* **Project Title : Natural Village View.**
* **Course : Computer Graphics**
* **Group : B**
* **Section : C**
* **Group Member:** 
  + ISLAM , MD. HASIBUL (17-34552-2)
  + OLIE , MD. EHSANUL ISLAM (14-28211-3)
  + KEYA , NABILA ZAMAN (16-32881-3)
  + CHOWDHURY , SARIHAN (16-32800-3)
* **Introduction:**

Our project is “Natural Village”. Where a scene of natural village is implemented in opengl. We have mainly created some artifacts in this project, like a house, School, cloudes, tree, train, river, national flag, boat, rain, and helicopter. In the project a school and houses are present in a natural village and in the sky couple of clouds will be kept on moving in the sky and there will be few trees and the train will be moving forward and backward side and there will be few boats in the river which are also moving. Inside the school, we include the national flag. Finally, we can include Day Night mode because we can change the mode from Day to Night and vice versa. An helicopter will be passing in the day and night sky. There is a night mode in which there will be the dark sky and clouds will be moving and the helicopter these two objects will be appear when we press **N** on the keyboard. And when we press **R** on the keyboard raining fall from sky and press **E** raining is stop. This is an overview of the project.

* **Background study:**

To implement this project we had to look into every small details and group them to increase efficiency of function usage. Such as, we can use same set of functions for big tree and small tree or boats can be drawn using same set of functions. After grouping we studied some glut functions which helped us shape the project. We also had to clear our understanding on graphical curves. Without graphical curves we could not draw wheels and other shapes. Also, we had to learn the scaling method for maintaining the resolution of glut screen. After that we chose the appropriate function set that will draw the project items. The list of functions which we studied and used in this project are given below:

1. glClearColor: specify clear values for the color buffers.
2. glPointSize: specify the diameter of rasterized points.
3. gluOrtho2D: define a 2D orthographic projection matrix
4. glVertex2i: specify a vertex
5. glPushMatrix:  push the current matrix stack
6. glTranslatef: multiply the current matrix by a translation matrix
7. glColor3ub: sets the current color
8. glVertex2i: Specifies a vertex
9. glPopMatrix: pop the current matrix stack
10. glScalef: multiply the current matrix by a general scaling matrix.
11. glutTimerFunc: registers a timer callback to be triggered in a specified number of milliseconds.
12. glutKeyboardFunc: sets the keyboard callback for the current window*.*
13. glutInit: A pointer to the program's unmodified argc variable from main. Upon return, the value pointed to by argcp will be updated, because glutInit extracts any command line options intended for the GLUT library

14.glutInitWindowSize: set the initial window position and size respectively

* **IMPLEMENTATION:**
  + **Code:**

#include <iostream>

#include <GL/gl.h>

#include <GL/glut.h>

#include <stdlib.h>

#include <math.h>

#include<windows.h>

#include<mmsystem.h>

using namespace std;

float \_run = 0.0;

float \_run1 = 0.0;

float \_run2 = 0.0;

float \_run3 = 0.0;

float \_rain = 0.0;

float \_nt = 0.0;

float \_ang\_tri = 0.0;

char text[] = "SCHOOL";

bool onOff;

bool frd = false;

bool bck = false;

bool rainday = false;

bool night = false;

//float \_angle = 0.0;

//float \_cameraAngle = 0.0;

//float \_run = 0.0;

void Sprint( float x, float y, char \*st)

{

int l,i;

l=strlen( st ); // see how many characters are in text string.

glColor3f(1.0,0.0,0.0);

//glDisable(GL\_LIGHTING);

glRasterPos2f( x, y); // location to start printing text

for( i=0; i < l; i++) // loop until i is greater then l

{

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, st[i]);

}

}

void init(){

glClearColor(0.0,0.5,0.8,1.0);

glColor3f(0.0,0.0,0.5);

glPointSize(4.0);

gluOrtho2D(0.0,1000.0,0.0,1000.0);

}

void display(){

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);

glColor3ub(0.0, 128, 0.0);

glVertex2i(0,550);

glVertex2i(1000,550);

glVertex2i(1000,0);

glVertex2i(0,0);

glEnd();

glPushMatrix();

glTranslatef(0, 50, 0);

glColor3ub(173, 151, 9); //BODY

glBegin(GL\_QUADS);

glVertex2i(575, 350);

glVertex2i(575, 425);

glVertex2i(825, 425);

glVertex2i(825, 350);

glEnd();

glColor3ub(77, 77, 219); //ROOF

glBegin(GL\_QUADS);

glVertex2i(550, 425);

glVertex2i(600, 460);

glVertex2i(800, 460);

glVertex2i(850, 425);

glEnd();

glColor3ub(139, 137, 143); //STAIR

glBegin(GL\_QUADS);

glVertex2i(565, 340);

glVertex2i(565, 350);

glVertex2i(835, 350);

glVertex2i(835, 340);

glEnd();

glColor3f(1.0, 1.0, 1.0); //DOOR

glBegin(GL\_QUADS);

glVertex2i(690, 350);

glVertex2i(690, 400);

glVertex2i(710, 400);

glVertex2i(710, 350);

glEnd();

glColor3f(1.0, 1.0, 1.0); //Left Windows

glBegin(GL\_QUADS);

glVertex2i(595, 375);

glVertex2i(595, 400);

glVertex2i(615, 400);

glVertex2i(615, 375);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(635, 375);

glVertex2i(635, 400);

glVertex2i(655, 400);

glVertex2i(655, 375);

glEnd();

glColor3f(1.0, 1.0, 1.0); //Right Windows

glBegin(GL\_QUADS);

glVertex2i(805, 375);

glVertex2i(805, 400);

glVertex2i(785, 400);

glVertex2i(785, 375);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(765, 400);

glVertex2i(745, 400);

glVertex2i(745, 375);

glVertex2i(765, 375);

glEnd();

glPopMatrix();

/////circle tree 1

glPushMatrix();

glTranslatef(200, 450, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(210, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(190, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(198, 380);

glVertex2i(198, 425);

glVertex2i(203, 425);

glVertex2i(203, 380);

glEnd();

/////circle tree 2

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(250, 50, 0);

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(200, 450, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(210, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(190, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(198, 380);

glVertex2i(198, 425);

glVertex2i(203, 425);

glVertex2i(203, 380);

glEnd();

glPopMatrix();

///circle tree 3

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(350, 50, 0);

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(200, 450, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(210, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(190, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(198, 380);

glVertex2i(198, 425);

glVertex2i(203, 425);

glVertex2i(203, 380);

glEnd();

glPopMatrix();

///near tree circle tree 4

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(250, 35, 0);

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(200, 450, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(210, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(190, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

//glColor3f(0.4, 0, 0.5);

// glBegin(GL\_QUADS);

//glVertex2i(198, 380);

//glVertex2i(198, 425);

//glVertex2i(203, 425);

//glVertex2i(203, 380);

//glEnd();

glPopMatrix();

///circle tree 4

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(350, 47, 0);

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(200, 450, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(210, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(190, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(198, 380);

glVertex2i(198, 425);

glVertex2i(203, 425);

glVertex2i(203, 380);

glEnd();

glPopMatrix();

///circle tree 5

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(756, 80, 0);

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(200, 450, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(210, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(190, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(198, 395);

glVertex2i(198, 425);

glVertex2i(203, 425);

glVertex2i(203, 395);

glEnd();

glPopMatrix();

///circle tree 7

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(650, 40, 0);

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(200, 450, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(210, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(190, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(198, 380);

glVertex2i(198, 425);

glVertex2i(203, 425);

glVertex2i(203, 380);

glEnd();

glPopMatrix();

///circle tree 6 near tree

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(50, 34, 0);

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(200, 450, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(210, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(190, 438, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0, 0.8, 0.5);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(198, 380);

glVertex2i(198, 425);

glVertex2i(203, 425);

glVertex2i(203, 380);

glEnd();

glPopMatrix();

//hill

glColor3ub(34, 177, 76);

glBegin(GL\_QUADS);

glVertex2i(0,550);

glVertex2i(0,560);

glVertex2i(100,650);

glVertex2i(200,550);

glEnd();

glColor3ub(34, 177, 76);

glBegin(GL\_TRIANGLES);

glVertex2i(400,550);

glVertex2i(500,650);

glVertex2i(600,550);

glEnd();

//hill

glColor3ub(34, 177, 76);

glBegin(GL\_TRIANGLES);

glVertex2i(100,550);

glVertex2i(300,650);

glVertex2i(500,550);

glEnd();

glColor3ub(34, 177, 76);

glBegin(GL\_TRIANGLES);

glVertex2i(600,550);

glVertex2i(700,650);

glVertex2i(800,550);

glEnd();

//hill

glColor3ub(34, 177, 76);

glBegin(GL\_TRIANGLES);

glVertex2i(800,550);

glVertex2i(1000,650);

glVertex2i(1000,550);

glEnd();

glColor3ub(34, 177, 76);

glBegin(GL\_TRIANGLES);

glVertex2i(700,550);

glVertex2i(900,650);

glVertex2i(1000,550);

glEnd();

//road

glColor3ub(66, 66, 49);

glBegin(GL\_QUADS);

glVertex2i(0,330);

glVertex2i(1000,330);

glVertex2i(1000,280);

glVertex2i(0,280);

glEnd();

glColor3ub(66, 66, 49);

glBegin(GL\_QUADS);

glVertex2i(690,390);

glVertex2i(710,390);

glVertex2i(710,320);

glVertex2i(690,320);

glEnd();

//rail line

glColor3ub(150, 150, 144);

glBegin(GL\_QUADS);

glVertex2i(0,180);

glVertex2i(0,205);

glVertex2i(1000,205);

glVertex2i(1000,180);

glEnd();

glColor3ub(0, 0, 0);

glBegin(GL\_QUADS);

glVertex2i(0,202);

glVertex2i(1000,202);

glVertex2i(1000,205);

glVertex2i(0,205);

glEnd();

glColor3ub(0, 0, 0);

glBegin(GL\_QUADS);

glVertex2i(0,180);

glVertex2i(1000,180);

glVertex2i(1000,184);

glVertex2i(0,184);

glEnd();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

float j;

for(j=0;j<=1000;j+=20) //rail line

{

glVertex2i(10+j,180);

glVertex2i(15+j,205);

}

glEnd();

//TREE 1

glPushMatrix();

glScalef(0.5, 0.5, 0.5);

glTranslatef(200, 440, 0);

glColor3f(0, 0.8, 0.2);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 340);

glVertex2i(492, 440);

glVertex2i(540, 340);

glEnd();

glColor3f(0, 0.8, 0.5);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 360);

glVertex2i(492, 460);

glVertex2i(540, 360);

glEnd();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(497, 340);

glVertex2i(486, 340);

glVertex2i(486, 250);

glVertex2i(497, 250);

glEnd();

glPopMatrix();

//TREE 2

glPushMatrix();

glScalef(0.50, 0.50, 0.25);

glTranslatef(1400, 500, 0);

glColor3f(0, 0.8, 0.2);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 340);

glVertex2i(492, 440);

glVertex2i(540, 340);

glEnd();

glColor3f(0, 0.8, 0.5);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 360);

glVertex2i(492, 460);

glVertex2i(540, 360);

glEnd();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(497, 340);

glVertex2i(486, 340);

glVertex2i(486, 250);

glVertex2i(497, 250);

glEnd();

glPopMatrix();

//tree 3

glPushMatrix();

glScalef(0.50, 0.50, 0.25);

glTranslatef(-400, 500, 0);

glColor3f(0, 0.8, 0.2);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 340);

glVertex2i(492, 440);

glVertex2i(540, 340);

glEnd();

glColor3f(0, 0.8, 0.5);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 360);

glVertex2i(492, 460);

glVertex2i(540, 360);

glEnd();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(497, 340);

glVertex2i(486, 340);

glVertex2i(486, 250);

glVertex2i(497, 250);

glEnd();

glPopMatrix();

//tree 4

glPushMatrix();

glScalef(0.50, 0.50, 0.25);

glTranslatef(250, 600, 0);

glColor3f(0, 0.8, 0.2);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 340);

glVertex2i(492, 440);

glVertex2i(540, 340);

glEnd();

glColor3f(0, 0.8, 0.5);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 360);

glVertex2i(492, 460);

glVertex2i(540, 360);

glEnd();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(497, 340);

glVertex2i(486, 340);

glVertex2i(486, 250);

glVertex2i(497, 250);

glEnd();

glPopMatrix();

//tree 5

glPushMatrix();

glTranslatef(0, 400, 0);

glScalef(0.25, 0.25, 0.25);

glColor3f(0, 0.8, 0.2);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 340);

glVertex2i(492, 440);

glVertex2i(540, 340);

glEnd();

glColor3f(0, 0.8, 0.5);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 360);

glVertex2i(492, 460);

glVertex2i(540, 360);

glEnd();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(497, 340);

glVertex2i(486, 340);

glVertex2i(486, 250);

glVertex2i(497, 250);

glEnd();

glPopMatrix();

//tree 6

glPushMatrix();

glTranslatef(150, 420, 0);

glScalef(0.25, 0.25, 0.25);

glColor3f(0, 0.8, 0.2);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 340);

glVertex2i(492, 440);

glVertex2i(540, 340);

glEnd();

glColor3f(0, 0.8, 0.5);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 360);

glVertex2i(492, 460);

glVertex2i(540, 360);

glEnd();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(497, 340);

glVertex2i(486, 340);

glVertex2i(486, 250);

glVertex2i(497, 250);

glEnd();

glPopMatrix();

//tree 7

glPushMatrix();

glTranslatef(-50, 400, 0);

glScalef(0.25, 0.25, 0.25);

glColor3f(0, 0.8, 0.2);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 340);

glVertex2i(492, 440);

glVertex2i(540, 340);

glEnd();

glColor3f(0, 0.8, 0.5);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 360);

glVertex2i(492, 460);

glVertex2i(540, 360);

glEnd();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(497, 340);

glVertex2i(486, 340);

glVertex2i(486, 250);

glVertex2i(497, 250);

glEnd();

glPopMatrix();

//tree 8

glPushMatrix();

glTranslatef(80, 420, 0);

glScalef(0.25, 0.25, 0.25);

glColor3f(0, 0.8, 0.2);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 340);

glVertex2i(492, 440);

glVertex2i(540, 340);

glEnd();

glColor3f(0, 0.8, 0.5);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 360);

glVertex2i(492, 460);

glVertex2i(540, 360);

glEnd();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(497, 340);

glVertex2i(486, 340);

glVertex2i(486, 250);

glVertex2i(497, 250);

glEnd();

glPopMatrix();

