**Lab Practice-7**

|  |
| --- |
| **Question-**  Create a simple day and night scenario that will automatically change from day to night |
| **Graph**    **DAY Scenario**    **NIGHT Scenario** |
| **Code**  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  // 22-46013-1  // MD. SHOHANUR RAHMAN SHOHAN  bool isDay = true;  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++)  {  glColor3f(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 displayDay()  {  glBegin(GL\_POLYGON); //sky  glColor3f(0.5607,0.7412,0.8902);  glVertex2f(-25.0f, 15.0f);  glVertex2f(10.0, 15.0f);  glVertex2f(10.0f, 0.0f);  glVertex2f(-25.0f, 0.0f);  glEnd();  circle(1.7592, 4.0, 12.0, 1.0,0.7,0.2);//sun  // ################################  // ## ##  // ## J U N G L E ##  // ## ##  // ################################  circle(1.0, -24.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -22.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -20.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -18.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -16.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -14.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -12.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -10.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 0.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 2.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 4.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 6.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 8.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 10.0, 0.0, 0.01,0.25,0.13);  //Big  circle(1.51329, 7.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, 3.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, -1.5, 1.0, 0.01,0.25,0.13);  circle(1.51329, -20.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, -23.5, 1.0, 0.01,0.25,0.13);  // ################################  // ## ##  // ## G R A S S ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.2f, 0.4, 0.24f);  glVertex2f(-25.0f, -8.0f);  glVertex2f(10.0, -8.0f);  glVertex2f(10.0f, 0.0f);  glVertex2f(-25.0f, 0.0f);  glEnd();  // ################################  // ## ##  // ## T R E E ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.4f, 0.18f, 0.0);  glVertex2f(-14.0f, -3.0f);  glVertex2f(-12.0, -3.0f);  glVertex2f(-12.0f, 3.0f);  glVertex2f(-14.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.4f, 0.18f, 0.0);  glVertex2f(-14.0f, -3.0f);  glVertex2f(-15.0, -4.0f);  glVertex2f(-13.6f, -3.3f);  glVertex2f(-13.0f, -4.0f);  glVertex2f(-12.4f, -3.3f);  glVertex2f(-11.0f, -4.0f);  glVertex2f(-12.0f, -3.0f);  glEnd();  circle(3.3801, -16.0, 6.0, 0.0,0.5,0.0);  circle(4.1227, -13.0, 7.5, 0.0,0.5,0.0);  circle(3.3844, -10.0, 6.0, 0.0,0.5,0.0);  circle(3.5737, -12.0, 4.0, 0.0,0.5,0.0);  circle(3.5737, -14.0, 4.0, 0.0,0.5,0.0);  circle(1.4889, -13.0, 1.0, 0.0,0.5,0.0);  // ################################  // ## ##  // ## S T R A W ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.7f, 0.7, 0.0f);  glVertex2f(3.0f, -3.0f);  glVertex2f(0.0, -3.0f);  glVertex2f(0.0f, -1.0f);  glVertex2f(0.4f, 0.4f);  glVertex2f(1.4f, 1.0f);  glVertex2f(1.6f, 1.0f);  glVertex2f(2.6f, 0.4f);  glVertex2f(3.0f, -1.0f);  glEnd();  // ################################  // ## ##  // ## H O U S E ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, -3.0f);  glVertex2f(-1.0f, -3.0f);  glVertex2f(-1.0f, 1.0f);  glVertex2f(-6.