.0);
    glBegin(GL_POLYGON);
    glVertex2f(925+n,625+o);
    glVertex2f(950+n,640+o);
    glVertex2f(1015+n,640+o);
    glVertex2f(1030+n,650+o);
    glVertex2f(1050+n,650+o);
    glVertex2f(1010+n,625+o);
    glEnd();
    glColor3f(0.8,0.8,0.8);
    glBegin(GL_LINE_LOOP);
    glVertex2f(925+n,625+o);
    glVertex2f(950+n,640+o);
    glVertex2f(1015+n,640+o);
    glVertex2f(1030+n,650+o);
    glVertex2f(1050+n,650+o);
    glVertex2f(1010+n,625+o);
    glEnd();
}

//cloud1
for(l=0;l<=20;l++)
{
    glColor3f(1.0,1.0,1.0);
    draw_circle(160+m,625,l);
}

```

```

for(l=0;l<=35;l++)
{
    glColor3f(1.0,1.0,1.0);
    draw_circle(200+m,625,l);
    draw_circle(225+m,625,l);
}
for(l=0;l<=20;l++)
{
    glColor3f(1.0,1.0,1.0);
    draw_circle(265+m,625,l);
}
//cloud2
for(l=0;l<=20;l++)
{
    glColor3f(1.0,1.0,1.0);
    draw_circle(370+m,615,l);
}
for(l=0;l<=35;l++)
{
    glColor3f(1.0,1.0,1.0);
    draw_circle(410+m,615,l);
    draw_circle(435+m,615,l);
    draw_circle(470+m,615,l);
}
for(l=0;l<=20;l++)
{
    glColor3f(1.0,1.0,1.0);
    draw_circle(500+m,615,l);
}

```

```

//grass
glColor3f(0.0,0.9,0.0);
glBegin(GL_POLYGON);
glVertex2f(0,160);
glVertex2f(0,380);
glVertex2f(1100,380);
glVertex2f(1100,160);
glEnd();
}
else
{
//sky
glColor3f(0.0,0.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(0,380);
glVertex2f(0,700);
glVertex2f(1100,700);
glVertex2f(1100,380);
glEnd();
//moon
int l;
    for(l=0;l<=35;l++)
    {
        glColor3f(1.0,1.0,1.0);
        draw_circle(100,625,l);
    }
//star1
glColor3f(1.0,1.0,1.0);
glBegin(GL_TRIANGLES);

```

```

glVertex2f(575,653);
glVertex2f(570,645);
glVertex2f(580,645);
glVertex2f(575,642);
glVertex2f(570,650);
glVertex2f(580,650);
glEnd();
//star2
glColor3f(1.0,1.0,1.0);
glBegin(GL_TRIANGLES);
glVertex2f(975,643);
glVertex2f(970,635);
glVertex2f(980,635);
glVertex2f(975,632);
glVertex2f(970,640);
glVertex2f(980,640);
glEnd();
//comet
if(comet==1)
{
    for(l=0;l<=7;l++)
    {
        glColor3f(1.0,1.0,1.0);
        draw_circle(300+c,675,l);
    }
glColor3f(1.0,1.0,1.0);
glBegin(GL_TRIANGLES);
glVertex2f(200+c,675);
glVertex2f(300+c,682);

```

```

        glVertex2f(300+c,668);
        glEnd();
    }
    //Plane
    if(plane==1)
    {
        for(l=0;l<=1;l++)
        {
            glColor3f(1.0,0.0,0.0);
            draw_circle(950+n,625+o,l);
            glColor3f(1.0,1.0,0.0);
            draw_circle(954+n,623+o,l);
        }
    }
    //grass
    glColor3f(0.0,0.3,0.0);
    glBegin(GL_POLYGON);
    glVertex2f(0,160);
    glVertex2f(0,380);
    glVertex2f(1100,380);
    glVertex2f(1100,160);
    glEnd();
}
//track boundary
glColor3f(1.0,1.0,1.0);
glBegin(GL_POLYGON);
glVertex2f(0,150);
glVertex2f(0,160);
glVertex2f(1100,160);

```

```
glVertex2f(1100,150);
glEnd();
//platform
glColor3f(0.560784,0.560784,0.737255);
glBegin(GL_POLYGON);
glVertex2f(0,160);
glVertex2f(0,250);
glVertex2f(1100,250);
glVertex2f(1100,160);
glEnd();
//table 1
glColor3f(1.0,0.498039,0.0);
glBegin(GL_POLYGON);
glVertex2f(140,190);
glVertex2f(140,210);
glVertex2f(155,210);
glVertex2f(155,190);
glEnd();
glColor3f(0.2,0.2,0.2);
glBegin(GL_POLYGON);
glVertex2f(130,210);
glVertex2f(130,215);
glVertex2f(225,215);
glVertex2f(225,210);
glEnd();
glColor3f(1.0,0.498039,0.0);
glBegin(GL_POLYGON);
glVertex2f(200,190);
glVertex2f(200,210);
```

```

glVertex2f(215,210);
glVertex2f(215,190);
glEnd();
//below track
glColor3f(0.623529,0.623529,0.372549);
glBegin(GL_POLYGON);
glVertex2f(0,0);
glVertex2f(0,150);
glVertex2f(1100,150);
glVertex2f(1100,0);
glEnd();
//Railway track
glColor3f(0.0,0.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(-100,0);
glVertex2f(-100,20);
glVertex2f(1100,20);
glVertex2f(1100,0);
glEnd();
glBegin(GL_POLYGON);
glVertex2f(-100,80);
glVertex2f(-100,100);
glVertex2f(1100,100);
glVertex2f(1100,80);
glEnd();
//track bounbary
glColor3f(0.647059,0.164706,0.164706);
glBegin(GL_POLYGON);
glVertex2f(-100,100);

```



