**Lab Taks-4**

|  |
| --- |
| **Question- 1**  Draw the scenario of a traffic signal |
| **Graph Plot (Picture)-**  **(Not Needed)** |
| **/\***  **\* GL02Primitive.cpp: Vertex, Primitive and Color**  **\* Draw Simple 2D colored Shapes: quad, triangle and polygon.**  **\*/**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **#include <cmath>**  **#include <math.h>**  **#include <cstdlib>**  **#include <ctime>**  **/\* Initialize OpenGL Graphics \*/**  **void initGL() {**  **// Set "clearing" or background color**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black and opaque**  **}**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void circle(float radius, float xc, float yc, float r, float g, float b)**  **{**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **for(int i=0;i<200;i++)**  **{**  **glColor3ub(r,g,b);**  **float pi=3.1416;**  **float A=(i\*2\*pi)/200;**  **float r=radius;**  **float x = r \* cos(A);**  **float y = r \* sin(A);**  **glVertex2f(x+xc,y+yc);**  **}**  **glEnd();**  **}**  **void roadDev(float x,float y){**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3ub(45, 53, 66); // Yellow**  **glVertex2f(x, -.55);**  **glVertex2f(y, -.55);**  **glEnd();**  **}**  **void sky(){**  **glBegin(GL\_POLYGON);**  **glColor3ub(18,63,119);**  **glVertex2f(-1, -.5);**  **glVertex2f(1, -.5);**  **glVertex2f(1, 1);**  **glVertex2f(-1, 1);**  **glEnd();**  **}**  **void tree(float x) {**  **// Brown trunk**  **glBegin(GL\_POLYGON);**  **glColor3ub(139, 69, 19);**  **glVertex2f(x + 0.57, -0.5);**  **glVertex2f(x + 0.59, -0.5);**  **glVertex2f(x + 0.59, -0.3);**  **glVertex2f(x + 0.57, -0.3);**  **glEnd();**  **// Foliage**  **glColor3ub(0, 128, 0); // Dark green color**  **circle(0.15, x + 0.58, -0.2, 0, 128, 0);**  **glColor3ub(34, 139, 34); // Lighter green color**  **circle(0.15, x + 0.52, -0.25, 34, 139, 34);**  **circle(0.15, x + 0.64, -0.25, 34, 139, 34);**  **circle(0.15, x + 0.58, -0.25, 34, 139, 34);**  **glColor3ub(0, 128, 0); // Dark green color**  **circle(0.15, x + 0.58, -0.15, 0, 128, 0);**  **}**  **void bench (){**  **glColor3ub(1,1,1);**  **glBegin(GL\_POLYGON);**  **glVertex2f(-.15, -.5);**  **glVertex2f(-.15, -.47);**  **glVertex2f(-.18, -.47);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(1,1,1);**  **glVertex2f(-.27, -.47);**  **glVertex2f(-.3, -.47);**  **glVertex2f(-.3, -.5);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(39,39,39);**  **glVertex2f(-.165, -.475);**  **glVertex2f(-.173, -.445);**  **glVertex2f(-.277, -.445);**  **glVertex2f(-.285, -.475);**  **glEnd();**  **}**  **void lampost(){**  **glBegin(GL\_POLYGON);**  **glColor3ub(1,1,1);**  **glVertex2f(-.425, -.25);**  **glVertex2f(-.425, -.275);**  **glVertex2f(-.25, -.275);**  **glVertex2f(-.25, -.25);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glVertex2f(-.35, -.275);**  **glVertex2f(-.35, -.5);**  **glVertex2f(-.325, -.5);**  **glVertex2f(-.325, -.275);**  **glEnd();**  **circle(.025,-.425,-.275,245,219,55);**  **circle(.025,-.25,-.275,245,219,55);**  **}**  **void trafficSignal(float x) {**  **// Pole**  **glColor3ub(139, 69, 19); // Brown color**  **glBegin(GL\_QUADS);**  **glVertex2f(x - 0.36, -0.5);**  **glVertex2f(x - 0.34, -0.5);**  **glVertex2f(x - 0.34, -0.25); // Halved the height**  **glVertex2f(x - 0.36, -0.25); // Halved the height**  **glEnd();**  **// Signal Box**  **glBegin(GL\_QUADS);**  **glColor3ub(0, 0, 0); // Black color**  **glVertex2f(x - 0.375, -0.4); // Halved the height**  **glVertex2f(x - 0.325, -0.4); // Halved the height**  **glVertex2f(x - 0.325, -0.23);**  **glVertex2f(x - 0.375, -0.23);**  **glEnd();**  **// Red Light**  **circle(0.02, x - 0.35, -0.265, 255, 0, 0);**  **// Yellow Light**  **circle(0.02, x - 0.35, -0.32, 255, 255, 0);**  **// Green Light**  **circle(0.02, x - 0.35, -0.375, 0, 128, 0);**  **}**  **void drawStars(int numStars) {**  **srand(time(0));**  **for (int i = 0; i < numStars; ++i) {**  **float x = (rand() % 200 - 100) / 100.0;**  **float y = (rand() % 150) / 100.0;**  **float radius = (rand() % 3 + 1) / 200.0; // Reduced size**  **circle(radius, x, y, 255, 255, 255);**  **}**  **}**  **void drawWind() {**  **glBegin(GL\_LINES);**  **glColor3ub(251,239,203);**  **glVertex2f(-0.249f, 0.421f);**  **glVertex2f(-0.404f, 0.583f);**  **glVertex2f(-0.464f, 0.476f);**  **glVertex2f(-0.460f, 0.492f);**  **glVertex2f(-0.582f, 0.215f);**  **glVertex2f(-0.380f, -0.081f);**  **glVertex2f(-0.311f, 0.144f);**  **glVertex2f(-0.198f, -0.049f);**  **glVertex2f(-0.089f, 0.152f);**  **glVertex2f(0.029f, -0.136f);**  **glVertex2f(0.209f, 0.219f);**  **glVertex2f(0.087f, 0.591f);**  **glVertex2f(-0.147f, 0.453f);**  **glVertex2f(-0.191f, 0.543f);**  **glEnd();**  **}**  **void ground() {**  **glBegin(GL\_POLYGON);**  **// Brown color at the bottom-left vertex**  **glColor3ub(139, 69, 19);**  **glVertex2f(-1, -1);**  **// Dark brown color at the bottom-right vertex**  **glColor3ub(72, 48, 28);**  **glVertex2f(1, -1);**  **// Green color at the top-right vertex**  **glColor3ub(72, 98, 48);**  **glVertex2f(1, -0.6);**  **// Dark green color at the top-left vertex**  **glColor3ub(34, 139, 34);**  **glVertex2f(-1, -0.6);**  **glEnd();**  **}**  **void drawCar(float x, float y, float size, int r, int g, int b) {**  **// Car body**  **glBegin(GL\_QUADS);**  **glColor3ub(r, g, b); // Set the color passed as a parameter**  **glVertex2f(x, y);**  **glVertex2f(x + size, y);**  **glVertex2f(x + size, y + size \* 0.3);**  **glVertex2f(x, y + size \* 0.3);**  **glEnd();**  **// Car roof**  **glBegin(GL\_QUADS);**  **glColor3ub(r + 30, g + 30, b + 30); // Lighten the color for the roof (adjust as needed)**  **glVertex2f(x + size \* 0.1, y + size \* 0.3);**  **glVertex2f(x + size \* 0.9, y + size \* 0.3);**  **glVertex2f(x + size \* 0.8, y + size \* 0.7);**  **glVertex2f(x + size \* 0.2, y + size \* 0.7);**  **glEnd();**  **// Left window**  **glBegin(GL\_QUADS);**  **glColor3ub(173, 216, 230); // Light blue color (window)**  **glVertex2f(x + size \* 0.25, y + size \* 0.4);**  **glVertex2f(x + size \* 0.45, y + size \* 0.4);**  **glVertex2f(x + size \* 0.45, y + size \* 0.6);**  **glVertex2f(x + size \* 0.25, y + size \* 0.6);**  **glEnd();**  **// Right window**  **glBegin(GL\_QUADS);**  **glColor3ub(173, 216, 230); // Light blue color (window)**  **glVertex2f(x + size \* 0.55, y + size \* 0.4);**  **glVertex2f(x + size \* 0.75, y + size \* 0.4);**  **glVertex2f(x + size \* 0.75, y + size \* 0.6);**  **glVertex2f(x + size \* 0.55, y + size \* 0.6);**  **glEnd();**  **// Left wheel**  **circle(size \* 0.1, x + size \* 0.2, y, 0, 0, 0);**  **// Right wheel**  **circle(size \* 0.1, x + size \* 0.8, y, 0, 0, 0);**  **}**  **void display() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer with current clearing color**  **sky();**  **drawStars(90);**  **tree(0);**  **tree(.4);**  **tree(-1.5);**  **//bench();**  **lampost();**  **trafficSignal(-.3);**  **ground();**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3ub(226,82,82); // Yellow**  **glVertex2f(-.1, -.0);**  **glVertex2f(-.1, -.5);**  **glVertex2f(.3, -.5);**  **glVertex2f(.3, -.0);**  **glEnd();**  **////////**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3ub(15, 105, 134); // Yellow**  **glVertex2f(-.1, -.0);**  **glVertex2f(-.1, -.1);**  **glVertex2f(.3, -.1);**  **glEnd();**  **////////**  **////////**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3ub(0,155,200); // Yellow**  **glVertex2f(-.1, -.1);**  **glVertex2f(-.1, -.2);**  **glVertex2f(.3, -.1);**  **glEnd();**  **////////////////**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3ub(15, 105, 134); // Yellow**  **glVertex2f(-.1, -.2);**  **glVertex2f(-.1, -.3);**  **glVertex2f(.3, -.3);**  **glEnd();**  **////////**  **////////**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3ub(0,155,200); // Yellow**  **glVertex2f(-.1, -.3);**  **glVertex2f(-.1, -.4);**  **glVertex2f(.3, -.3);**  **glEnd();**  **////////**  **glBegin(GL\_POLYGON); // These vertices form a closed polygon**  **glColor3ub(45, 53, 66); // Yellow**  **glVertex2f(-.0, -.4);**  **glVertex2f(-.0, -.5);**  **glVertex2f(.2, -.5);**  **glVertex2f(.2, -.4);**  **glEnd();**  **////////**  **glLineWidth(1.5);**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3ub(45, 53, 66); // Yellow**  **glVertex2f(-.1, -.1);**  **glVertex2f(.3, -.1);**  **glEnd();**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3ub(45, 53, 66); // Yellow**  **glVertex2f(-.1, -.2);**  **glVertex2f(.3, -.2);**  **glEnd();**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3ub(45, 53, 66); // Yellow**  **glVertex2f(-.1, -.3);**  **glVertex2f(.3, -.3);**  **glEnd();**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3ub(45, 53, 66); // Yellow**  **glVertex2f(-.1, -.4);**  **glVertex2f(.3, -.4);**  **glEnd();**  **////**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3ub(45, 53, 66); // Yellow**  **glVertex2f(-1, -.5);**  **glVertex2f(1, -.5);**  **glEnd();**  **glBegin(GL\_LINES); // Each set of 4 vertices form a quad**  **glColor3ub(45, 53, 66); // Yellow**  **glVertex2f(-1, -.6);**  **glVertex2f(1, -.6);**  **glEnd();**  **roadDev(-.6,-.9);**  **roadDev(-.3,0);**  **roadDev(.3,.6);**  **roadDev(1,.9);**  **////**  **drawCar(-.3,-.53,.15,255,0,0);**  **drawCar(-.63,-.53, 0.15, 0, 128, 0); // Green car**  **drawCar(.7,-.53,.15,0, 0, 255);**  **////**  **glFlush(); // Render now**  **}**  **/\* Main function: GLUT runs as a console application starting at main() \*/**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(900, 900); // Set the window's initial width & height**  **glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title**  **glutInitWindowPosition(50, 50); // Position the window's initial top-left corner**  **glutDisplayFunc(display); // Register callback handler for window re-paint event**  **initGL(); // Our own OpenGL initialization**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Draw two village scenarios for day and night |
| **Graph Plot (Picture)-**  **(Not Needed)** |
| **Code:**  #include<GL/gl.h>  #include<iostream>  #include <GL/glut.h>  int posx=0,posy=0;  int day = 1; //1 for day ,0 for night  ///tree  void tree()  {  ///tree1  glColor3f(0.6156863,0,0);  glBegin(GL\_POLYGON);  glVertex3i(50, 350, 0);  glVertex3i(70, 350, 0);  glVertex3i(70, 500, 0);  glVertex3i(50, 500, 0);  glEnd();  glColor3f(0.0, 0.5, 0.0);  glBegin(GL\_POLYGON);  glVertex3i(10, 500, 0);  glVertex3i(110, 500, 0);  glVertex3i(60, 600, 0);  glEnd();  glBegin(GL\_POLYGON);  glVertex3i(15, 550, 0);  glVertex3i(105, 550, 0);  glVertex3i(60, 650, 0);  glEnd();  ///tree2  glColor3f(0.6156863,0,0);  glBegin(GL\_POLYGON);  glVertex3i(50+100, 350, 0);  glVertex3i(70+100, 350, 0);  glVertex3i(70+100, 500, 0);  glVertex3i(50+100, 500, 0);  glEnd();  glColor3f(0.0, 0.5, 0.0);  glBegin(GL\_POLYGON);  glVertex3i(10+100, 500, 0);  glVertex3i(110+100, 500, 0);  glVertex3i(60+100, 600, 0);  glEnd();  glBegin(GL\_POLYGON);  glVertex3i(15+100, 550, 0);  glVertex3i(105+100, 550, 0);  glVertex3i(60+100, 650, 0);  glEnd();  ///tree3  glColor3f(0.6156863,0,0);  glBegin(GL\_POLYGON);  glVertex3i(50+1000, 350+100, 0);  glVertex3i(70+1000, 350+100, 0);  glVertex3i(70+1000, 500+100, 0);  glVertex3i(50+1000, 500+100, 0);  glEnd();  glColor3f(0.0, 0.5, 0.0);  glBegin(GL\_POLYGON);  glVertex3i(10+1000, 500+100, 0);  glVertex3i(110+1000, 500+100, 0);  glVertex3i(60+1000, 600+100, 0);  glEnd();  glBegin(GL\_POLYGON);  glVertex3i(15+1000, 550+100, 0);  glVertex3i(105+1000, 550+100, 0);  glVertex3i(60+1000, 650+100, 0);  glEnd();  ///tree4  glColor3f(0.6156863,0,0);  glBegin(GL\_POLYGON);  glVertex3i(50+1100, 350+200, 0);  glVertex3i(70+1100, 350+200, 0);  glVertex3i(70+1100, 500+200, 0);  glVertex3i(50+1100, 500+200, 0);  glEnd();  glColor3f(0.0, 0.5, 0.0);  glBegin(GL\_POLYGON);  glVertex3i(10+1100, 500+200, 0);  glVertex3i(110+1100, 500+200, 0);  glVertex3i(60+1100, 600+200, 0);  glEnd();  glBegin(GL\_POLYGON);  glVertex3i(15+1100, 550+200, 0);  glVertex3i(105+1100, 550+200, 0);  glVertex3i(60+1100, 