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, -3.0f);  glVertex2f(-8.0f, -2.5f);  glVertex2f(-8.0f, 1.0f);  glVertex2f(-6.0f, 1.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, 1.0f);  glVertex2f(-8.2f, 1.0f);  glVertex2f(-7.2f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.25f, 0.25f, 0.25f);  glVertex2f(-8.0f, 1.0f);  glVertex2f(-8.2f, 1.0f);  glVertex2f(-7.2f, 3.0f);  glVertex2f(-7.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.35f, 0.35f, 0.35f);  glVertex2f(-6.0f, 1.0f);  glVertex2f(-1.0f, 1.0f);  glVertex2f(-2.0f, 3.0f);  glVertex2f(-7.2f, 3.0f);  glEnd();  //DOOR  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-3.0f, -3.0f);  glVertex2f(-4.0f, -3.0f);  glVertex2f(-4.0f, -1.0f);  glVertex2f(-3.0f, -1.0f);  glEnd();  //Window  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-2.5f, -1.0f);  glVertex2f(-1.5f, -1.0f);  glVertex2f(-1.5f, 0.0f);  glVertex2f(-2.5f, 0.0f);  glEnd();  //Window  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-5.5f, -1.0f);  glVertex2f(-5.5f, 0.0f);  glVertex2f(-4.5f, 0.0f);  glVertex2f(-4.5f, -1.0f);  glEnd();  //DOOR  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-6.8f, -2.8f);  glVertex2f(-6.8f, -1.1f);  glVertex2f(-7.2f, -1.0f);  glVertex2f(-7.2f, -2.7f);  glEnd();  //glFlush(); // Render now  }  void displayNight()  {  glBegin(GL\_POLYGON); //sky  glColor3f(0.0f,0.0,0.17f);  glVertex2f(-25.0f, 15.0f);  glVertex2f(10.0, 15.0f);  glVertex2f(10.0f, 0.0f);  glVertex2f(-25.0f, 0.0f);  glEnd();  circle(1.7592, 4.0, 12.0, 1.0,1.0,1.0);//sun  // STAR  circle(0.0981, 1.9, 8.9, 1.0,1.0,1.0);  circle(0.0981, -1.9, 8.3, 1.0,1.0,1.0);  circle(0.0981, -6.1, 8.5, 1.0,1.0,1.0);  circle(0.0981, -9.1, 10.5, 1.0,1.0,1.0);  circle(0.0981, -9.3, 12.1, 1.0,1.0,1.0);  circle(0.0981, -10.9, 13.1, 1.0,1.0,1.0);  circle(0.0981, -3.88, 13.51, 1.0,1.0,1.0);  circle(0.0981, -1.4, 14.3, 1.0,1.0,1.0);  // ################################  // ## ##  // ## J U N G L E ##  // ## ##  // ################################  circle(1.0, -24.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -22.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -20.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -18.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -16.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -14.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -12.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -10.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 0.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 2.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 4.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 6.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 8.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 10.0, 0.0, 0.01,0.25,0.13);  //Big  circle(1.51329, 7.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, 3.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, -1.5, 1.0, 0.01,0.25,0.13);  circle(1.51329, -20.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, -23.5, 1.0, 0.01,0.25,0.13);  // ################################  // ## ##  // ## G R A S S ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.2f, 0.4, 0.24f);  glVertex2f(-25.0f, -8.0f);  glVertex2f(10.0, -8.0f);  glVertex2f(10.0f, 0.0f);  glVertex2f(-25.0f, 0.0f);  glEnd();  // ################################  // ## ##  // ## T R E E ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.4f, 0.18f, 0.0);  glVertex2f(-14.0f, -3.0f);  glVertex2f(-12.0, -3.0f);  glVertex2f(-12.0f, 3.0f);  glVertex2f(-14.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.4f, 0.18f, 0.0);  glVertex2f(-14.0f, -3.0f);  glVertex2f(-15.0, -4.0f);  glVertex2f(-13.6f, -3.3f);  glVertex2f(-13.0f, -4.0f);  glVertex2f(-12.4f, -3.3f);  glVertex2f(-11.0f, -4.0f);  glVertex2f(-12.0f, -3.0f);  glEnd();  circle(3.3801, -16.0, 6.0, 0.0,0.5,0.0);  circle(4.1227, -13.0, 7.5, 0.0,0.5,0.0);  circle(3.3844, -10.0, 6.0, 0.0,0.5,0.0);  circle(3.5737, -12.0, 4.0, 0.0,0.5,0.0);  circle(3.5737, -14.0, 4.0, 0.0,0.5,0.0);  circle(1.4889, -13.0, 1.0, 0.0,0.5,0.0);  // ################################  // ## ##  // ## S T R A W ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.7f, 0.7, 0.0f);  glVertex2f(3.0f, -3.0f);  glVertex2f(0.0, -3.0f);  glVertex2f(0.0f, -1.0f);  glVertex2f(0.4f, 0.4f);  glVertex2f(1.4f, 1.0f);  glVertex2f(1.6f, 1.0f);  glVertex2f(2.6f, 0.4f);  glVertex2f(3.0f, -1.0f);  glEnd();  // ################################  // ## ##  // ## H O U S E ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, -3.0f);  glVertex2f(-1.0f, -3.0f);  glVertex2f(-1.0f, 1.0f);  glVertex2f(-6.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, -3.0f);  glVertex2f(-8.0f, -2.5f);  glVertex2f(-8.0f, 1.0f);  glVertex2f(-6.0f, 1.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, 1.0f);  glVertex2f(-8.2f, 1.0f);  glVertex2f(-7.2f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.25f, 0.25f, 0.25f);  glVertex2f(-8.0f, 1.0f);  glVertex2f(-8.2f, 1.0f);  glVertex2f(-7.2f, 3.0f);  glVertex2f(-7.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.35f, 0.35f, 0.35f);  glVertex2f(-6.0f, 1.0f);  glVertex2f(-1.0f, 1.0f);  glVertex2f(-2.0f, 3.0f);  glVertex2f(-7.2f, 3.0f);  glEnd();  //DOOR  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-3.0f, -3.0f);  glVertex2f(-4.0f, -3.0f);  glVertex2f(-4.0f, -1.0f);  glVertex2f(-3.0f, -1.0f);  glEnd();  //Window  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-2.5f, -1.0f);  glVertex2f(-1.5f, -1.0f);  glVertex2f(-1.5f, 0.0f);  glVertex2f(-2.5f, 0.0f);  glEnd();  //Window  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-5.5f, -1.0f);  glVertex2f(-5.5f, 0.0f);  glVertex2f(-4.5f, 0.0f);  glVertex2f(-4.5f, -1.0f);  glEnd();  //DOOR  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-6.8f, -2.8f);  glVertex2f(-6.8f, -1.1f);  glVertex2f(-7.2f, -1.0f);  glVertex2f(-7.2f, -2.7f);  glEnd();  //glFlush(); // Render now  }  void switchScenario(int value) {  isDay = !isDay;  glutPostRedisplay();  glutTimerFunc(3000, switchScenario, 0);  }  void display() {  glClearColor(0.0f, 0.0f, 0.0f, 0.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  glLineWidth(1);  if (isDay) {  displayDay();  } else {  displayNight();  }  glFlush();  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutInitWindowSize(920, 520);// Set the window's initial width & height  glutCreateWindow("Day and Night Scene");  //gluOrtho2D(-0.1,0.7,-0.1,0.3); // Create a window with the given title  //glutInitWindowSize(320, 320);// Set the window's initial width & height  glutDisplayFunc(display);// Register display callback handler for window re-paint  glutTimerFunc(3000, switchScenario, 0);  gluOrtho2D(-25,10,-8,15);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-**    **DAY**    **NIGHT** |

|  |
| --- |
| **Question-**  Create a simple day and night scenario using keyboard interaction. The key ‘D’ or ‘d’ will initiate the day mode and the key ‘N’ or ‘n’ will initiate the night mode. |
