

COMPUTER GRAPHICS (UCS505)
Project Report On Trains Stoppage

Submitted By:

Kush Gupta	101803454
Priyanshi Tiwari	101803417
Neetika	101853024

COE -19

B.E. Third Year - COE

Submitted To:

Dr. Shatrughan Modi



THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department
Thapar Institute Of Engineering and Technology
Patiala-147001
June 2021

Table of Contents

S. No	Title	Page number
1.	Introduction of project	2
2.	Computer Graphics concepts used	3
3.	Functions used	4
4.	Code	5
5.	Snapshots	14

INTRODUCTION:

Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D or 3D pattern-recognition abilities allow us to perceive and process pictorial data rapidly. Computers have become a powerful medium for the rapid and economical production of pictures. There is virtually no area in which graphical displays cannot be used to some advantage. Graphics provide a natural means of communicating with the computer that they have become widespread. Interactive graphics is the most important means of producing pictures since the invention of photography and television. We can make pictures of not only the real world objects but also of abstract objects such as mathematical surfaces on 4D and of data that have no inherent geometry. A computer graphics system is a computer system with all the components of a general purpose computer system. There are five major elements in a system: input devices, processor, memory, frame buffer, output devices.

The idea behind this project is to display the working of the train using computer graphics. This project demonstrates the scene in which one train is moving from left to right and another one is moving from right to left. This project consists of three segments: the sky, the tracks, and the grass or ground. Both trains consist of an engine and three bogies. We made separate stations for both trains.

Both trains will move at the consistent speed until it reaches specified stations. After pressing the particular keys both trains will start to move till it reaches the end of the window. When both trains reach the end of the window the name of both stations will be changed.

CONCEPT:

Computer graphics is one of the most exciting and rapidly growing computer fields. It is also an extremely effective medium for communication between man and computer; a human being can understand the information content of a displayed diagram or perspective view much faster than he can understand a table of numbers or text containing the same information.

OpenGL (open graphics library) is a standard specification defining a cross language across platform API for writing applications that produce 2D and 3D computer graphics.

OpenGL is an application program interface (API) offering various functions to implement primitives, models and images. This offers functions to create and manipulate render lighting, coloring, viewing the models. OpenGL offers different coordinate systems and frames. OpenGL offers translation, rotation and scaling of objects.

The main aim of this mini project is to illustrate the concepts and usage of pre-built functions in OpenGL. It simulates two trains where two tracks are not connected but trains work simultaneously with constant speed.

The best method for building the objects is by placing the drawing code for each object in a separate function. Each object is defined using a coordinate system that makes modelling convenient.

After we finish the drawing our next aim would be to give motion to the train. As we are going to draw a simple train so our track will be straight. Train engines and bogies are made of many boxes or rectangles while wheels are circular. Track is made by using 2 lines and contains small rectangles continuously. For developing *the train - it's bogies and engine* we have defined a simple function. In this function, we tried to code in such a way that both trains work simultaneously.

We have represented the Trains stoppage by using two flags . The two flags are initially named as 'K' and 'G' but when trains have crossed over then the name on flag is changed to 'S' and 'D'. Both trains stop at a particular location destined to them. We have used the input device keyboard to interact with the program.

FUNCTIONS USED:

- **myint()**- This function calls tracks and train functions in it and also includes inbuilt glut functions which are GenList () and glNewList() will create a new list and store all the other functions into it. It is the most important function of the code. It will contain all the figures. basic1 and basic2 will contain a matrix of figures.
- **drawstring()**-This function included in the display function which includes the glutBitmapCharacter function which is used to display text in the window screen. If f1 and f2 is zero it will draw K and G on the flags, otherwise it will display S and D on them.
- **draw2()**-This function will show the initial window which contains the details of the project and name,roll no. of team mates who have made the project.This function includes the Key 'S' which gives the initial window to our project.
- **TRAINS1() & TRAINS2()**- It will display the train on the window. Initially it will create an engine of the train which is drawn similar to a rectangle. We have drawn 3 bogies of the train. For drawing bogies a loop is run from 1 to 3 and bogies are also created similar to rectangles.
- **TRACKS()& TRACK1()**- Both tracks initially contain two parallel lines and these lines are having length same as the window size and then after a gap of 10, vertical tiny rectangles are created giving a shape of track.
- **idle()**- This function gives the time laps in the code when the function is doing nothing.
- **keb()**- We are taking inputs from users and setting flags accordingly.The movement of the train is decided by this function as it contains flag, flag2 and flag3 variables which decide the stoppage and starting of the train at various points in the code.
- **display()**- Initially it will show the background i.e. sky and ground.This function is for the movement of the train. It sets the speed of trains by changing the value of x. X coordinates of both the trains are set manually. Both trains stop at the 0 location set by us. Flags are static so we created them using polygon. basic1 and train1 are called from myinit() function.

CODE:

```
#include<stdio.h>
#include<stdlib.h>
#include<GL/glut.h>
#include <math.h>

int c = 0, d = 500, g = 380, h = 970, i = 0, p = 0, q = 300;
GLfloat x = 0.0, y = 0.0;
GLfloat flag = 0, flag2 = 0, flag3 = 0;
int train1, basic1, train2, basic2;
float f1 = 0, f2 = 0;

void draw2()
{
    glColor3f(1.0, 0.0, 0.0);
    //      glRasterPos3f(450,950, 0.0);
    char c[100] = "Computer Graphics(UCS505) Mini Project";
    char c1[100] = "TITLE : TRAINS STOPPAGE";

    char c3[100] = "SUBMITTED TO : Dr. Shatrughan Modi";
    char c4[100] = "SUBMITTED BY :";
    char c9[100] = "Priyanshi Tiwari    Roll no. : 101803417";
    char c5[100] = "Kush Gupta        Roll no. : 101803454";
    char c6[100] = "Neetika           Roll no. : 101853024";
    char c7[100] = "PRESS 'S' TO START";
    int i;
    glColor3f(rand() % 3, rand() % 3, rand() % 3);
    glRasterPos3f(400, 900, 0.0);
    for (i = 0; c[i] != '\0'; i++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, c[i]); // Updates the
position
    glColor3f(rand() % 3, rand() % 3, rand() % 3);
    glRasterPos3f(400, 850, 0.0);
    for (i = 0; c1[i] != '\0'; i++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, c1[i]);
    glColor3f(0.0, 1.0, 0.0);
    glRasterPos3f(300, 670, 0.0);

    for (i = 0; c3[i] != '\0'; i++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, c3[i]);
    glColor3f(1.0, 1.0, 0.5);
    glRasterPos3f(300, 560, 0.0);

    for (i = 0; c4[i] != '\0'; i++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, c4[i]);
    glColor3f(1.0, 1.0, 0.5);
    glRasterPos3f(300, 500, 0.0);
```

```

    for (i = 0; c9[i] != '\0'; i++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, c9[i]);
    glColor3f(1.0, 1.0, 0.5);
    glRasterPos3f(300, 440, 0.0);

    for (i = 0; c5[i] != '\0'; i++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, c5[i]);
    glColor3f(1.0, 1.0, 0.5);
    glRasterPos3f(300, 380, 0.0);

    for (i = 0; c6[i] != '\0'; i++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, c6[i]);
    glColor3f(rand() % 3, rand() % 3, rand() % 3);
    glRasterPos3f(400, 200, 0.0);

    for (i = 0; c7[i] != '\0'; i++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, c7[i]);
}
void drawString()
{
    glColor3f(0.0, 0.0, 0.0);
    glRasterPos3f(306, 615, 0.0);
    if (f1 == 0)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18, 'K');
    else
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18, 'S');
    glRasterPos3f(806, 415, 0.0);
    if (f2 == 0)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18, 'G');
    else
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18, 'D');

}

void sky()
{
    glBegin(GL_POLYGON);
    glColor3f(0, 0.8, 1);//sky
    glVertex2f(0, 730);
    glVertex2f(999, 730);
    glVertex2f(999, 999);
    glVertex2f(0, 999);
    glEnd();
}

```

```
}
```

```
void TRAINS1(int x1, int y1, int c, int d)
```

```
{
```

```
    int i = 0;
```

```
    glBegin(GL_QUADS);
```

```
    glColor3f(1, 0.0, 0.0);
```

```
    //ENGINE
```

```
    glVertex2f(x1, y1);
```

```
    //length of engine=60;height of engine=30;
```

```
    glColor3f(1, 0.0, 0.0);
```

```
    glVertex2f(x1 + 60, y1);
```

```
    glColor3f(0.0, 0.0, 1.0);
```

```
    glVertex2f(x1 + 60, y1 - 30);
```

```
    glColor3f(0, 0.0, 0.0);
```

```
    glVertex2f(x1, y1 - 30);
```

```
    glEnd();
```

```
    while (i < 4)
```

```
    {
```

```
        glBegin(GL_QUADS);
```

```
        //BOGIES
```

```
        glColor3f(0.0, 0.0, 0.0);
```

```
        //For right train a=795,b=510
```

```
        glVertex2f(c, d);
```

```
        glColor3f(0.0, 0.0, 0.0);
```

```
        glVertex2f(c + 60, d);
```

```
        glColor3f(0.0, 0.0, 1.0);
```

```
        glVertex2f(c + 60, d - 20);
```

```
        glColor3f(0.0, 0.0, 1.0);
```

```
        glVertex2f(c, d - 20);
```

```
        glEnd();
```

```
        c += 65;
```

```
        i++;
```

```
    }
```

```
}
```

```
void TRAINS2(int x2, int y2, int s, int t)
```

```
{
```

```
    int r = 0;
```

```
    while (r < 4)
```

```
    {
```

```
        glBegin(GL_QUADS);
```

```
        //BOGIES
```

```
        glColor3f(0.0, 0.0, 1.0);
```

```
        //For right train a=795,b=510
```

```
        glVertex2f(s, t);
```

```
        glColor3f(0.0, 0.0, 1.0);
```

```
        glVertex2f(s - 60, t);
```



```

        glColor3f(0.0, 0.0, 0.0);
        glVertex2f(s - 60, t + 20);
        glColor3f(0.0, 0.0, 0.0);
        glVertex2f(s, t + 20);
        glEnd();
        s -= 65;
        r++;
    }

```

```

glBegin(GL_QUADS);
glColor3f(0, 0.0, 0.0);
glVertex2f(x2, y2);
glColor3f(0, 0.0, 1.0);
glVertex2f(x2 - 60, y2);
glColor3f(1.0, 0.0, 0.0);
glVertex2f(x2 - 60, y2 + 30);
glColor3f(1, 0.0, 0.0);
glVertex2f(x2, y2 + 30);
glEnd();
}

```

//ENGINE
//length of engine=60;height of engine=30;

```

void TRACKS()
{

```

```

    sky();
    /* Track */
    glBegin(GL_LINES);
    glColor3f(0.0, 0.0, 0.0);
    glVertex2f(0, 500);
    glVertex2f(1000, 500);
    glVertex2f(0, 499);
    glVertex2f(1000, 499);

    glVertex2f(0, 485);
    glVertex2f(1000, 485);
    glVertex2f(0, 486);
    glVertex2f(1000, 486);
    while (c != 1000)
    {
        glVertex2f(c, d);
        glVertex2f(c, d - 15);
        c += 10;
    }
    glEnd();

```

```

}

void TRACKS1()
{
    glBegin(GL_LINES);
    glColor3f(0.0, 0.0, 0.0);
    glVertex2f(0, 300);
    glVertex2f(1000, 300);
    glVertex2f(0, 299);
    glVertex2f(1000, 299);

    glVertex2f(0, 285);
    glVertex2f(1000, 285);
    glVertex2f(0, 286);
    glVertex2f(1000, 286);
    while (p != 1000)
    {
        glVertex2f(p, q);
        glVertex2f(p, q - 15);
        p += 10;
    }
    glEnd();
}

```

```

void myinit()
{
    //background color
    gluOrtho2D(0, 999, 0, 999);
    glPointSize(2.0);
    glLineWidth(2.0);

    basic1 = glGenLists(1);
    glNewList(basic1, GL_COMPILE);
    TRACKS();
    glEndList();

    train1 = glGenLists(1);
    glNewList(train1, GL_COMPILE);
    TRAINS1(730, 520, 795, 510);
    glEndList();

    basic2 = glGenLists(1);
    glNewList(basic2, GL_COMPILE);
    TRACKS1();
    glEndList();
}

```

```

    train2 = glGenLists(1);
    glNewList(train2, GL_COMPILE);
    TRAINS2(270, 288, 205, 288);
    glEndList();

}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glClearColor(0.0, 0.7, 0.0, 0.0);

    glBegin(GL_LINES);
    glColor3f(0.2, 0.2, 0.2);
    glVertex2f(300, 500);
    glVertex2f(300, 650);
    glVertex2f(301, 500);
    glVertex2f(301, 650);
    glEnd();
    glColor3f(1.0, 1.0, 0);
    glBegin(GL_POLYGON);
    glVertex2f(302, 650);
    glVertex2f(302, 600);
    glVertex2f(330, 625);

    glEnd();

    glBegin(GL_LINES);
    glColor3f(0.2, 0.2, 0.2);
    glVertex2f(800, 300);
    glVertex2f(800, 450);
    glVertex2f(801, 300);
    glVertex2f(801, 450);
    glVertex2f(802, 300);
    glVertex2f(802, 450);
    glEnd();
    glColor3f(1.0, 1.0, 0);
    glBegin(GL_POLYGON);
    glVertex2f(802, 450);
    glVertex2f(802, 400);
    glVertex2f(830, 425);
    glEnd();

    glPushMatrix();

```

```

glPushMatrix();
glCallList(basic1);
glPopMatrix();

glPushMatrix();
glTranslatef(x, 0.0, 0.0);
glCallList(train1);
glPopMatrix();
drawString();

if (x > -450)
{
    x = x - 0.1;
}

if (flag == 1)
{
    if (x > -1099)
    {
        x = x - 0.1;
    }
    else
    {
        if (fl == 1)
            fl = 0;
        else
            fl = 1;
        x = 200;

        flag = 0;
    }
}
/*if(flag==0)
{
    if(x>-150)
    {
        x=x-0.1;
    }
}*/

glPopMatrix();
glutPostRedisplay();

glPushMatrix();