//tree 9

glPushMatrix();

glTranslatef(780, 415, 0);

glScalef(0.25, 0.25, 0.25);

glColor3f(0, 0.8, 0.2);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 340);

glVertex2i(492, 440);

glVertex2i(540, 340);

glEnd();

glColor3f(0, 0.8, 0.5);

glBegin(GL\_TRIANGLES);

glVertex2i(445, 360);

glVertex2i(492, 460);

glVertex2i(540, 360);

glEnd();

glColor3f(0.4, 0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(497, 340);

glVertex2i(486, 340);

glVertex2i(486, 250);

glVertex2i(497, 250);

glEnd();

glPopMatrix();

///river

glPushMatrix();

glColor3ub(152, 222, 245);

glBegin(GL\_QUADS);

glVertex2i(0, 120);

glVertex2i(1000, 120);

glVertex2i(1000, 0);

glVertex2i(0, 0);

glEnd();

glPopMatrix();

///Left Moving Boat

glPushMatrix();

glTranslatef(-\_run, 0.0, 0.0);

///boat 2

glPushMatrix();

glTranslatef(200, 35, 0);

glColor3f(0.423, 0.329, 0.588);

glBegin(GL\_QUADS);

glVertex2i(130, 40);

glVertex2i(110, 60);

glVertex2i(190, 60);

glVertex2i(170, 40);

glEnd();

glColor3f(0.329, 0.517, 0.588);

glBegin(GL\_QUADS);

glVertex2i(150, 80);

glVertex2i(170, 80);

glVertex2i(170, 60);

glVertex2i(150, 60);

glEnd();

glPopMatrix();

glPopMatrix();

///Right Moving Boat

glPushMatrix();

glTranslatef(\_run, 0.0, 0.0);

///boat 1

glPushMatrix();

glColor3f(1, 0, 0);

glBegin(GL\_QUADS);

glVertex2i(130, 40);

glVertex2i(110, 60);

glVertex2i(190, 60);

glVertex2i(170, 40);

glEnd();

glColor3f(0, 1, 0);

glBegin(GL\_QUADS);

glVertex2i(130, 80);

glVertex2i(150, 80);

glVertex2i(150, 60);

glVertex2i(130, 60);

glEnd();

glPopMatrix();

///boat 2

glPushMatrix();

glTranslatef(500, 15, 0);

glColor3f(0.5, 0.0, 0.0);

glBegin(GL\_QUADS);

glVertex2i(130, 40);

glVertex2i(110, 60);

glVertex2i(190, 60);

glVertex2i(170, 40);

glEnd();

glColor3f(0.5, 0.0, 0.5);

glBegin(GL\_QUADS);

glVertex2i(130, 80);

glVertex2i(150, 80);

glVertex2i(150, 60);

glVertex2i(130, 60);

glEnd();

glPopMatrix();

///boat 3

glPushMatrix();

glTranslatef(300, 0, 0);

glColor3f(1, 0, 1);

glBegin(GL\_QUADS);

glVertex2i(130, 40);

glVertex2i(110, 60);

glVertex2i(190, 60);

glVertex2i(170, 40);

glEnd();

glColor3f(0, 0, 1);

glBegin(GL\_QUADS);

glVertex2i(130, 80);

glVertex2i(150, 80);

glVertex2i(150, 60);

glVertex2i(130, 60);

glEnd();

glPopMatrix();

glPopMatrix();

///Left Moving Boat

glPushMatrix();

glTranslatef(-\_run, 0.0, 0.0);

///boat 3

glPushMatrix();

glTranslatef(700, -30, 0);

glColor3f(0.247, 0.505, 0.231);

glBegin(GL\_QUADS);

glVertex2i(130, 40);

glVertex2i(100, 65);

glVertex2i(220, 65);

glVertex2i(190, 40);

glEnd();

glColor3f(0.145, 0.156, 0.337);

glBegin(GL\_QUADS);

glVertex2i(190, 88);

glVertex2i(150, 88);

glVertex2i(150, 65);

glVertex2i(190, 65);

glEnd();

glPopMatrix();

glPopMatrix();

///1st CART

glPushMatrix();

glTranslatef(\_run1, 0.0, 0.0);

glPushMatrix();

glTranslatef(40, 80, 0.0);

glPushMatrix();

glTranslatef(30, 220, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(225, 245, 93);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=10;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