| **Graph**    **DAY**    **NIGHT** |
| **Code**  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  // 22-46013-1  // MD. SHOHANUR RAHMAN SHOHAN  bool isDay = true;  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++)  {  glColor3f(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 displayDay()  {  glBegin(GL\_POLYGON); //sky  glColor3f(0.5607,0.7412,0.8902);  glVertex2f(-25.0f, 15.0f);  glVertex2f(10.0, 15.0f);  glVertex2f(10.0f, 0.0f);  glVertex2f(-25.0f, 0.0f);  glEnd();  circle(1.7592, 4.0, 12.0, 1.0,0.7,0.2);//sun  // ################################  // ## ##  // ## J U N G L E ##  // ## ##  // ################################  circle(1.0, -24.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -22.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -20.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -18.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -16.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -14.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -12.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -10.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 0.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 2.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 4.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 6.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 8.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 10.0, 0.0, 0.01,0.25,0.13);  //Big  circle(1.51329, 7.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, 3.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, -1.5, 1.0, 0.01,0.25,0.13);  circle(1.51329, -20.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, -23.5, 1.0, 0.01,0.25,0.13);  // ################################  // ## ##  // ## G R A S S ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.2f, 0.4, 0.24f);  glVertex2f(-25.0f, -8.0f);  glVertex2f(10.0, -8.0f);  glVertex2f(10.0f, 0.0f);  glVertex2f(-25.0f, 0.0f);  glEnd();  // ################################  // ## ##  // ## T R E E ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.4f, 0.18f, 0.0);  glVertex2f(-14.0f, -3.0f);  glVertex2f(-12.0, -3.0f);  glVertex2f(-12.0f, 3.0f);  glVertex2f(-14.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.4f, 0.18f, 0.0);  glVertex2f(-14.0f, -3.0f);  glVertex2f(-15.0, -4.0f);  glVertex2f(-13.6f, -3.3f);  glVertex2f(-13.0f, -4.0f);  glVertex2f(-12.4f, -3.3f);  glVertex2f(-11.0f, -4.0f);  glVertex2f(-12.0f, -3.0f);  glEnd();  circle(3.3801, -16.0, 6.0, 0.0,0.5,0.0);  circle(4.1227, -13.0, 7.5, 0.0,0.5,0.0);  circle(3.3844, -10.0, 6.0, 0.0,0.5,0.0);  circle(3.5737, -12.0, 4.0, 0.0,0.5,0.0);  circle(3.5737, -14.0, 4.0, 0.0,0.5,0.0);  circle(1.4889, -13.0, 1.0, 0.0,0.5,0.0);  // ################################  // ## ##  // ## S T R A W ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.7f, 0.7, 0.0f);  glVertex2f(3.0f, -3.0f);  glVertex2f(0.0, -3.0f);  glVertex2f(0.0f, -1.0f);  glVertex2f(0.4f, 0.4f);  glVertex2f(1.4f, 1.0f);  glVertex2f(1.6f, 1.0f);  glVertex2f(2.6f, 0.4f);  glVertex2f(3.0f, -1.0f);  glEnd();  // ################################  // ## ##  // ## H O U S E ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, -3.0f);  glVertex2f(-1.0f, -3.0f);  glVertex2f(-1.0f, 1.0f);  glVertex2f(-6.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, -3.0f);  glVertex2f(-8.0f, -2.5f);  glVertex2f(-8.0f, 1.0f);  glVertex2f(-6.0f, 1.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, 1.0f);  glVertex2f(-8.2f, 1.0f);  glVertex2f(-7.2f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.25f, 0.25f, 0.25f);  glVertex2f(-8.0f, 1.0f);  glVertex2f(-8.2f, 1.0f);  glVertex2f(-7.2f, 3.0f);  glVertex2f(-7.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.35f, 0.35f, 0.35f);  glVertex2f(-6.0f, 1.0f);  glVertex2f(-1.0f, 1.0f);  glVertex2f(-2.0f, 3.0f);  glVertex2f(-7.2f, 3.0f);  glEnd();  //DOOR  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-3.0f, -3.0f);  glVertex2f(-4.0f, -3.0f);  glVertex2f(-4.0f, -1.0f);  glVertex2f(-3.0f, -1.0f);  glEnd();  //Window  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-2.5f, -1.0f);  glVertex2f(-1.5f, -1.0f);  glVertex2f(-1.5f, 0.0f);  glVertex2f(-2.5f, 0.0f);  glEnd();  //Window  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-5.5f, -1.0f);  glVertex2f(-5.5f, 0.0f);  glVertex2f(-4.5f, 0.0f);  glVertex2f(-4.5f, -1.0f);  glEnd();  //DOOR  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-6.8f, -2.8f);  glVertex2f(-6.8f, -1.1f);  glVertex2f(-7.2f, -1.0f);  glVertex2f(-7.2f, -2.7f);  glEnd();  //glFlush(); // Render now  }  void displayNight()  {  glBegin(GL\_POLYGON); //sky  glColor3f(0.0f,0.0,0.17f);  glVertex2f(-25.0f, 15.0f);  glVertex2f(10.0, 15.0f);  glVertex2f(10.0f, 0.0f);  glVertex2f(-25.0f, 0.0f);  glEnd();  circle(1.7592, 4.0, 12.0, 1.0,1.0,1.0);//sun  // STAR  circle(0.0981, 1.9, 8.9, 1.0,1.0,1.0);  circle(0.0981, -1.9, 8.3, 1.0,1.0,1.0);  circle(0.0981, -6.1, 8.5, 1.0,1.0,1.0);  circle(0.0981, -9.1, 10.5, 1.0,1.0,1.0);  circle(0.0981, -9.3, 12.1, 1.0,1.0,1.0);  circle(0.0981, -10.9, 13.1, 1.0,1.0,1.0);  circle(0.0981, -3.88, 13.51, 1.0,1.0,1.0);  circle(0.0981, -1.4, 14.3, 1.0,1.0,1.0);  // ################################  // ## ##  // ## J U N G L E ##  // ## ##  // ################################  circle(1.0, -24.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -22.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -20.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -18.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -16.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -14.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -12.0, 0.0, 0.01,0.25,0.13);  circle(1.0, -10.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 0.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 2.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 4.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 6.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 8.0, 0.0, 0.01,0.25,0.13);  circle(1.0, 10.0, 0.0, 0.01,0.25,0.13);  //Big  circle(1.51329, 7.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, 3.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, -1.5, 1.0, 0.01,0.25,0.13);  circle(1.51329, -20.0, 1.0, 0.01,0.25,0.13);  circle(1.51329, -23.5, 1.0, 0.01,0.25,0.13);  // ################################  // ## ##  // ## G R A S S ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.2f, 0.4, 0.24f);  glVertex2f(-25.0f, -8.0f);  glVertex2f(10.0, -8.0f);  glVertex2f(10.0f, 0.0f);  glVertex2f(-25.0f, 0.0f);  glEnd();  // ################################  // ## ##  // ## T R E E ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.4f, 0.18f, 0.0);  glVertex2f(-14.0f, -3.0f);  glVertex2f(-12.0, -3.0f);  glVertex2f(-12.0f, 3.0f);  glVertex2f(-14.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.4f, 0.18f, 0.0);  glVertex2f(-14.0f, -3.0f);  glVertex2f(-15.0, -4.0f);  glVertex2f(-13.6f, -3.3f);  glVertex2f(-13.0f, -4.0f);  glVertex2f(-12.4f, -3.3f);  glVertex2f(-11.0f, -4.0f);  glVertex2f(-12.0f, -3.0f);  glEnd();  circle(3.3801, -16.0, 6.0, 0.0,0.5,0.0);  circle(4.1227, -13.0, 7.5, 0.0,0.5,0.0);  circle(3.3844, -10.0, 6.0, 0.0,0.5,0.0);  circle(3.5737, -12.0, 4.0, 0.0,0.5,0.0);  circle(3.5737, -14.0, 4.0, 0.0,0.5,0.0);  circle(1.4889, -13.0, 1.0, 0.0,0.5,0.0);  // ################################  // ## ##  // ## S T R A W ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.7f, 0.7, 0.0f);  glVertex2f(3.0f, -3.0f);  glVertex2f(0.0, -3.0f);  glVertex2f(0.0f, -1.0f);  glVertex2f(0.4f, 0.4f);  glVertex2f(1.4f, 1.0f);  glVertex2f(1.6f, 1.0f);  glVertex2f(2.6f, 0.4f);  glVertex2f(3.0f, -1.0f);  glEnd();  // ################################  // ## ##  // ## H O U S E ##  // ## ##  // ################################  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, -3.0f);  glVertex2f(-1.0f, -3.0f);  glVertex2f(-1.0f, 1.0f);  glVertex2f(-6.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, -3.0f);  glVertex2f(-8.0f, -2.5f);  glVertex2f(-8.0f, 1.0f);  glVertex2f(-6.0f, 1.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.44f, 0.4, 0.24f);  glVertex2f(-6.0f, 1.0f);  glVertex2f(-8.2f, 1.0f);  glVertex2f(-7.2f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.25f, 0.25f, 0.25f);  glVertex2f(-8.0f, 1.0f);  glVertex2f(-8.2f, 1.0f);  glVertex2f(-7.2f, 3.0f);  glVertex2f(-7.0f, 3.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3f(0.35f, 0.35f, 0.35f);  glVertex2f(-6.0f, 1.0f);  glVertex2f(-1.0f, 1.0f);  glVertex2f(-2.0f, 3.0f);  glVertex2f(-7.2f, 3.0f);  glEnd();  //DOOR  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-3.0f, -3.0f);  glVertex2f(-4.0f, -3.0f);  glVertex2f(-4.0f, -1.0f);  glVertex2f(-3.0f, -1.0f);  glEnd();  //Window  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-2.5f, -1.0f);  glVertex2f(-1.5f, -1.0f);  glVertex2f(-1.5f, 0.0f);  glVertex2f(-2.5f, 0.0f);  glEnd();  //Window  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-5.5f, -1.0f);  glVertex2f(-5.5f, 0.0f);  glVertex2f(-4.5f, 0.0f);  glVertex2f(-4.5f, -1.0f);  glEnd();  //DOOR  glBegin(GL\_POLYGON);  glColor3f(0.5f, 0.35, 0.05f);  glVertex2f(-6.8f, -2.8f);  glVertex2f(-6.8f, -1.1f);  glVertex2f(-7.2f, -1.0f);  glVertex2f(-7.2f, -2.7f);  glEnd();  //glFlush(); // Render now  }  void switchToDay() {  isDay = true;  glutPostRedisplay();  }  void switchToNight() {  isDay = false;  glutPostRedisplay();  }  void handleKeypress(unsigned char key, int x, int y) {  switch (key)  {  case 'D':  case 'd':  switchToDay();  break;  case 'N':  case 'n':  switchToNight();  break;  glutPostRedisplay();  }  }  void display() {  glClearColor(0.0f, 0.0f, 0.0f, 0.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  glLineWidth(1);  if (isDay) {  displayDay();  } else {  displayNight();  }  glFlush();  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutInitWindowSize(920, 520);// Set the window's initial width & height  glutCreateWindow("Day and Night Scene");  //gluOrtho2D(-0.1,0.7,-0.1,0.3); // Create a window with the given title  //glutInitWindowSize(320, 320);// Set the window's initial width & height  glutDisplayFunc(display);// Register display callback handler for window re-paint  glutKeyboardFunc(handleKeypress);  gluOrtho2D(-25,10,-8,15);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-**    **DAY**    **NIGHT** |