```

```
glPushMatrix();  
glCallList(basic2);  
glPopMatrix();
```

```
glPushMatrix();  
glTranslatef(y, 0.0, 0.0);  
glCallList(train2);  
glPopMatrix();
```

```
if (y < 550)  
{  
    y = y + 0.1;  
  
}  
if (flag2 == 1)  
{  
    if (y < 1099)  
    {  
        y = y + 0.1;  
    }  
    else  
    {  
        if (f2 == 1)  
            f2 = 0;  
        else  
            f2 = 1;  
        y = -199;  
        flag2 = 0;  
    }  
}
```

```
glPopMatrix();  
glutPostRedisplay();
```

```
if (flag3 == 0)  
{  
    glColor3f(0.0, 0.0, 0.0);  
    glBegin(GL_POLYGON);  
    glVertex2f(0, 0);  
    glVertex2f(0, 999);  
    glVertex2f(999, 999);  
    glVertex2f(999, 0);  
    glEnd();  
}
```

```

    }
    if (flag3 == 0)
    {
        draw2();
    }
    glFlush();
}

void Idle()
{

}

void keb(unsigned char key, int x, int y)
{
    if (key == 'k' || key == 'K')
        flag = 1;
    if (key == 'g' || key == 'G')
        flag2 = 1;
    if (key == 's' || key == 'S')
        flag3 = 1;
}

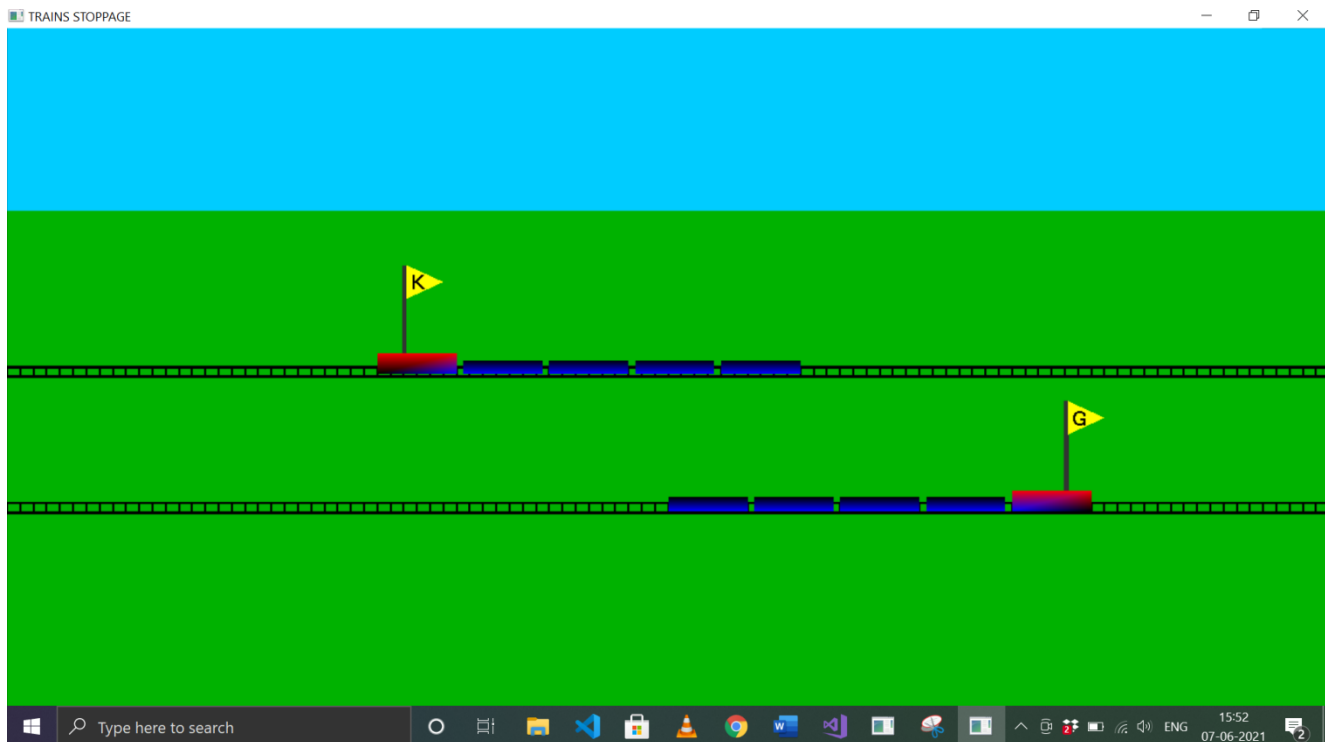
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(0, 0);
    glutInitWindowSize(1200, 700);
    glutCreateWindow("TRAINS STOPPAGE");
    myinit();
    glutDisplayFunc(display);
    glutKeyboardFunc(keb);
    glutIdleFunc(Idle);
    glutMainLoop();
}

```

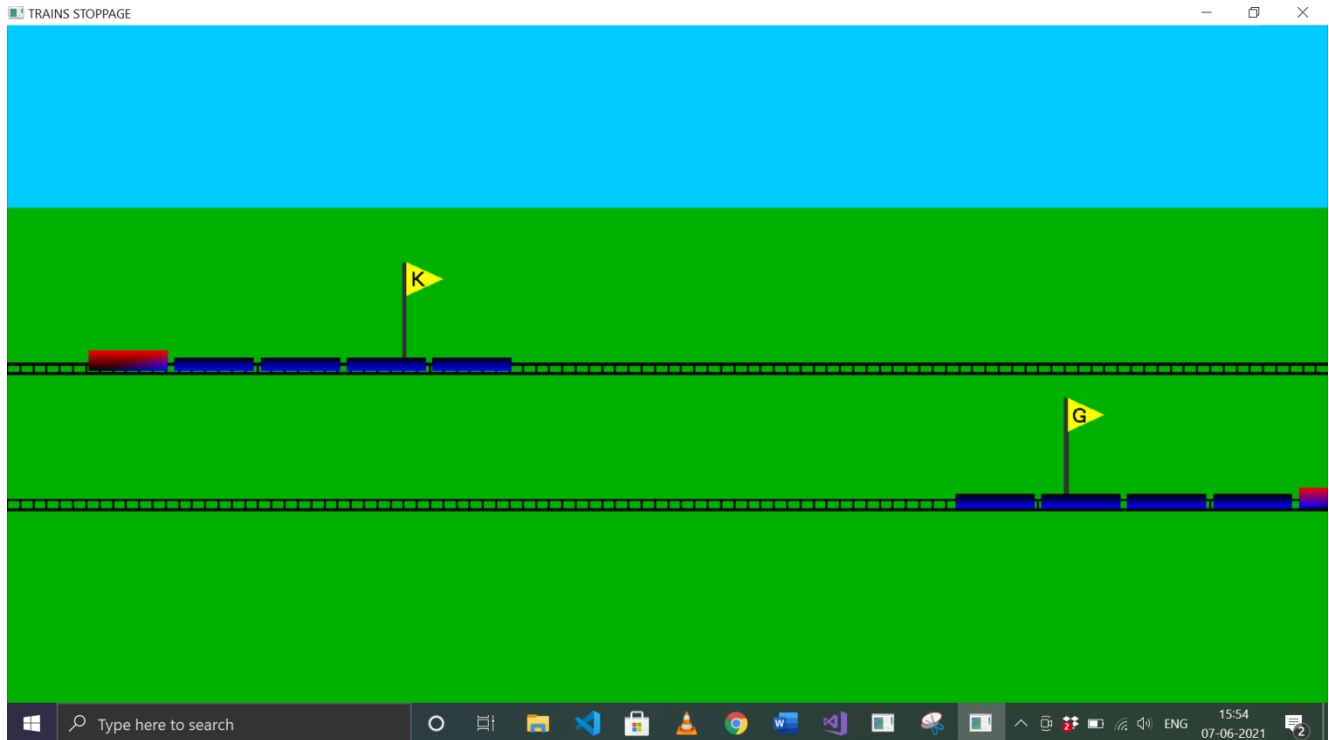
SNAPSHOTS:



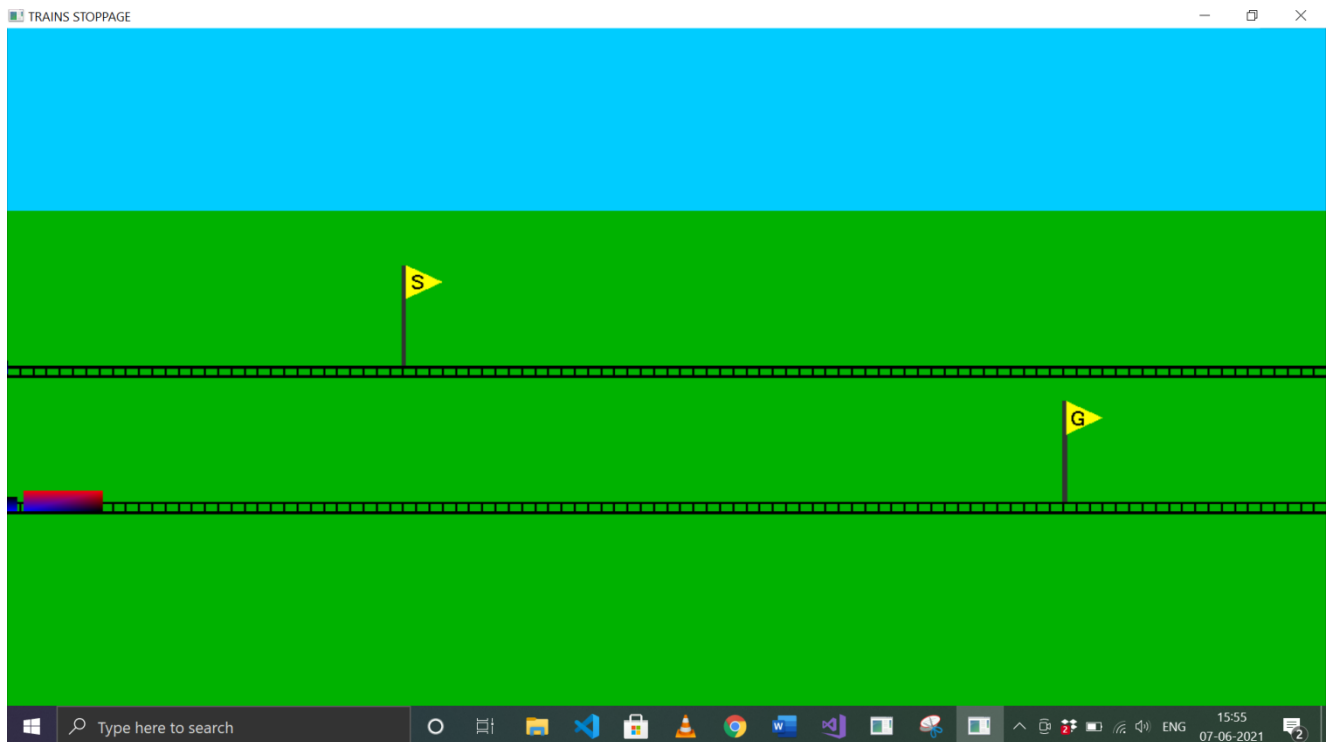
This window will disappear after pressing 's or S' and new window containing project will appear.



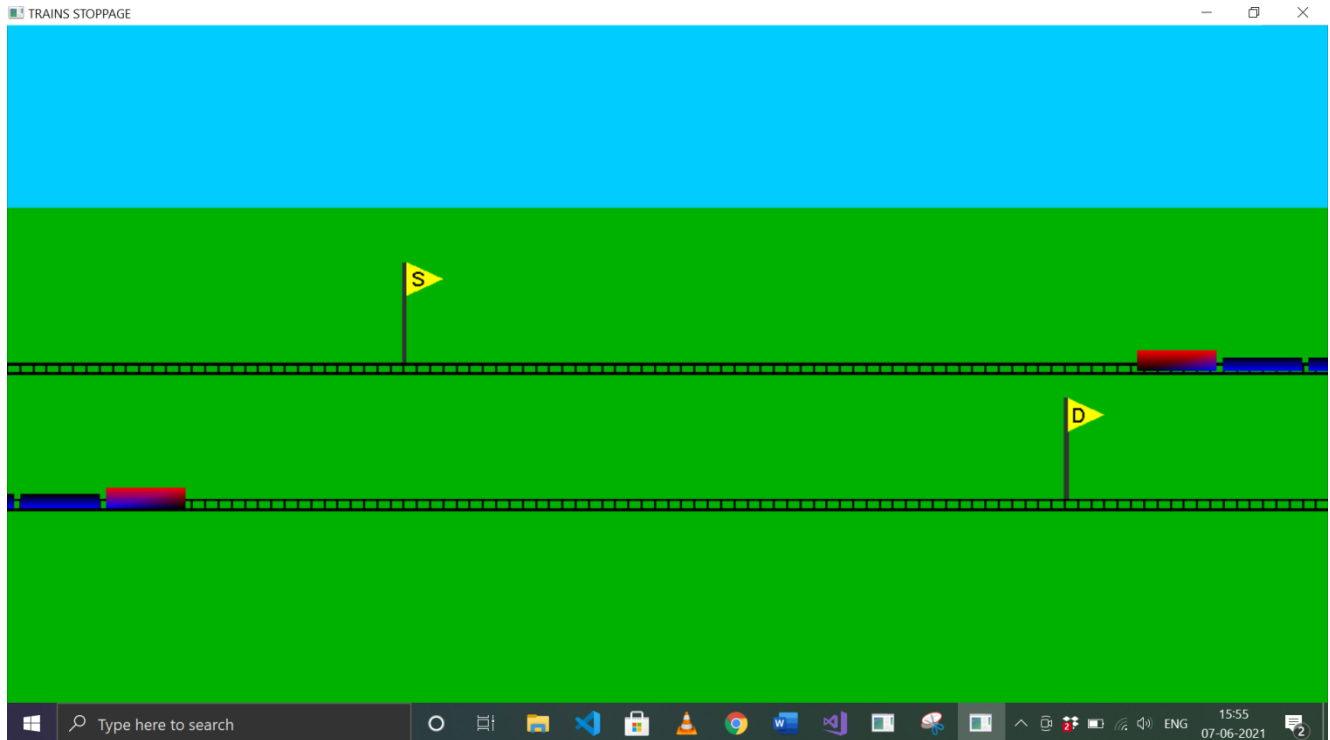
Here one train is moving from left to right and other one from right to left.



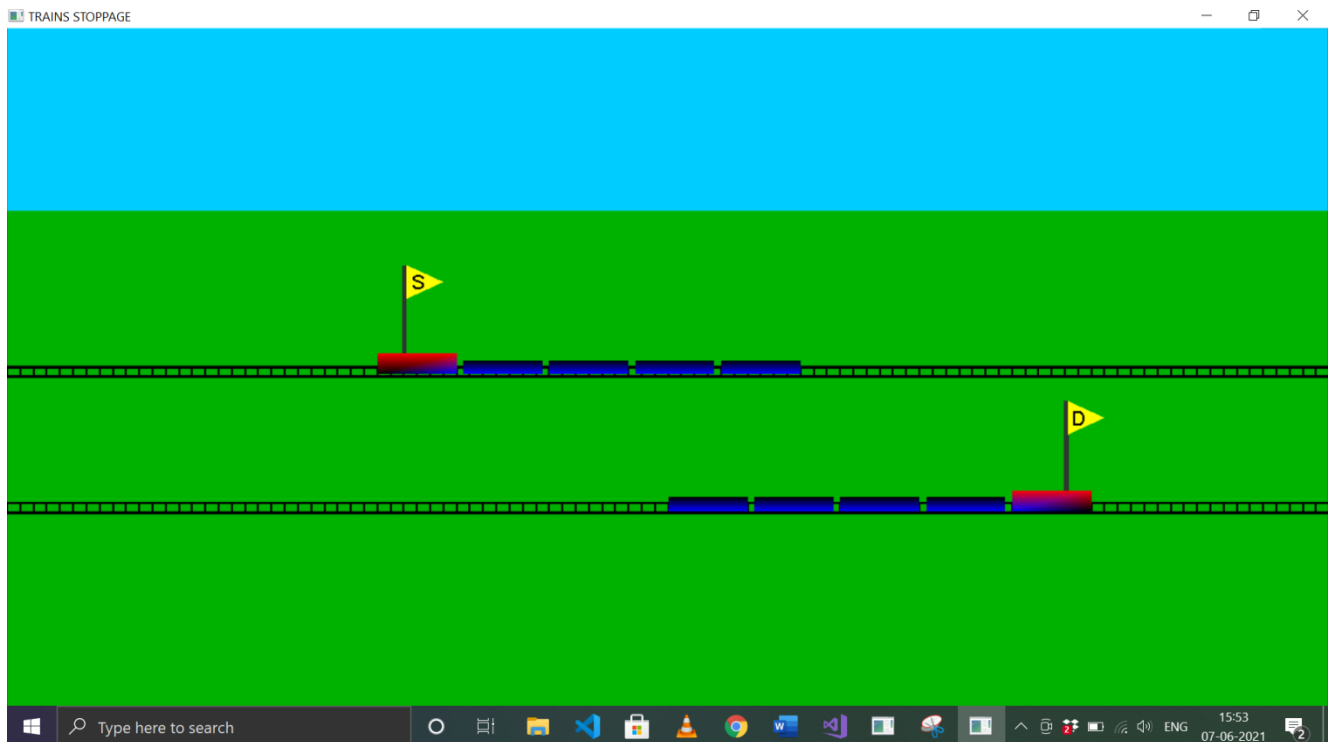
By pressing k and g key both trains started moving.



By pressing the G key the train will start to move till it reaches the end of the window. when the train reaches the end of the window the name of the station is changed from k to s.



Similarly, G is changed with D.



Train will start moving again after pressing k and g.