///box of 1st CART

glColor3ub(242, 242, 237);

glBegin(GL\_LINES);

glVertex2i(20,220);

glVertex2i(40,220);

glVertex2i(30,230);

glVertex2i(30,210);

glVertex2i(37,227);

glVertex2i(23,213);

glVertex2i(37,213);

glVertex2i(23,227);

glVertex2i(5,230);

glVertex2i(100,230);

glVertex2i(100,230);

glVertex2i(100,232);

glVertex2i(100,232);

glVertex2i(5,232);

glVertex2i(7,232);

glVertex2i(7,262);

glVertex2i(17,232);

glVertex2i(17,262);

glVertex2i(27,232);

glVertex2i(27,262);

glVertex2i(37,232);

glVertex2i(37,262);

glVertex2i(47,232);

glVertex2i(47,262);

glVertex2i(57,232);

glVertex2i(57,262);

glVertex2i(7,242);

glVertex2i(59,242);

glVertex2i(7,252);

glVertex2i(59,252);

glVertex2i(99,233);

glVertex2i(97,245);

glVertex2i(101,233);

glVertex2i(103,245);

glEnd();

///COW of 1st CART

glColor3ub(242, 242, 237);

glBegin(GL\_LINE\_LOOP);

glVertex2i(60,210);

glVertex2i(60,230);

glVertex2i(90,230);

glVertex2i(92,233);

glVertex2i(95,230);

glVertex2i(97,230);

glVertex2i(99,233);

glVertex2i(101,233);

glVertex2i(105,226);

glVertex2i(102,226);

glVertex2i(100,228);

glVertex2i(98,228);

glVertex2i(96,220);

glVertex2i(95,210);

glVertex2i(93,210);

glVertex2i(93,220);

glVertex2i(93,216);

glVertex2i(65,216);

glVertex2i(65,220);

glVertex2i(65,210);

glEnd();

glPopMatrix();

///End of COW of 1st Cart

//2nd CART

glPushMatrix();

glTranslatef(200, 100, 0.0);

glPushMatrix();

glTranslatef(30, 220, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(225, 245, 93);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=10;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

//box of 2nd CART

glColor3ub(242, 242, 237);

glBegin(GL\_LINES);

glVertex2i(20,220);

glVertex2i(40,220);

glVertex2i(30,230);

glVertex2i(30,210);

glVertex2i(37,227);

glVertex2i(23,213);

glVertex2i(37,213);

glVertex2i(23,227);

glVertex2i(5,230);

glVertex2i(100,230);

glVertex2i(100,230);

glVertex2i(100,232);

glVertex2i(100,232);

glVertex2i(5,232);

glVertex2i(7,232);

glVertex2i(7,262);

glVertex2i(17,232);

glVertex2i(17,262);

glVertex2i(27,232);

glVertex2i(27,262);

glVertex2i(37,232);

glVertex2i(37,262);

glVertex2i(47,232);

glVertex2i(47,262);

glVertex2i(57,232);

glVertex2i(57,262);

glVertex2i(7,242);

glVertex2i(59,242);

glVertex2i(7,252);

glVertex2i(59,252);

glVertex2i(99,233);

glVertex2i(97,245);

glVertex2i(101,233);

glVertex2i(103,245);

glEnd();

//COW of 2nd CART

glColor3ub(242, 242, 237);

glBegin(GL\_LINE\_LOOP);

glVertex2i(60,210);

glVertex2i(60,230);

glVertex2i(90,230);

glVertex2i(92,233);

glVertex2i(95,230);

glVertex2i(97,230);

glVertex2i(99,233);

glVertex2i(101,233);

glVertex2i(105,226);

glVertex2i(102,226);

glVertex2i(100,228);

glVertex2i(98,228);

glVertex2i(96,220);

glVertex2i(95,210);

glVertex2i(93,210);

glVertex2i(93,220);

glVertex2i(93,216);

glVertex2i(65,216);

glVertex2i(65,220);

glVertex2i(65,210);

glEnd();

glPopMatrix();

glPopMatrix();

//End of COW of 2nd Cart

//small home

glPushMatrix();

glTranslatef(-10, 70, 0.0);

glColor3ub(182, 163, 240);

glBegin(GL\_POLYGON);

glVertex2i(85,310);

glVertex2i(85,350);

glVertex2i(130,350);

glVertex2i(130,310);

glEnd();

glColor3ub(240, 242, 242);

glBegin(GL\_POLYGON);

glVertex2i(100,310);

glVertex2i(100,338);

glVertex2i(115,338);

glVertex2i(115,310);

glEnd();

glColor3ub(55, 161, 163);

glBegin(GL\_TRIANGLES);

glVertex2i(80,350);

glVertex2i(107,380);

glVertex2i(135,350);

glEnd();

glColor3f(0.850, 0.490, 0.756);

glBegin(GL\_QUADS);

glVertex2i(80,310);

glVertex2i(80,300);

glVertex2i(135,300);

glVertex2i(135,310);

glEnd();

glPopMatrix();

//small home2

glPushMatrix();

glTranslatef(60, 70, 0.0);

glColor3ub(182, 163, 240);

glBegin(GL\_POLYGON);

glVertex2i(85,310);

glVertex2i(85,350);

glVertex2i(130,350);

glVertex2i(130,310);

glEnd();

glColor3ub(240, 242, 242);

glBegin(GL\_POLYGON);

glVertex2i(100,310);

glVertex2i(100,338);

glVertex2i(115,338);

glVertex2i(115,310);

glEnd();

glColor3ub(55, 161, 163);

glBegin(GL\_TRIANGLES);

glVertex2i(80,350);

glVertex2i(107,380);

glVertex2i(135,350);

glEnd();

glColor3f(0.850, 0.490, 0.756);

glBegin(GL\_QUADS);

glVertex2i(80,310);

glVertex2i(80,300);

glVertex2i(135,300);

glVertex2i(135,310);

glEnd();

glPopMatrix();

///Making of Rail Body

glPushMatrix();

glTranslatef(\_ang\_tri, 0.0, 0.0);

glColor3f(0.192, 0.576, 0.705);

glBegin(GL\_QUADS);

glVertex2i(100, 200);

glVertex2i(170, 200);

glVertex2i(170, 250);

glVertex2i(100, 250);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(110, 225);

glVertex2i(120, 225);

glVertex2i(120, 240);

glVertex2i(110, 240);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(130, 225);

glVertex2i(140, 225);

glVertex2i(140, 240);

glVertex2i(130, 240);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(150, 225);

glVertex2i(160, 225);

glVertex2i(160, 240);

glVertex2i(150, 240);

glEnd();

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_QUADS);

glVertex2i(100, 210);

glVertex2i(170, 210);

glVertex2i(170, 215);

glVertex2i(100, 215);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(170, 200);

glVertex2i(176, 210);

glVertex2i(176, 260);

glVertex2i(170, 250);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(101, 250);

glVertex2i(170, 250);

glVertex2i(176, 260);

glVertex2i(105, 260);

glEnd();

glPushMatrix();

glTranslatef(115, 192, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(109, 109, 115);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=10;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(155, 192, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(109, 109, 115);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=10;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(75, 0, 0);

glColor3f(0.192, 0.576, 0.705);

glBegin(GL\_QUADS);

glVertex2i(100, 200);

glVertex2i(170, 200);

glVertex2i(170, 250);

glVertex2i(100, 250);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(110, 225);

glVertex2i(120, 225);

glVertex2i(120, 240);

glVertex2i(110, 240);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(130, 225);

glVertex2i(140, 225);

glVertex2i(140, 240);

glVertex2i(130, 240);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(150, 225);

glVertex2i(160, 225);

glVertex2i(160, 240);

glVertex2i(150, 240);

glEnd();

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_QUADS);

glVertex2i(100, 210);

glVertex2i(170, 210);

glVertex2i(170, 215);

glVertex2i(100, 215);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(170, 200);

glVertex2i(176, 210);

glVertex2i(176, 260);

glVertex2i(170, 250);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(101, 250);

glVertex2i(170, 250);

glVertex2i(176, 260);

glVertex2i(105, 260);

glEnd();

glPushMatrix();

glTranslatef(115, 192, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(109, 109, 115);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=10;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(155, 192, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(109, 109, 115);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=10;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

glTranslatef(-75, 0, 0);

glColor3f(0.192, 0.576, 0.705);

glBegin(GL\_QUADS);

glVertex2i(100, 200);

glVertex2i(170, 200);

glVertex2i(170, 250);

glVertex2i(100, 250);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(110, 225);

glVertex2i(120, 225);

glVertex2i(120, 240);

glVertex2i(110, 240);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(130, 225);

glVertex2i(140, 225);

glVertex2i(140, 240);

glVertex2i(130, 240);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(150, 225);

glVertex2i(160, 225);

glVertex2i(160, 240);

glVertex2i(150, 240);

glEnd();

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_QUADS);

glVertex2i(100, 210);

glVertex2i(170, 210);

glVertex2i(170, 215);

glVertex2i(100, 215);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(170, 200);

glVertex2i(176, 210);

glVertex2i(176, 260);

glVertex2i(170, 250);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(101, 250);

glVertex2i(170, 250);

glVertex2i(176, 260);

glVertex2i(105, 260);

glEnd();

glPushMatrix();

glTranslatef(115, 192, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(109, 109, 115);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=10;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(155, 192, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(109, 109, 115);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=10;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

glTranslatef(150, 0, 0);

glColor3f(0.192, 0.576, 0.705);

glBegin(GL\_QUADS);

glVertex2i(100, 200);

glVertex2i(170, 200);

glVertex2i(170, 250);

glVertex2i(100, 250);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(110, 225);

glVertex2i(120, 225);

glVertex2i(120, 240);

glVertex2i(110, 240);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(130, 225);

glVertex2i(140, 225);

glVertex2i(140, 240);

glVertex2i(130, 240);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(150, 225);

glVertex2i(160, 225);

glVertex2i(160, 240);

glVertex2i(150, 240);

glEnd();

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_QUADS);

glVertex2i(100, 210);

glVertex2i(170, 210);

glVertex2i(170, 215);

glVertex2i(100, 215);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(170, 200);

glVertex2i(176, 210);

glVertex2i(176, 260);

glVertex2i(170, 250);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_QUADS);

glVertex2i(101, 250);

glVertex2i(170, 250);

glVertex2i(176, 260);

glVertex2i(105, 260);

glEnd();

glPushMatrix();

glTranslatef(115, 192, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(109, 109, 115);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=10;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(155, 192, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(109, 109, 115);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=10;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_QUADS);

glVertex2i(322, 245);

glVertex2i(330, 245);

glVertex2i(330, 232);

glVertex2i(322, 232);

glEnd();

glColor3f(0.325, 0.101, 0.619);

glBegin(GL\_QUADS);

glVertex2i(310, 253);

glVertex2i(300, 253);

glVertex2i(300, 258);

glVertex2i(310, 258);

glEnd();

glPushMatrix();

glTranslatef(303, 267, 0.0);

glBegin(GL\_POLYGON);

glColor3f(0.709, 0.701, 0.717);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=3;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(293, 270, 0.0);

glBegin(GL\_POLYGON);

glColor3f(0.709, 0.701, 0.717);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=4;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(283, 272, 0.0);

glBegin(GL\_POLYGON);

glColor3f(0.709, 0.701, 0.717);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=5;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

///End of Making of Rail Body

//flag

glPushMatrix();

glTranslatef(0, 40, 0);

glColor3ub(156, 143, 6); //STAIR

glBegin(GL\_QUADS);

glVertex2i(465, 340);

glVertex2i(465, 350);

glVertex2i(490, 350);

glVertex2i(490, 340);

glEnd();

glColor3ub(104, 138, 143); //STAND

glBegin(GL\_QUADS);

glVertex2i(470, 350);

glVertex2i(470, 425);

glVertex2i(485, 425);

glVertex2i(485, 350);

glEnd();

glColor3ub(9, 107, 4); //QUAD

glBegin(GL\_QUADS);

glVertex2i(485, 390);

glVertex2i(485, 425);

glVertex2i(550, 425);

glVertex2i(550, 390);

glEnd();

//Circle

glPushMatrix();

// glScalef(0.5, 0.5, 0.5);

glTranslatef(516, 407, 0);

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(1,0,0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=12;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPopMatrix();

///Cloud

glPushMatrix();

glTranslatef(\_run3, 0.0, 0.0);

glPushMatrix();

glTranslatef(200, 800, 0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=40;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(180, 750, 0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=50;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(250, 800, 0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=40;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(230, 750, 0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=50;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

//cloud 2

glPushMatrix();

glTranslatef(350, 100, 0);

glPushMatrix();

glTranslatef(200, 800, 0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=40;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(180, 750, 0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=50;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(250, 800, 0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=50;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(230, 750, 0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=50;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(300, 800, 0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=40;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(290, 760, 0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=40;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

glPopMatrix();

//end of cloud

//Helicopter

glPushMatrix();

glTranslatef(\_run2, 0.0, 0.0);

glBegin(GL\_POLYGON);

glColor3ub(163, 14, 225);

glVertex2i(20,850);

glVertex2i(30,820);

glVertex2i(100,820);

glVertex2i(110,800);

glVertex2i(200,800);

glVertex2i(200,815);

glVertex2i(190,830);

glVertex2i(180,830);

glVertex2i(170,860);

glVertex2i(155,860);

glVertex2i(150,870);

glVertex2i(135,860);

glVertex2i(115,860);

glVertex2i(100,825);

glVertex2i(30,850);

glEnd();

glColor3ub(225, 19, 14);

glBegin(GL\_TRIANGLES);

glVertex2i(200,800);

glVertex2i(200,815);

glVertex2i(215,808);

glEnd();

glPushMatrix();

glTranslatef(28, 850, 0);

glBegin(GL\_LINE\_LOOP);

glColor3ub(225, 19, 14);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=15;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(28, 850, 0);

glBegin(GL\_LINE\_LOOP);

glColor3ub(225, 19, 14);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=16;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glColor3ub(225, 19, 14);

glBegin(GL\_TRIANGLES);

glVertex2i(200,800);

glVertex2i(200,815);

glVertex2i(215,808);

glEnd();

glPushMatrix();

glTranslatef(142, 870, 0);

glBegin(GL\_LINE\_LOOP);

glColor3ub(225, 19, 14);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=30;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(142, 870, 0);

glBegin(GL\_LINE\_LOOP);

glColor3ub(225, 19, 14);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=29;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(130, 790, 0);

glBegin(GL\_POLYGON);

glColor3ub(44, 42, 45);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=8;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(165, 790, 0);

glBegin(GL\_POLYGON);

glColor3ub(44, 42, 45);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=8;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPopMatrix();

//End of Helicopter

//text school

glColor3f(1.0, 1.0, 1.0); //QUAD

glBegin(GL\_QUADS);

glVertex2i(675, 495);

glVertex2i(725, 495);

glVertex2i(725, 475);

glVertex2i(675, 475);

glEnd();

glPushMatrix();

glTranslatef(680, 478, 0);

Sprint(1.0,0,text);

glPopMatrix();

// end of text

glFlush();

glutSwapBuffers();

}

void update(int value) {

\_run += 1.0f;

if (\_run > 1000)

{

\_run -= 1700;

}

\_run1 += 1.0f;

if (\_run1 > 1000)

{

\_run1 -= 1300;

}

\_run2 += 3.5f;

if (\_run2 > 1000)

{

\_run2 -= 1300;

}

\_run3 += 0.8f;

if (\_run3 > 1000)

{

\_run3 -= 1700;

}

if(onOff){

\_ang\_tri += 2.5f;

if (\_ang\_tri > 1000){

\_ang\_tri = 1300;

}

}

glutPostRedisplay(); //Tell GLUT that the display has changed

glutTimerFunc(25, update, 0);

}

void railForward(int value){

if(frd){

\_ang\_tri += 2.2f;

if (\_ang\_tri > 1000) {

\_ang\_tri -= 1400;

}

glutPostRedisplay();

glutTimerFunc(25, railForward, 0);

}

}

void railBackward(int value){

if(bck){

\_ang\_tri -= 2.2f;

if (\_ang\_tri < -350) {

\_ang\_tri = 1100;

}

glutPostRedisplay();

glutTimerFunc(25, railBackward, 0);

}

}

void Rain(int value){

if(rainday){

\_rain += 0.01f;

glBegin(GL\_POINTS);

for(int i=1;i<=1000;i++)

{

int x=rand(),y=rand();

x%=1000; y%=1000;

glBegin(GL\_LINES);

glColor3f(1.0, 1.0, 1.0);

glVertex2d(x,y);

glVertex2d(x+5,y+5);

glEnd();

}

glutPostRedisplay();

glutTimerFunc(5, Rain, 0);

glFlush();

}

}

void Night(int value){

if(night){

/\* glBegin(GL\_QUADS);

glColor3f(0.10, 0.10, 0.10);

glVertex2i(0,1000);

glVertex2i(1000,1000);

glVertex2i(1000,0);

glVertex2i(0,0);

glEnd();\*/

glClearColor(0.0,0.0,0.0,0.0);

glutPostRedisplay();

glutTimerFunc(5, Night, 0);

glFlush();

}

}

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

switch (key)

{

case 'a':

frd = false;

bck = true;

railBackward(\_ang\_tri);

break;

case 'd':

frd = true;

bck = false;

railForward(\_ang\_tri);

break;

case 's':

frd = false;

bck = false;

break;

case 'r':

rainday = true;

Rain(\_rain);

sndPlaySound("River.wav",SND\_ASYNC|SND\_LOOP);

//sndPlaySound("River.wav",SND\_MEMORY|SND\_ASYNC);

break;

case 'e':

rainday = false;

sndPlaySound(NULL,SND\_ASYNC);

break;

case 'n':

night = true;

Night(\_nt);

break;

case 'b':

night = false;

glClearColor(0.0,0.5,0.8,1.0);

break;

case 27: // ESC key

exit(0);

break;

default:

break;

}

}

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

{

cout << endl << "\*\*\*\*\*\*\*\*\*\*\* Natural View Of A Village \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl << endl;

cout << "Press D : To Forward the Train" << endl << endl;

cout << "Press A : To Backward the Train" << endl << endl;

cout << "Press S : To Stop the Train" << endl << endl;

cout << "Press R : For Rain " << endl << endl;

cout << "Press E : For Stop Rain" << endl << endl;

cout << "Press N : For Night " << endl << endl;

cout << "Press B : For Day" << endl << endl;

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE| GLUT\_RGB | GLUT\_DEPTH);

glutInitWindowSize(1200, 600);

glutCreateWindow("Natural View Of A Village"); // creating the window

//glutFullScreen(); // making the window full screen

//glutInitWindowPosition(0,0);

glutDisplayFunc(display);

glutKeyboardFunc(myKeyboard);

glutTimerFunc(25, update, 0);

init();

glutMainLoop();

return 0;

}

* **Function list:**
* display()
* glClear()
* glBegin()
* glEnd()
* glColor3ub()
* glVertex2i()
* glPushMatrix()
* glTranslaterf()
* glPopMatrix()
* glColor3f()
* glScalef()
* update()
* sprint()
* glFlush()
* glutSwapBuffers()
* railforward()
* railBackward()
* rain()
* glutPostRedisplay()
* glutTimerFunc()
* Night()
* glClearColor()
* myKeyboard()
* main()
* glutInit()
* glutInitDisplayMode()
* glutInitWindowSize()
* glutCreateWindow()
* glutDisplayFunc()
* glutKeyboardFunc()
* glutTimerFunc()
* init()
* glutMainLoop()
* **Conclusion:**

We have put in place a “NATURAL VILLAGE‘’ which is a colorful and simple project. And this project includes a lot of options in it. We have given you the idea to implement the projects as we have used in this project you can also develop projects to this project and it will look even better. For the future implementations try to create a sun and some birds moving in the sky and etc.