650+200, 0);  glEnd();  ///tree5  glColor3f(0.6156863,0,0);  glBegin(GL\_POLYGON);  glVertex3i(50+900, 350+200, 0);  glVertex3i(70+900, 350+200, 0);  glVertex3i(70+900, 500+200, 0);  glVertex3i(50+900, 500+200, 0);  glEnd();  glColor3f(0.0, 0.5, 0.0);  glBegin(GL\_POLYGON);  glVertex3i(10+900, 500+200, 0);  glVertex3i(110+900, 500+200, 0);  glVertex3i(60+900, 600+200, 0);  glEnd();  glBegin(GL\_POLYGON);  glVertex3i(15+900, 550+200, 0);  glVertex3i(105+900, 550+200, 0);  glVertex3i(60+900, 650+200, 0);  glEnd();  }  ///river  void river()  {  glColor3f(0.0352941176470588, 0.5098039215686275, 0.9568627450980392);  glBegin(GL\_POLYGON);  glVertex3i(0, 0, 0);  glVertex3i(1200, 0, 0);  glVertex3i(1200, 300, 0);  glVertex3i(0, 300, 0);  glEnd();  }  //land  void land()  {  glColor3f(0.5 ,1.0 ,0.5);  glBegin(GL\_POLYGON);  glVertex3i(0, 300, 0);  glVertex3i(1200, 300, 0);  glVertex3i(1200, 600, 0);  glVertex3i(0, 600, 0);  glEnd();  }  ///hill  void hill()  {  glColor3f(0.50196, 0.25098, 0.0);  glBegin(GL\_POLYGON);  glVertex3i(600, 600, 0);  glVertex3i(800, 900, 0);  glVertex3i(900, 650, 0);  //glVertex3i(600, 600, 0);  glColor3f(0.50196, 0.25098, 0.0);  glBegin(GL\_POLYGON);  //glVertex3i(600, 600, 0);  glVertex3i(900, 650, 0);  glVertex3i(1000, 800, 0);  glVertex3i(1100, 620, 0);  glColor3f(0.50196, 0.25098, 0.0);  glBegin(GL\_POLYGON);  glVertex3i(1050, 620, 0);  glVertex3i(1200, 800, 0);  glVertex3i(1200, 600, 0);  //glVertex3i(1100, 620, 0);  glEnd();  }  ///house  void house()  {  ///house1  glColor3f(0.5411764705882353, 0.0, 0.5411764705882353);  glBegin(GL\_POLYGON);  glVertex3i(220, 350, 0);  glVertex3i(380, 350, 0);  glVertex3i(380, 450, 0);  glVertex3i(220, 450, 0);  glEnd();  glColor3f(0, 0, 1);  glBegin(GL\_POLYGON);  glVertex3i(200, 450, 0);  glVertex3i(400, 450, 0);  glVertex3i(300, 650, 0);  glEnd();  glColor3f(1,1,0);  glBegin(GL\_POLYGON);  glVertex3i(280, 350, 0);  glVertex3i(320, 350, 0);  glVertex3i(320, 450, 0);  glVertex3i(280, 450, 0);  glEnd();  ///house2  glColor3f(0.5411764705882353, 0.0, 0.5411764705882353);  glBegin(GL\_POLYGON);  glVertex3i(220+200, 350+50, 0);  glVertex3i(380+200, 350+50, 0);  glVertex3i(380+200, 450+50, 0);  glVertex3i(220+200, 450+50, 0);  glEnd();  glColor3f(1.444, .5, 0);  glBegin(GL\_POLYGON);  glVertex3i(200+200, 450+50, 0);  glVertex3i(400+200, 450+50, 0);  glVertex3i(300+200, 650+50, 0);  glEnd();  glColor3f(1,1,0);  glBegin(GL\_POLYGON);  glVertex3i(280+200, 350+50, 0);  glVertex3i(320+200, 350+50, 0);  glVertex3i(320+200, 450+50, 0);  glVertex3i(280+200, 450+50, 0);  glEnd();  }  int rad,x,y,a,b,h;  bool SE,E;  void sun()  {  glColor3f(0.9764705882352941, 0.4705882352941176, 0.0196078431372549);  a=1000;  b=1000;  x=0;  rad=50;  y=rad;  h=1-rad;  glBegin(GL\_POLYGON);  glVertex3i(x+b,y+b,0);  if(h>=0)  {  SE=true;  E=false;  }  else  {  E=true;  SE=false;  }  while(y>=x)  {  if(SE)  {  h=h+5+2\*(x-y);  if(h>=0)  {  SE=true;  E=false;  }  else  {  E=true;  SE=false;  }  x++;  y--;  glVertex3f(x+a, y+b ,0.0);  glVertex3f(y+a, x+b, 0.0);  glVertex3f(-y+a, x+b, 0.0);  glVertex3f(-x+a, -y+b, 0.0);  glVertex3f(-y+a, -x+b, 0.0);  glVertex3f(y+a,-x+b , 0.0);  glVertex3f(x+a, -y+b, 0.0);  glVertex3f(x+a, y+b, 0.0);  glVertex3f(-x+a, y+b, 0.0);  }  if(E)  {  h=h+3+2\*x;  if(h>=0)  {  SE=true;  E=false;  }  else  {  E=true;  SE=false;  }  x++;  glVertex3f(x+a, y+b,0.0);  glVertex3i(y+a, x+b, 0.0);  glVertex3i(-y+a, x+b, 0.0);  glVertex3i(-x+a, -y+b, 0.0);  glVertex3i(-y+a, -x+b, 0.0);  glVertex3i(y+a,-x+b , 0.0);  glVertex3i(x+a, -y+b, 0.0);  glVertex3i(x+a, y+b, 0.0);  glVertex3i(-x+a, y+b, 0.0);  }  }  glEnd();  glColor3f(0.9764705882352941, 0.4705882352941176, 0.0196078431372549);  glBegin(GL\_LINES);  glVertex3i(1000,1000,0);  glVertex3i(1000,950,0);  glEnd();  }  ///boat  void boat()  {  ///boat1  glColor3f(0,0,0);  glBegin(GL\_POLYGON);  glVertex3i(10,150,0);  glVertex3i(150+50,150,0);  glVertex3i(120+50,100,0);  glVertex3i(30,100,0);  glEnd();  glColor3f(1,0,0);  glBegin(GL\_POLYGON);  glVertex3i(50,150,0);  glVertex3i(150,150,0);  glVertex3i(150,220,0);  glVertex3i(50,220,0);  glEnd();  ///boat2  glColor3f(0,0,0);  glBegin(GL\_POLYGON);  glVertex3i(10+200,150+50,0);  glVertex3i(150+50+200,150+50,0);  glVertex3i(120+50+200,100+50,0);  glVertex3i(30+200,100+50,0);  glEnd();  glColor3f(1,0,0);  glBegin(GL\_POLYGON);  glVertex3i(50+200,150+50,0);  glVertex3i(150+200,150+50,0);  glVertex3i(150+200,220+50,0);  glVertex3i(50+200,220+50,0);  glEnd();  }  //night  void night\_effect()  {  glColor3f(0 ,0 ,0);  glBegin(GL\_POLYGON);  glVertex3i(0, 300, 0);  glVertex3i(1200, 300, 0);  glVertex3i(1200, 1200, 0);  glVertex3i(0, 1200, 0);  glEnd();  }  //star  void star()  {  //star1  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(30, 1020, 0);  glVertex3i(50, 1050, 0);  glVertex3i(50, 1100, 0);  glEnd();  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(70, 1020, 0);  glVertex3i(50, 1100, 0);  glVertex3i(50, 1050, 0);  glEnd();  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(50, 1050, 0);  glVertex3i(80, 1070, 0);  glVertex3i(20, 1070, 0);  glEnd();  //star2  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(30+60+60, 1020+60, 0);  glVertex3i(50+60+60, 1050+60, 0);  glVertex3i(50+60+60, 1100+60, 0);  glEnd();  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(70+60+60, 1020+60, 0);  glVertex3i(50+60+60, 1100+60, 0);  glVertex3i(50+60+60, 1050+60, 0);  glEnd();  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(50+60+60, 1050+60, 0);  glVertex3i(80+60+60, 1070+60, 0);  glVertex3i(20+60+60, 1070+60, 0);  glEnd();  //star3  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(30+120+120, 1020, 0);  glVertex3i(50+120+120, 1050, 0);  glVertex3i(50+120+120, 1100, 0);  glEnd();  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(70+120+120, 1020, 0);  glVertex3i(50+120+120, 1100, 0);  glVertex3i(50+120+120, 1050, 0);  glEnd();  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(50+120+120, 1050, 0);  glVertex3i(80+120+120, 1070, 0);  