```
glVertex2f(-100,150);
glVertex2f(1100,150);
glVertex2f(1100,100);
glEnd();
//railway station boundary (fence)
glColor3f(0.647059,0.164706,0.164706);
glBegin(GL_POLYGON);
glVertex2f(0,250);
glVertex2f(0,310);
glVertex2f(5,320);
glVertex2f(10,310);
glVertex2f(10,250);
glEnd();
glBegin(GL_POLYGON);
glVertex2f(90,250);
glVertex2f(90,310);
glVertex2f(95,320);
glVertex2f(100,310);
glVertex2f(100,250);
glEnd();
glBegin(GL_POLYGON);
glVertex2f(140,250);
glVertex2f(140,310);
glVertex2f(145,320);
glVertex2f(150,310);
glVertex2f(150,250);
glEnd();
//tree 1
glColor3f(0.9,0.2,0.0);
```

```

glBegin(GL_POLYGON);
glVertex2f(50,185);
glVertex2f(50,255);
glVertex2f(65,255);
glVertex2f(65,185);
glEnd();
for(l=0;l<=30;l++)
{
    glColor3f(0.0,0.5,0.0);
    draw_circle(40,250,l);
    draw_circle(80,250,l);
}

for(l=0;l<=25;l++)
{
    glColor3f(0.0,0.5,0.0);
    draw_circle(50,290,l);
    draw_circle(70,290,l);
}

for(l=0;l<=20;l++)
{
    glColor3f(0.0,0.5,0.0);
    draw_circle(60,315,l);
}

//railway station
glColor3f(0.647059,0.164706,0.164706);
glBegin(GL_POLYGON);
glVertex2f(400,250);

```

```
glVertex2f(400,450);
glVertex2f(950,450);
glVertex2f(950,250);
glEnd();
//roof
glColor3f(0.556863,0.419608,0.137255);
glBegin(GL_POLYGON);
glVertex2f(350,450);
glVertex2f(450,500);
glVertex2f(900,500);
glVertex2f(1000,450);
glEnd();
//window 1
glColor3f(0.196078,0.6,0.8);
glBegin(GL_POLYGON);
glVertex2f(450,300);
glVertex2f(450,375);
glVertex2f(550,375);
glVertex2f(550,300);
glEnd();
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(450,337.5);
glVertex2f(550,337.5);
glEnd();
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(500,375);
glVertex2f(500,300);
```

```
glEnd();  
  
//window 2  
glColor3f(0.196078,0.6,0.8);  
glBegin(GL_POLYGON);  
glVertex2f(800,300);  
glVertex2f(800,375);  
glVertex2f(900,375);  
glVertex2f(900,300);  
glEnd();  
  
glColor3f(0.0,0.0,0.0);  
glBegin(GL_LINES);  
glVertex2f(800,337.5);  
glVertex2f(900,337.5);  
glEnd();  
  
glColor3f(0.0,0.0,0.0);  
glBegin(GL_LINES);  
glVertex2f(850,375);  
glVertex2f(850,300);  
glEnd();  
  
//door  
glColor3f(0.329412,0.329412,0.329412);  
glBegin(GL_POLYGON);  
glVertex2f(625,250);  
glVertex2f(625,375);  
glVertex2f(725,375);  
glVertex2f(725,250);  
glEnd();  
  
//signal  
glColor3f(1.0,0.0,0.0);
```

```

glBegin(GL_POLYGON);
    glVertex2f(1060,160);
    glVertex2f(1060,350);
    glVertex2f(1070,350);
    glVertex2f(1070,160);
glEnd();
glColor3f(0.7,0.7,0.7);
glBegin(GL_POLYGON);
    glVertex2f(1040,350);
    glVertex2f(1040,500);
    glVertex2f(1090,500);
    glVertex2f(1090,350);
glEnd();
for(l=0;l<=20;l++)
{
    glColor3f(0.0,0.0,0.0);
    draw_circle(1065,475,l);
    glColor3f(1.0,1.0,0.0);
    draw_circle(1065,425,l);
    glColor3f(0.0,0.0,0.0);
    draw_circle(1065,375,l);
}
man();
if(man_show==1){
women();
}
if(train==1)
{
//train carrier 2

```

```

glColor3f(255.0,255.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(-150+i-xm,50);
glVertex2f(-150+i-xm,300);
glVertex2f(-550+i-xm,300);
glVertex2f(-550+i-xm,50);
glEnd();

//top
glColor3f(0.309804,0.184314,0.184314);
glBegin(GL_POLYGON);
glVertex2f(-140+i-xm,300);
glVertex2f(-140+i-xm,310);
glVertex2f(-560+i-xm,310);
glVertex2f(-560+i-xm,300);
glEnd();

// carrier 2 Wheels
for(l=0;l<50;l++)
{
    glColor3f(0.35,0.16,0.14);
    draw_circle(-225+i-xm,50,l);
    draw_circle(-475+i-xm,50,l);
}

// train carrier 1
glColor3f(0.258824,0.435294,0.258824);
glBegin(GL_POLYGON);
glVertex2f(300+i-xm,50);
glVertex2f(300+i-xm,300);
glVertex2f(-100+i-xm,300);
glVertex2f(-100+i-xm,50);

```

```

glEnd();

//top
glColor3f(0.309804,0.184314,0.184314);
glBegin(GL_POLYGON);
glVertex2f(310+i-xm,300);
glVertex2f(310+i-xm,310);
glVertex2f(-110+i-xm,310);
glVertex2f(-110+i-xm,300);
glEnd();

// Windows
glColor3f(1.0,1.0,1.0);
glBegin(GL_POLYGON);
glVertex2f(75+i-xm,175);
glVertex2f(75+i-xm,285);
glVertex2f(-85+i-xm,285);
glVertex2f(-85+i-xm,175);
glEnd();

glBegin(GL_POLYGON);
glVertex2f(285+i-xm,175);
glVertex2f(285+i-xm,285);
glVertex2f(125+i-xm,285);
glVertex2f(125+i-xm,175);
glEnd();

// carrier 1 Wheels
for(l=0;l<50;l++)
{
    glColor3f(0.35,0.16,0.14);
    draw_circle(-25+i-xm,50,l);
    draw_circle(225+i-xm,50,l);
}

```

```

    }

    //train base
    glColor3f(0.196078,0.6,0.8);
    glBegin(GL_POLYGON);
    glVertex2f(350+i-xm,50);
    glVertex2f(350+i-xm,125);
    glVertex2f(755+i-xm,125);
    glVertex2f(820+i-xm,50);
    glEnd();

    //train control chamber
    glColor3f(1.0,0.25,0.0);
    glBegin(GL_POLYGON);
    glVertex2f(360+i-xm,125);
    glVertex2f(360+i-xm,325);
    glVertex2f(560+i-xm,325);
    glVertex2f(560+i-xm,125);
    glEnd();

    //train top
    glColor3f(0.309804,0.184314,0.184314);
    glBegin(GL_POLYGON);
    glVertex2f(350+i-xm,325);
    glVertex2f(350+i-xm,350);
    glVertex2f(570+i-xm,350);
    glVertex2f(570+i-xm,325);
    glEnd();

    //train engine
    glColor3f(1.0,0.5,0.0);
    glBegin(GL_POLYGON);
    glVertex2f(560+i-xm,125);

```



```

glVertex2f(560+i-xm,225);
glVertex2f(755+i-xm,225);
glVertex2f(755+i-xm,125);
glEnd();
glColor3f(0.0,0.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(580+i-xm,125);
glVertex2f(580+i-xm,225);
glVertex2f(590+i-xm,225);
glVertex2f(590+i-xm,125);
glEnd();
glColor3f(0.0,0.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(600+i-xm,125);
glVertex2f(600+i-xm,225);
glVertex2f(610+i-xm,225);
glVertex2f(610+i-xm,125);
glEnd();
//train smoke
glColor3f(0.196078,0.6,0.9);
glBegin(GL_POLYGON);
glVertex2f(650+i-xm,225);
glVertex2f(650+i-xm,275);
glVertex2f(700+i-xm,275);
glVertex2f(700+i-xm,225);
glEnd();
glColor3f(0.309804,0.184314,0.184314);
glBegin(GL_POLYGON);
glVertex2f(640+i-xm,275);

```

```

glVertex2f(640+i-xm,300);
glVertex2f(710+i-xm,300);
glVertex2f(710+i-xm,275);
glEnd();
//train head-light
glColor3f(1.0,1.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(755+i-xm,225);
glVertex2f(765+i-xm,225);
glVertex2f(765+i-xm,185);
glVertex2f(755+i-xm,185);
glEnd();
glColor3f(0.0,0.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(755+i-xm,225);
glVertex2f(785+i-xm,225);
glVertex2f(755+i-xm,205);
glEnd();
//Railway Station Board
glColor3f(0.435294,0.258824,0.258824);
glBegin(GL_POLYGON);
glVertex2f(560,510);
glVertex2f(560,540);
glVertex2f(580,550);
glVertex2f(780,550);
glVertex2f(800,540);
glVertex2f(800,510);
glEnd();
glColor3f(1.0,1.0,1.0);

```

```
    glRasterPos2f(570,520);  
  
    declare("RAILWAY STATION");  
  
    glFlush();}
```

5: void train_comp() represents the interior of the train having windows, seating arrangements and the passengers

```
void train_comp()  
{ int i;  
  //base  
  glColor3f(0.6, 0.6, 0.6);  
  glBegin(GL_POLYGON);  
  glVertex2f(0,0);  
  glVertex2f(400,360);  
  glVertex2f(700,360);  
  glVertex2f(1100,0);  
  glEnd();  
  //borderbase  
  glColor3f(0.0,0.0,0.0);  
  glBegin(GL_POLYGON);  
  glVertex2f(400,350);  
  glVertex2f(400,360);  
  glVertex2f(700,360);  
  glVertex2f(700,350);  
  glEnd();  
  //front  
  glColor3f(0.6, 0.6, 0.6);  
  glBegin(GL_POLYGON);  
  glVertex2f(400,360);  
  glVertex2f(400,1250);  
  glVertex2f(700,360);
```

```
glVertex2f(700,1250);
glEnd();
//borderfront1
glColor3f(0.0,0.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(390,350);
glVertex2f(390,1250);
glVertex2f(400,350);
glVertex2f(400,1250);
glEnd();
//borderfront2
glColor3f(0.0,0.0,0.0);
glBegin(GL_POLYGON);
glVertex2f(700,350);
glVertex2f(700,1250);
glVertex2f(710,1250);
glVertex2f(710,350);
glEnd();
//left
glColor3f(0.0, 0.4, 1.0);
glBegin(GL_POLYGON);
glVertex2f(0,0);
glVertex2f(0,1250);
glVertex2f(390,1250);
glVertex2f(390,350);
glEnd();
//right
glColor3f(0.0, 0.4, 1.0);
glBegin(GL_POLYGON);
```

```
glVertex2f(710,350);
glVertex2f(710,1250);
glVertex2f(1100,1250);
glVertex2f(1100,0);
glEnd();
//Left window 1
//blue
glColor3f(0.0, 255.0, 1.0);
glBegin(GL_POLYGON);
glVertex2f(20,275);
glVertex2f(20,500);
glVertex2f(155,575);
glVertex2f(155,380);
glEnd();
//white
glColor3f(1.0,1.6,1.8);
glBegin(GL_POLYGON);
glVertex2f(25,300);
glVertex2f(25,475);
glVertex2f(150,550);
glVertex2f(150,395);
glEnd();
//tree 1
glColor3f(0.9,0.2,0.0);
glBegin(GL_POLYGON);
glVertex2f(70,335);
glVertex2f(70,400);
glVertex2f(90,400);
glVertex2f(90,350);
```

```

glEnd();
for(l=0;l<=30;l++)
{
    glColor3f(0.0,0.5,0.0);
    draw_circle(60,400,l);
    draw_circle(100,400,l);
}
for(l=0;l<=25;l++)
{
    glColor3f(0.0,0.5,0.0);
    draw_circle(70,440,l);
    draw_circle(90,440,l);
}
for(l=0;l<=20;l++)
{
    glColor3f(0.0,0.5,0.0);
    draw_circle(80,470,l);
}

//Left grills 1
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(25,350);
glVertex2f(150,435);
glEnd();
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(25,390);
glVertex2f(150,475);
glEnd();

```

```
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(25,430);
glVertex2f(150,515);
glEnd();

//Right window 1
//blue
glColor3f(0.0, 255.0, 1.0);
glBegin(GL_POLYGON);
glVertex2f(930,380);
glVertex2f(930,570);
glVertex2f(1070,500);
glVertex2f(1070,275);
glEnd();

//white
glColor3f(1.0,1.6,1.8);
glBegin(GL_POLYGON);
glVertex2f(940,390);
glVertex2f(940,545);
glVertex2f(1060,480);
glVertex2f(1060,305);
glEnd();

//tree
glColor3f(0.9,0.2,0.0);
glBegin(GL_POLYGON);
glVertex2f(990,355);
glVertex2f(990,455);
glVertex2f(1010,455);
glVertex2f(1010,340);
```

```

glEnd();
for(l=0;l<=30;l++)
{
    glColor3f(0.0,0.5,0.0);
    draw_circle(980,420,l);
    draw_circle(1020,410,l);
}
for(l=0;l<=25;l++)
{
    glColor3f(0.0,0.5,0.0);
    draw_circle(990,450,l);
    draw_circle(1010,450,l);
}
for(l=0;l<=20;l++)
{
    glColor3f(0.0,0.5,0.0);
    draw_circle(1000,480,l);
}
//right grills 1
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(940,500);
glVertex2f(1060,435);
glEnd();
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(940,465);
glVertex2f(1060,400);
glEnd();

```



```

glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(940,430);
glVertex2f(1060,360);
glEnd();

    //seats 7 left3
    glPushMatrix();
    glTranslatef(-10,-100,0);
    glColor3f(0.647059,0.164706,0.164706);
    glBegin(GL_POLYGON);
        glVertex2f(300,350);
        glVertex2f(310,410);
        glVertex2f(470,410);
        glVertex2f(480,350);
    glEnd();
    glColor3f(0.647059,0.164706,0.164706);
    glBegin(GL_POLYGON);
        glVertex2f(300,350);
        glVertex2f(280,330);
        glVertex2f(460,330);
        glVertex2f(480,350);
    glEnd();

    //seat legs
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex2f(480,350);
        glVertex2f(480,300);
    glEnd();
    glColor3f(0,0,0);

```

```

        glBegin(GL_LINES);
            glVertex2f(460,330);
            glVertex2f(460,300);
        glEnd();
        glColor3f(0,0,0);
        glBegin(GL_LINES);
            glVertex2f(280,330);
            glVertex2f(280,280);
        glEnd();
        glColor3f(0,0,0);
        glBegin(GL_LINES);
            glVertex2f(300,330);
            glVertex2f(300,300);
        glEnd();
        glPopMatrix();
man();
women();
//seats 8 right4
        glPushMatrix();
        glTranslatef(280,0,0);
        glColor3f(0.647059,0.164706,0.164706);
        glBegin(GL_POLYGON);
            glVertex2f(300,350);
            glVertex2f(310,410);
            glVertex2f(470,410);
            glVertex2f(480,350);
        glEnd();
        glColor3f(0.647059,0.164706,0.164706);
        glBegin(GL_POLYGON);

```

```

        glVertex2f(300,350);
        glVertex2f(280,330);
        glVertex2f(460,330);
        glVertex2f(480,350);
    glEnd();
    //seat legs
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex2f(480,350);
        glVertex2f(480,300);
    glEnd();
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex2f(460,330);
        glVertex2f(460,300);
    glEnd();
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex2f(280,330);
        glVertex2f(280,280);
    glEnd();
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex2f(300,330);
        glVertex2f(300,300);
    glEnd();
    glPopMatrix();
    //seats 1 right3
    glPushMatrix();

```

```

glTranslatef(330,-100,0);
glColor3f(0.647059,0.164706,0.164706);
glBegin(GL_POLYGON);
    glVertex2f(300,350);
    glVertex2f(310,410);
    glVertex2f(470,410);
    glVertex2f(480,350);
glEnd();
glColor3f(0.647059,0.164706,0.164706);
glBegin(GL_POLYGON);
    glVertex2f(300,350);
    glVertex2f(280,330);
    glVertex2f(460,330);
    glVertex2f(480,350);
glEnd();
//seat legs
glColor3f(0,0,0);
glBegin(GL_LINES);
    glVertex2f(480,350);
    glVertex2f(480,300);
glEnd();
glColor3f(0,0,0);
glBegin(GL_LINES);
    glVertex2f(460,330);
    glVertex2f(460,310);
glEnd();
glColor3f(0,0,0);
glBegin(GL_LINES);
    glVertex2f(280,330);

```

```

        glVertex2f(280,280);
    glEnd();
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex2f(300,330);
        glVertex2f(300,300);
    glEnd();
    glPopMatrix();
//seats 2 right2
glPushMatrix();
    glTranslatef(380,-200,0);
    glColor3f(0.647059,0.164706,0.164706);
    glBegin(GL_POLYGON);
        glVertex2f(300,350);
        glVertex2f(310,410);
        glVertex2f(470,410);
        glVertex2f(480,350);
    glEnd();
glColor3f(0.647059,0.164706,0.164706);
    glBegin(GL_POLYGON);
        glVertex2f(300,350);
        glVertex2f(280,330);
        glVertex2f(460,330);
        glVertex2f(480,350);
    glEnd();
//seat legs
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex2f(480,350);

```

```

        glVertex2f(480,300);
    glEnd();
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex2f(460,330);
        glVertex2f(460,310);
    glEnd();
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex2f(280,330);
        glVertex2f(280,280);
    glEnd();
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex2f(300,330);
        glVertex2f(300,300);
    glEnd();
    glPopMatrix();
    glFlush();
}

```

6: void hill() is used to display the hills along with the sky in the morning along with a house.

```

void hill()
{
    int i;
    //sky
    glColor3f(0.0,0.9,0.9);
    glBegin(GL_POLYGON);
    glColor3f(1,1,1 );

```

```

glVertex2f(0,380);
glColor3f(0.3,0.2,1 );
glVertex2f(0,700);
glVertex2f(1100,700);
glColor3f(1,1,1 );
glVertex2f(1100,380);
glEnd();
//sun
    for(l=0;l<=35;l++)
    {
        glColor3f(1.0,0.9,0.0);
        draw_circle(300,625,l);
    }
//hills
glColor3f(0.6,0.3,0.1);
glBegin(GL_TRIANGLES);
glVertex2f(0,420);
glVertex2f(150,640);
glVertex2f(300,420);
glEnd();
glColor3f(0.6,0.3,0.1);
glBegin(GL_TRIANGLES);
glVertex2f(280,420);
glVertex2f(430,640);
//grass
glColor3f(0.4,0.8,0.0);
glBegin(GL_POLYGON);
glVertex2f(0,160);
glVertex2f(0,420);

```

```

glVertex2f(1100,420);
glVertex2f(1100,160);
glEnd();
//tree 1
glColor3f(0.5,0.1,0);
glBegin(GL_POLYGON);
glVertex2f(50,245);
glVertex2f(50,355);
glVertex2f(70,355);
glVertex2f(70,245);
glEnd();
    for(l=0;l<=30;l++)
    {
        glColor3f(0.0,0.2,0.0);
        draw_circle(40,350,l);
        draw_circle(80,350,l);
    }
    for(l=0;l<=25;l++)
    {
        glColor3f(0.0,0.2,0.0);
        draw_circle(50,390,l);
        draw_circle(70,390,l);
    }
    for(l=0;l<=20;l++)
    {
        glColor3f(0.0,0.2,0.0);
        draw_circle(60,415,l);
    }
//house

```



```
//roof
glColor3f(0.6,0.2,0.12);
glBegin(GL_POLYGON);
glVertex2f(500,320);
glVertex2f(650,400);
glVertex2f(750,400);
glVertex2f(900,320);
glEnd();

//room
glColor3f(0.8,0.7,0.5);
glBegin(GL_POLYGON);
glVertex2f(550,200);
glVertex2f(550,320);
glVertex2f(850,320);
glVertex2f(850,200);
glEnd();

//door
glColor3f(0.196078,0.6,0.8);
glBegin(GL_POLYGON);
glVertex2f(670,200);
glVertex2f(670,300);
glVertex2f(730,300);
glVertex2f(730,200);
glEnd();

glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(670,200);
glVertex2f(670,300);
glEnd();
```

```
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(670,300);
glVertex2f(730,300);
glEnd();
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(730,300);
glVertex2f(730,200);
glEnd();
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(700,200);
glVertex2f(700,300);
glEnd();
//window1
glColor3f(0.196078,0.6,0.8);
glBegin(GL_POLYGON);
glVertex2f(570,250);
glVertex2f(570,300);
glVertex2f(650,300);
glVertex2f(650,250);
glEnd();
glColor3f(0.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2f(570,275);
glVertex2f(650,275);
glEnd();
glColor3f(0.0,0.0,0.0);
```

```

    glBegin(GL_LINES);
    glVertex2f(610,250);
    glVertex2f(610,300);
    glEnd();
train();
glFlush();
}

```

7: void river() displays the view of river along with trees, bushes, boats and sunset.

```

void river()
{
    int l;
    //sky
    glBegin(GL_POLYGON);
    glColor3f(1,0.3,0.2);
    glVertex2f(0,380);
    glColor3f(0.2,0.4,1);
    glVertex2f(0,700);
    glVertex2f(1100,700);
    glColor3f(1,0.3,0.2);
    glVertex2f(1100,380);
    glEnd();
    //sun
    for(l=0;l<=35;l++)
    {
        glColor3f(1,0.5,0);
        draw_circle(580,380,l);
    }
    //grass

```

```
glBegin(GL_POLYGON);
glColor3f(0.1,0.3,0);
glVertex2f(0,160);
glVertex2f(0,380);
glVertex2f(1100,380);
glVertex2f(1100,160);
glEnd();

//river
glBegin(GL_POLYGON);
glColor3f(0.3,0,0.8);
glColor3f(0.3,0,1);
glColor3f(0.3,0.1,0.9);
glVertex2f(0,160);
glColor3f(1,0.6,0.4);
glVertex2f(400,380);
glVertex2f(800,380);
glColor3f(0.3,0,1);
glColor3f(0.3,0.1,0.9);
glVertex2f(1100,160);
glEnd();

//boat
glBegin(GL_POLYGON);
glColor3f(0,0,0);
glVertex2f(460,240);
glVertex2f(410,270);
glVertex2f(580,270);
glVertex2f(520,240);
glEnd();
glBegin(GL_POLYGON);
```

```

glColor3f(0,0,0);
glVertex2f(470,270);
glVertex2f(490,300);
glVertex2f(510,270);
glEnd();
glBegin(GL_POLYGON);
glColor3f(0,0,0);
glVertex2f(720,330);
glVertex2f(890,330);
glVertex2f(850,305);
glVertex2f(770,305);
glEnd();
glBegin(GL_POLYGON);
glColor3f(0,0,0);
glVertex2f(780,330);
glVertex2f(805,350);
glVertex2f(830,330);
glEnd();
//tree 1
glColor3f(0.5,0.1,0);
glBegin(GL_POLYGON);
glVertex2f(50,245);
glVertex2f(50,355);
glVertex2f(70,355);
glVertex2f(70,245);
glEnd();
    for(l=0;l<=30;l++)
    {
        glColor3f(0.0,0.2,0.0);

```

```
        draw_circle(40,350,l);
        draw_circle(80,350,l);
    }
    for(l=0;l<=25;l++)
    {
        glColor3f(0.0,0.2,0.0);
        draw_circle(50,390,l);
        draw_circle(70,390,l);
    }
    for(l=0;l<=20;l++)
    {
        glColor3f(0.0,0.2,0.0);
        draw_circle(60,415,l);
    }
    train();
    glflush();
}
```

CONCLUSION

In this project we have demonstrated the simulation of a train. There is a railway station surrounded by trees and bushes and passengers waiting for the train. The train is simulated with the help of signal. We have shown the passengers get into the train and it passes by several hills and rivers. Also the interior of the train is displayed along with the passengers.

OpenGL is a low-level graphics library specification. It makes available to the programmer a small set of geometric primitives - points, lines, polygons, images, and bitmaps. OpenGL provides a set of commands that allow the specification of geometric objects in two or three dimensions, using the provided primitives, together with commands that control how these objects are rendered (drawn).

OpenGL supports enormous flexibility in design and the use of OpenGL graphics programs. The Presence of many built in classes methods take care of much functionality and reduce the job of coding as well as makes the implementation simpler. We have implemented the project making it user-friendly and error free as possible.

REFERENCES

- <https://www.opengl.org/resources/>
- <https://learnopengl.com/Advanced-OpenGL/Advanced-Data>
- <http://www.opengl-tutorial.org/beginners-tutorials/tutorial-1-opening-a-window/>
- <https://open.gl/transformations>
- <http://www.cs.uccs.edu/~ssemwal/indexGLTutorial.html>

APPENDIX

Figure 1: Introductory page - Name USN

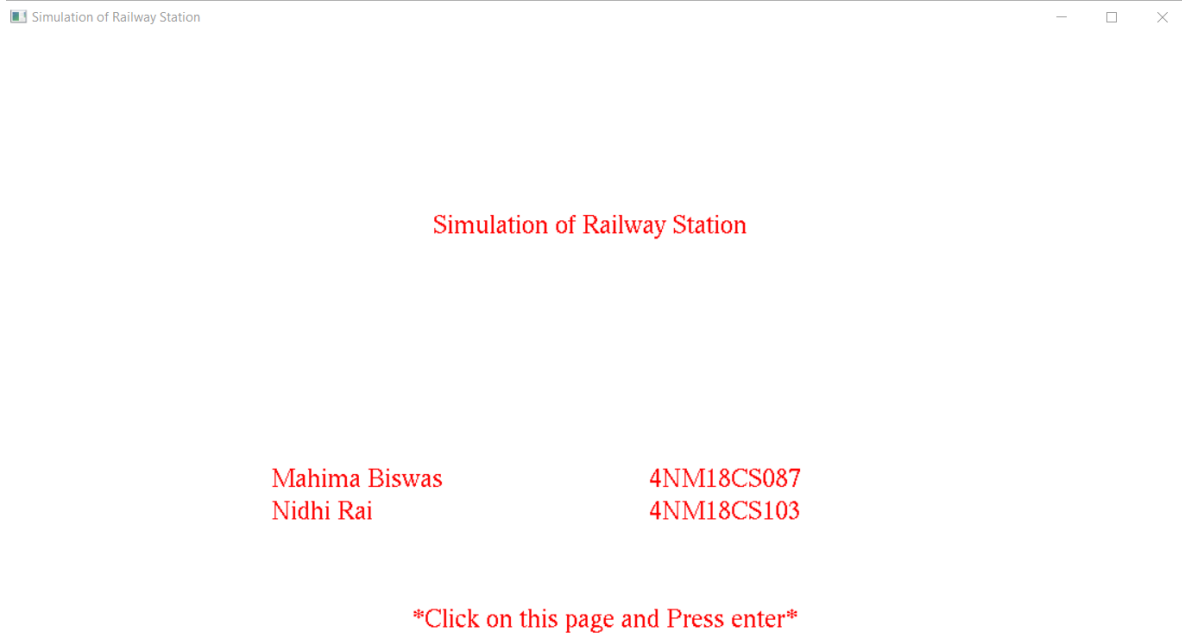


Figure 2 : View of railway station with Aeroplane in day mode



Figure 3: Night Mode along with moon,stars and comet

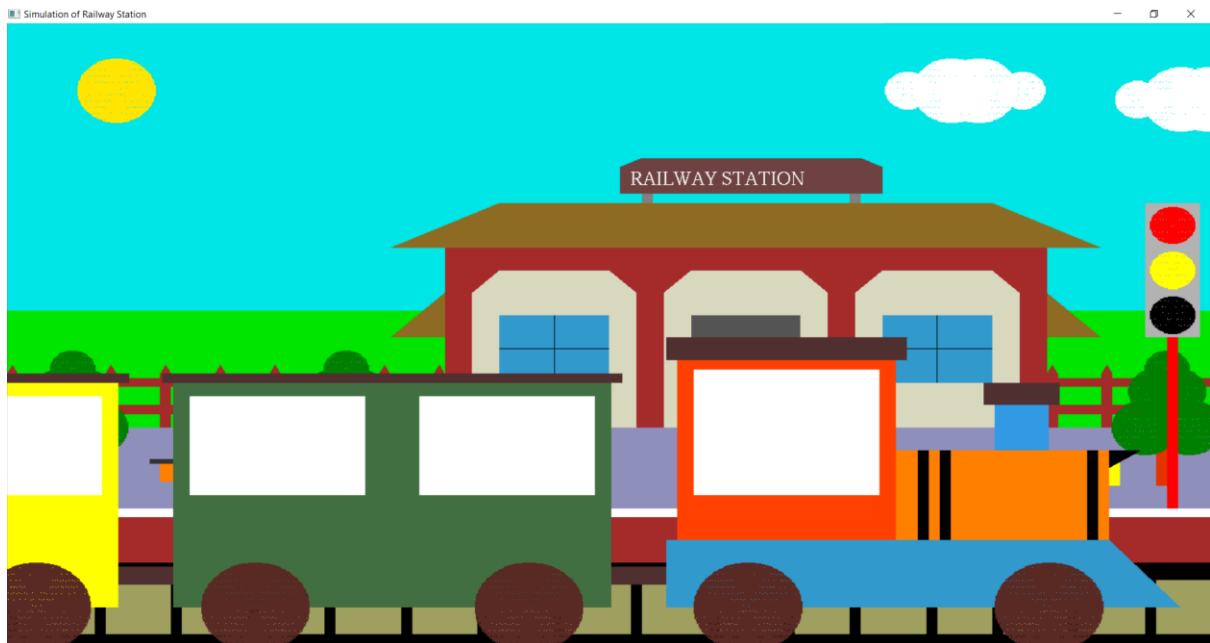


Figure 4: Train stops when the signal turns red and passengers get into train



Figure 5: View of railway station after departure of train with passenger

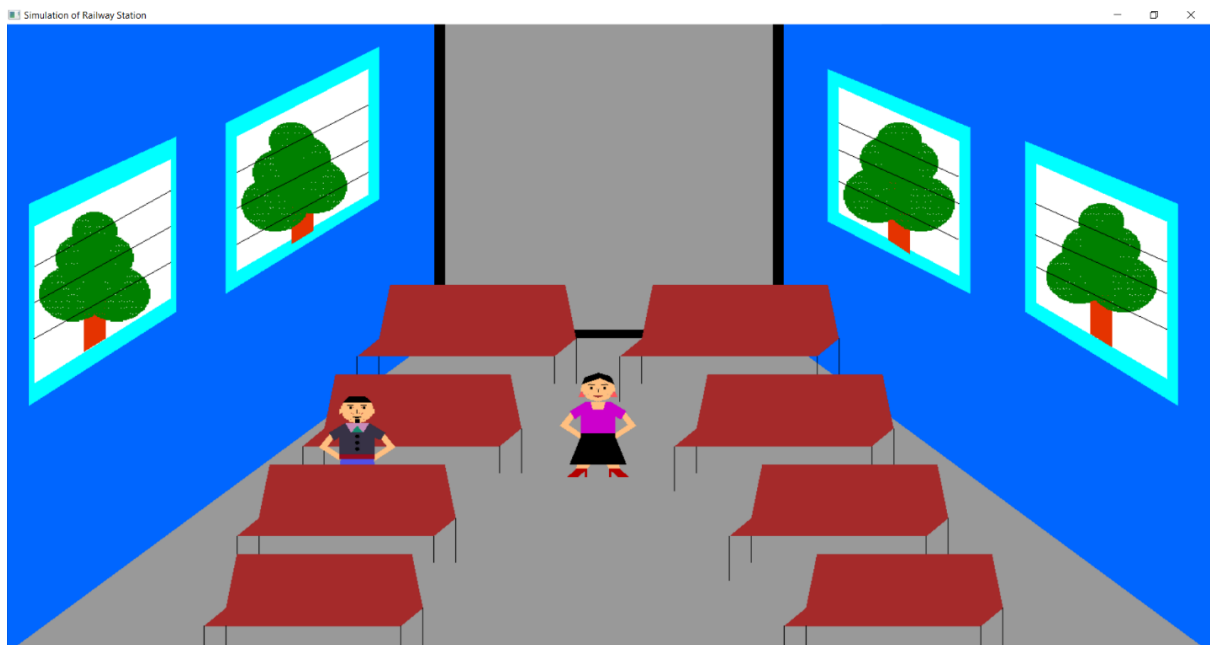


Figure 6: Train compartment along with passengers

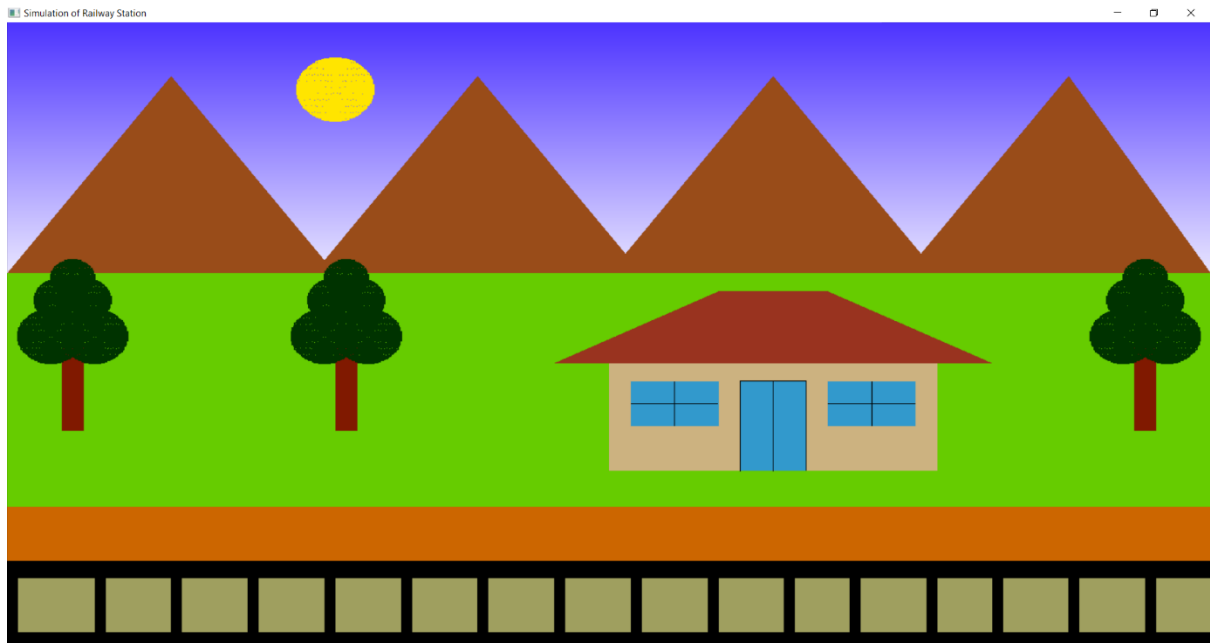


Figure 7: Early morning view of village having trees,hills and house

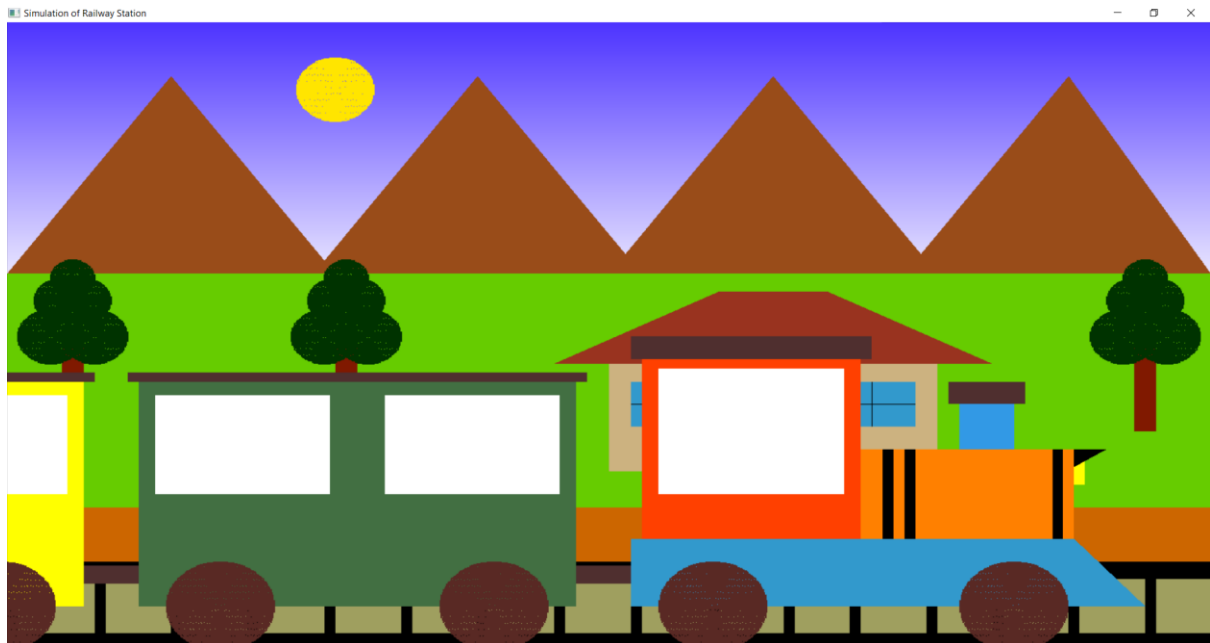


Figure 8: Train passing by the village

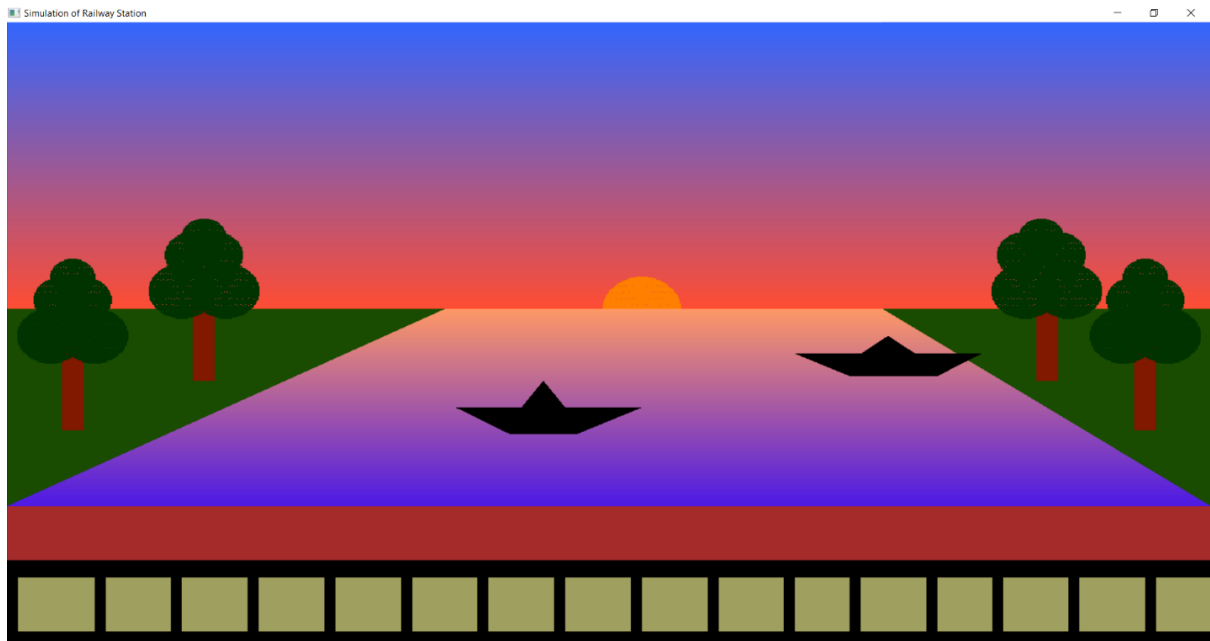


Figure 9: View of river during sunset

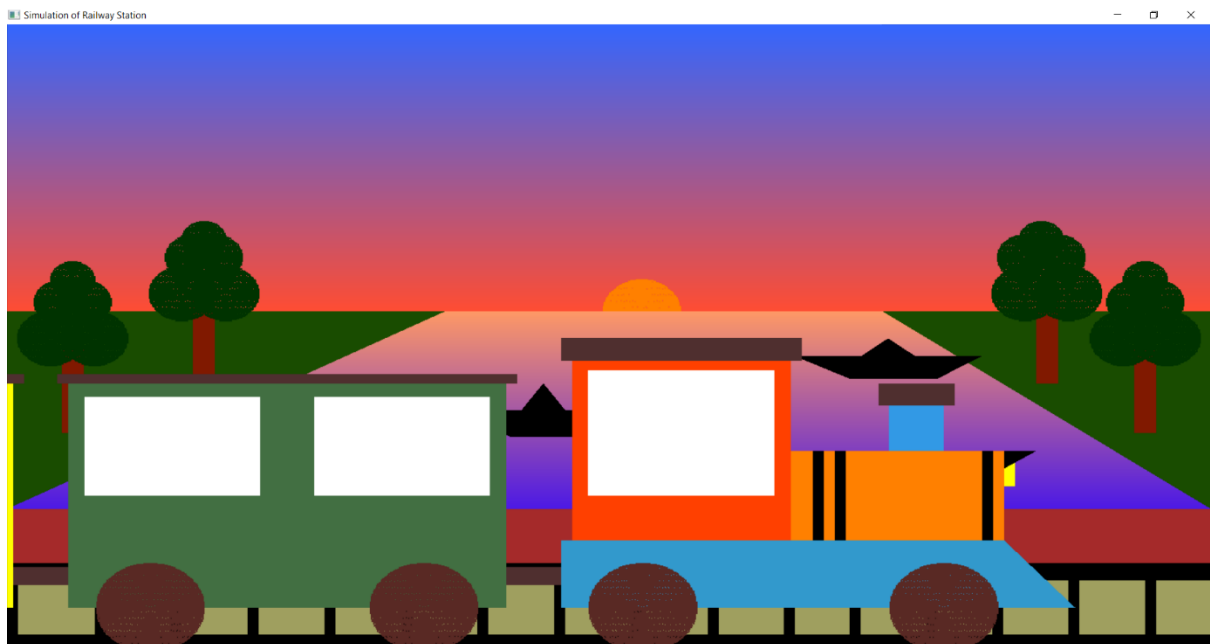


Figure 10: Train crossing the river