glVertex3i(20+120+120, 1070, 0);  glEnd();  //star4  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(30+120+120+120, 1020+60, 0);  glVertex3i(50+120+120+120, 1050+60, 0);  glVertex3i(50+120+120+120, 1100+60, 0);  glEnd();  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(70+120+120+120, 1020+60, 0);  glVertex3i(50+120+120+120, 1100+60, 0);  glVertex3i(50+120+120+120, 1050+60, 0);  glEnd();  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex3i(50+120+120+120, 1050+60, 0);  glVertex3i(80+120+120+120, 1070+60, 0);  glVertex3i(20+120+120+120, 1070+60, 0);  glEnd();  }  //mon  void moon()  {  glColor3f(1,1,1);  a=600;  b=1000;  x=0;  rad=50;  y=rad;  h=1-rad;  glVertex3i(x+a,y+b,0);  glBegin(GL\_POLYGON);  glVertex3i(x+a,y+b,0);  if(h>=0)  {  SE=true;  E=false;  }  else  {  E=true;  SE=false;  }  while(y>=x)  {  if(SE)  {  h=h+5+2\*(x-y);  if(h>=0)  {  SE=true;  E=false;  }  else  {  E=true;  SE=false;  }  x++;  y--;  glVertex3f(x+a, y+b ,0.0);  glVertex3f(y+a, x+b, 0.0);  glVertex3f(-y+a, x+b, 0.0);  glVertex3f(-x+a, -y+b, 0.0);  glVertex3f(-y+a, -x+b, 0.0);  glVertex3f(y+a,-x+b , 0.0);  glVertex3f(x+a, -y+b, 0.0);  glVertex3f(x+a, y+b, 0.0);  glVertex3f(-x+a, y+b, 0.0);  }  if(E)  {  h=h+3+2\*x;  if(h>=0)  {  SE=true;  E=false;  }  else  {  E=true;  SE=false;  }  x++;  glVertex3f(x+a, y+b,0.0);  glVertex3i(y+a, x+b, 0.0);  glVertex3i(-y+a, x+b, 0.0);  glVertex3i(-x+a, -y+b, 0.0);  glVertex3i(-y+a, -x+b, 0.0);  glVertex3i(y+a,-x+b , 0.0);  glVertex3i(x+a, -y+b, 0.0);  glVertex3i(x+a, y+b, 0.0);  glVertex3i(-x+a, y+b, 0.0);  }  }  glEnd();  glColor3f(1, 1, 1);  glBegin(GL\_LINES);  glVertex3i(601,980,0);  glVertex3i(601,950,0);  glVertex3i(600,980,0);  glVertex3i(600,950,0);  glEnd();  }  ///clouds  void draw\_object()  {  if(day==1)  {  land();  hill();  tree();  river();  house();  sun();  glPushMatrix();  glTranslatef(posx,0,0);  boat();  glPopMatrix();  }  else  {  night\_effect();  land();  star();  hill();  tree();  river();  house();  //sun();  moon();  glPushMatrix();  glTranslatef(posx,0,0);  boat();  glPopMatrix();  }  }  void display(void)  {  glClear(GL\_COLOR\_BUFFER\_BIT);  draw\_object();  glFlush();  }  void keyboardFunc(unsigned char key, int x, int y)  {  switch(key)  {  case 'n':  day=0;  display();  break;  case 'd':  day=1;  display();  break;  case 'N':  day=0;  display();  break;  case 'D':  day=1;  display();  break;  }  }  void update(int value) {  // Keep posx constant to stop the boat's movement  posx = 0;  glutPostRedisplay();  glutTimerFunc(10, update, 0);  }  void init(void)  {  glClearColor(0.6196078431372549,0.9333333333333333, 0.996078431372549, 1.0);  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(0.0, 1200.0, 0, 1200.0);  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(1000, 600);  glutInitWindowPosition(0, 0);  glutCreateWindow("Village Scenario");  glutDisplayFunc(display);  glutKeyboardFunc(keyboardFunc);  init();  glutTimerFunc(25, update, 0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |