**Lab Practice-7**

|  |
| --- |
| **Question-**  Create a simple day and night scenario that will automatically change from day to night |
| **Graph** |
| **Code-**  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  /\* Handler for window-repaint event. Call back when the window first appears and  whenever the window needs to be re-painted. \*/  int state = 1;  void khor\_pala(){  //upper extension of the khor pala  glColor3f(0, 0, 0);  glBegin(GL\_POLYGON);  //right yellow chad  glVertex2f(-433.98356, 57.3442926);  glVertex2f(-434, 80);  glVertex2f(-425, 80);  glVertex2f(-425, 57.3442926);  glEnd();  //yellow khor pala  glColor3f(1, 1, 0);  glBegin(GL\_POLYGON);  glVertex2f(-387.48784, 23.76182488);  glVertex2f(-400, 45);  glVertex2f(-430, 60);  glVertex2f(-468, 34);  glVertex2f(-484, 6);  glVertex2f(-490, -52);  glVertex2f(-486, -78);  glVertex2f(-455, -100);  glVertex2f(-405, -100);  glVertex2f(-380, -80);  glVertex2f(-380, -20);  glEnd();  }  void house\_2(){  //right yellow chad  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-365, 50);  glVertex2f(-320, -20);  glVertex2f(-165, -20);  glVertex2f(-225, 50);  glEnd();  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  //left yellow chad  glVertex2f(-365, 50);  glVertex2f(-425, -20);  glVertex2f(-380, -20);  glVertex2f(-347.9209, 23.43264033);  glEnd();  //under 1 gray color area (door)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-347.9209, 23.43264033);  glVertex2f(-380, -20);  glVertex2f(-380, -80);  glVertex2f(-320, -90);  glVertex2f(-320, -20);  glEnd();  //under 2 gray color area (window)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-320, -20);  glVertex2f(-320, -90);  glVertex2f(-185, -65);  glVertex2f(-185, -20);  glEnd();  //under extension of house 2  // number 1(door)  glColor3f(0.9, 0.8, 0.7);  glBegin(GL\_POLYGON);  glVertex2f(-380, -80);  glVertex2f(-405, -100);  glVertex2f(-320, -115);  glVertex2f(-320, -90);  glEnd();  // number 2(extension)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-320, -90);  glVertex2f(-320, -115);  glVertex2f(-170, -80);  glVertex2f(-185, -65);  glEnd();  //door  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-360, -30);  glVertex2f(-360, -70);  glVertex2f(-340, -70);  glVertex2f(-340, -30);  glEnd();  //window  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-280, -30);  glVertex2f(-280, -60);  glVertex2f(-250, -60);  glVertex2f(-250, -30);  glEnd();  }  void house\_1(){  //black area under traffic light poll  //upper portion  glColor3f(1, 0.7, 0);  glBegin(GL\_POLYGON);  glVertex2f(-455, 50);  glVertex2f(-555, 50);  glVertex2f(-650, -25);  glVertex2f(-490, -25);  glVertex2f(-440, 0);  glEnd();  //lowerportion  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-620.0788546, -25);  glVertex2f(-619.975899, -78);  glVertex2f(-486, -78);  glVertex2f(-486, -25);  glEnd();  //extream lower portion  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex2f(-636, -78);  glVertex2f(-636, -100);  glVertex2f(-455, -100);  glVertex2f(-455, -78);  glEnd();  //door  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-570, -30);  glVertex2f(-570, -70);  glVertex2f(-540, -70);  glVertex2f(-540, -30);  glEnd();  }  void grass(){  glColor3f(0, 1, 0);  glBegin(GL\_POLYGON);  glVertex2f(-1000, -75);  glVertex2f(-1000, -100);  glVertex2f(-710, -140);  glVertex2f(-765, -180);  glVertex2f(-425, -230);  glVertex2f(-445, -250);  glVertex2f(-435, -265);  glVertex2f(-380, -275);  glVertex2f(-315, -280);  glVertex2f(-150, -300);  glVertex2f(50, -300);  glVertex2f(50, -40);  glEnd();  }  void tree(){  glBegin(GL\_POLYGON);  glColor3f(0.0f, 1.0f, 0.0f);  glVertex2f(-140, 30.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-168, 16.5);  glVertex2f(-179, 27);  glVertex2f(-180, 40);  glVertex2f(-189, 44);  glVertex2f(-197.2, 53.4);  glVertex2f(-200.2, 68);  glVertex2f(-197.5,80.5);  glVertex2f(-192.5,87.5);  glVertex2f(-184.3,91.3);  glVertex2f(-182.6,101.1);  glVertex2f(-176,109);  glVertex2f(-166,111.5);  glVertex2f(-155.8,111.85 );  glVertex2f(-150.8,110.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-140.4,114 );  glVertex2f(-129.8,117.6 );  glVertex2f(-119.05,117.35 );  glVertex2f(-111.1,113.7 );  glVertex2f(-104.4,106.2 );  glVertex2f(-100,100 );  glVertex2f(-92,100.6 );  glVertex2f(-82,97.4 );  glVertex2f(-76.1,88.8);  glVertex2f(-76,78.8);  glVertex2f(-78.45, 71.9);  glVertex2f(-73.6, 67.8);  glVertex2f(-70.2, 61.4);  glVertex2f(-71, 51);  glVertex2f(-75.86, 43.68);  glVertex2f(-73.04, 37.78);  glVertex2f(-70, 30);  glVertex2f(-73, 19);  glVertex2f(-85, 8.5);  glVertex2f(-99, 7.2);  glVertex2f(-112, 10);  glVertex2f(-120, 24.5);  glEnd();  glColor3f(0.5f, 0.3f, 0.0f);  glBegin(GL\_POLYGON);  /\*  glVertex2f(-120, 24.5);  glVertex2f(-140, 24.5);  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, -24.5);  \*/  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, 24.5);  glVertex2f(-140, 24.5);  /\*  \*/  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(-120, 24.5);  glVertex2f(-107.7, 35);  glVertex2f(-113.7, 42);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 43);  glVertex2f(-140.2, 50.6);  glVertex2f(-144.7, 42);  glVertex2f(-140, 37);  glVertex2f(-140, 24.5);  glEnd();  }  void outline\_tree(){  glColor3f(0, 0, 0);  glLineWidth(1.5);  glBegin(GL\_LINES);  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, -65);  glVertex2f(-120, 24.5);  glVertex2f(-120, 24.5);  glVertex2f(-107.7, 35);  glVertex2f(-107.7, 35);  glVertex2f(-113.7, 42);  glVertex2f(-113.7, 42);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 61);  glVertex2f(-119.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 43);  glVertex2f(-131.7, 43);  glVertex2f(-140.2, 50.6);  glVertex2f(-140.2, 50.6);  glVertex2f(-144.7, 42);  glVertex2f(-144.7, 42);  glVertex2f(-140, 37);  glVertex2f(-140, 37);  glVertex2f(-140, 24.5);  glVertex2f(-140, 24.5);  glVertex2f(-140, -65);  //round green leafs  glVertex2f(-140, 30.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-168, 16.5);  glVertex2f(-168, 16.5);  glVertex2f(-179, 27);  glVertex2f(-179, 27);  glVertex2f(-180, 40);  glVertex2f(-180, 40);  glVertex2f(-189, 44);  glVertex2f(-189, 44);  glVertex2f(-197.2, 53.4);  glVertex2f(-197.2, 53.4);  glVertex2f(-200.2, 68);  glVertex2f(-200.2, 68);  glVertex2f(-197.5,80.5);  glVertex2f(-197.5,80.5);  glVertex2f(-192.5,87.5);  glVertex2f(-192.5,87.5);  glVertex2f(-184.3,91.3);  glVertex2f(-184.3,91.3);  glVertex2f(-182.6,101.1);  glVertex2f(-182.6,101.1);  glVertex2f(-176,109);  glVertex2f(-176,109);  glVertex2f(-166,111.5);  glVertex2f(-166,111.5);  glVertex2f(-155.8,111.85 );  glVertex2f(-155.8,111.85 );  glVertex2f(-150.8,110.2 );  glVertex2f(-150.8,110.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-140.4,114 );  glVertex2f(-140.4,114 );  glVertex2f(-129.8,117.6 );  glVertex2f(-129.8,117.6 );  glVertex2f(-119.05,117.35 );  glVertex2f(-119.05,117.35 );  glVertex2f(-111.1,113.7 );  glVertex2f(-111.1,113.7 );  glVertex2f(-104.4,106.2 );  glVertex2f(-104.4,106.2 );  glVertex2f(-100,100 );  glVertex2f(-100,100 );  glVertex2f(-92,100.6 );  glVertex2f(-92,100.6 );  glVertex2f(-82,97.4 );  glVertex2f(-82,97.4 );  glVertex2f(-76.1,88.8);  glVertex2f(-76.1,88.8);  glVertex2f(-76,78.8);  glVertex2f(-76,78.8);  glVertex2f(-78.45, 71.9);  glVertex2f(-78.45, 71.9);  glVertex2f(-73.6, 67.8);  glVertex2f(-73.6, 67.8);  glVertex2f(-70.2, 61.4);  glVertex2f(-70.2, 61.4);  glVertex2f(-71, 51);  glVertex2f(-71, 51);  glVertex2f(-75.86, 43.68);  glVertex2f(-75.86, 43.68);  glVertex2f(-73.04, 37.78);  glVertex2f(-73.04, 37.78);  glVertex2f(-70, 30);  glVertex2f(-70, 30);  glVertex2f(-73, 19);  glVertex2f(-73, 19);  glVertex2f(-85, 8.5);  glVertex2f(-85, 8.5);  glVertex2f(-99, 7.2);  glVertex2f(-99, 7.2);  glVertex2f(-112, 10);  glVertex2f(-112, 10);  glVertex2f(-120, 16.5);  //glVertex2f(-120, 24.5);  glEnd();  }  void two\_hills(){  // first hill  glColor3f(0, 0.8, 0);  glBegin(GL\_POLYGON);  glVertex2f(-1000, -45.5);  glVertex2f(-900, 50);  glVertex2f(-883, 61);  glVertex2f(-870, 61);  glVertex2f(-850, 51);  glVertex2f(-812.251844, 26.727090);  glVertex2f(-895, -75);  glVertex2f(-1000, -75);  glEnd();  //second hill  glBegin(GL\_POLYGON);  glColor3f(0, 0.5, 0);  glVertex2f(-895, -75);  //glVertex2f(-812.251844, 26.727090);  glVertex2f(-786, 59);  glVertex2f(-780, 65);  glVertex2f(-773, 65);  glVertex2f(-764, 61);  glVertex2f(-580, -75);  glEnd();  }  void sun(){  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<360;i++)  {  glColor3f(1,0.9,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r= 85.3901037;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x - 824.535315,y + 53.8050655 );  }  //glVertex2f(0.3f,0.4f);  //glVertex2f(0.1f,0.4f);  glEnd();  glLineWidth(5);  glColor3f(0, 0, 0);  glBegin(GL\_LINES);  glVertex2f(-968, 48);  glVertex2f(-926, 56);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-980, 88);  glVertex2f(-918, 70);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-954, 124);  glVertex2f(-922, 104);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-954, 124);  glVertex2f(-922, 104);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-925, 150);  glVertex2f(-896, 124);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-888,158);  glVertex2f(-874,138);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-852, 194);  glVertex2f(-850, 150);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-814, 170);  glVertex2f(-816, 152);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-758, 172);  glVertex2f(-780, 142);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-744, 142);  glVertex2f(-760, 128);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-680, 150);  glVertex2f(-740, 110);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-680, 100);  glVertex2f(-730, 90);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-728, 66);  glVertex2f(-674, 58);  glEnd();  }  void boat(){  //sitting area  glBegin(GL\_POLYGON);  glColor3f(0.5, 0.3, 0);  glVertex2f(-936, -218);  glVertex2f(-824, -218);  glVertex2f(-800.6849, -179.6123);  glVertex2f(-972, -180);  glEnd();  //shade  glBegin(GL\_POLYGON);  glColor3f(1, .9, .6);  glVertex2f(-880, -194);  glVertex2f(-836, -194);  glVertex2f(-817.9758973, -186.61688);  glVertex2f(-826.261432, -169.25207);  glVertex2f(-850, -148);  glVertex2f(-907, -148);  glVertex2f(-894, -157);  glVertex2f(-885.1914705, -180.2796);  glEnd();  //shade left  glBegin(GL\_POLYGON);  glColor3f(1, .9, .6);  glVertex2f(-936.8605, -180.1132086);  glVertex2f(-885.1914705, -180.2796);  glVertex2f(-894, -157);  glVertex2f(-907, -148);  glVertex2f(-924, -156);  glVertex2f(-934, -174);  glEnd();  //stick  glLineWidth(6);  glBegin(GL\_LINES);  glColor3f(0, 0, 0);  glVertex2f(-846, -240);  glVertex2f(-806, -106);  glEnd();  //black lower area  glBegin(GL\_POLYGON);  glColor3f(0, 0, 0);  glVertex2f(-936, -218);  glVertex2f(-824, -218);  glVertex2f(-782, -172);  glVertex2f(-836, -194);  glVertex2f(-930, -194);  glVertex2f(-972, -180);  glEnd();  }  void change\_day\_night(int value){  if (state== 1){  state = 2;  }  else if (state == 2){  state = 1;  }  glutPostRedisplay();  glutTimerFunc(2000,change\_day\_night, 0 );  }  void display() {  if (state == 1)  glClearColor(0.0f, 0.6f,0.90f, 1.0f); // Set background color to black and opaque  else if (state == 2)  glClearColor(0.0f, 0.0f,0.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT);  if (state == 1)  sun();  two\_hills();  grass(); // Clear the color buffer (background)  tree();  house\_1();  khor\_pala();  house\_2();  boat();  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInitWindowSize(1020, 520); // Set the window's initial width & height  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title  glutDisplayFunc(display); // Register display callback handler for window re-paint  gluOrtho2D(-1000, +50, -300, 200);  glutTimerFunc(2000,change\_day\_night, 0 );  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **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** |
| **Code-**  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  #include<iostream>  using namespace std;  /\* Handler for window-repaint event. Call back when the window first appears and  whenever the window needs to be re-painted. \*/  int state = 1;  void khor\_pala(){  //upper extension of the khor pala  glColor3f(0, 0, 0);  glBegin(GL\_POLYGON);  //right yellow chad  glVertex2f(-433.98356, 57.3442926);  glVertex2f(-434, 80);  glVertex2f(-425, 80);  glVertex2f(-425, 57.3442926);  glEnd();  //yellow khor pala  glColor3f(1, 1, 0);  glBegin(GL\_POLYGON);  glVertex2f(-387.48784, 23.76182488);  glVertex2f(-400, 45);  glVertex2f(-430, 60);  glVertex2f(-468, 34);  glVertex2f(-484, 6);  glVertex2f(-490, -52);  glVertex2f(-486, -78);  glVertex2f(-455, -100);  glVertex2f(-405, -100);  glVertex2f(-380, -80);  glVertex2f(-380, -20);  glEnd();  }  void house\_2(){  //right yellow chad  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  glVertex2f(-365, 50);  glVertex2f(-320, -20);  glVertex2f(-165, -20);  glVertex2f(-225, 50);  glEnd();  glColor3f(1, 0.5, 0);  glBegin(GL\_POLYGON);  //left yellow chad  glVertex2f(-365, 50);  glVertex2f(-425, -20);  glVertex2f(-380, -20);  glVertex2f(-347.9209, 23.43264033);  glEnd();  //under 1 gray color area (door)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-347.9209, 23.43264033);  glVertex2f(-380, -20);  glVertex2f(-380, -80);  glVertex2f(-320, -90);  glVertex2f(-320, -20);  glEnd();  //under 2 gray color area (window)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-320, -20);  glVertex2f(-320, -90);  glVertex2f(-185, -65);  glVertex2f(-185, -20);  glEnd();  //under extension of house 2  // number 1(door)  glColor3f(0.9, 0.8, 0.7);  glBegin(GL\_POLYGON);  glVertex2f(-380, -80);  glVertex2f(-405, -100);  glVertex2f(-320, -115);  glVertex2f(-320, -90);  glEnd();  // number 2(extension)  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-320, -90);  glVertex2f(-320, -115);  glVertex2f(-170, -80);  glVertex2f(-185, -65);  glEnd();  //doo4  if (state == 1)  glColor3f(1, 0.5, 0);  else if (state == 2)  glColor3f(0, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(-360, -30);  glVertex2f(-360, -70);  glVertex2f(-340, -70);  glVertex2f(-340, -30);  glEnd();  //window  if (state == 1)  glColor3f(1, 0.5, 0);  else if (state == 2)  glColor3f(0, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(-280, -30);  glVertex2f(-280, -60);  glVertex2f(-250, -60);  glVertex2f(-250, -30);  glEnd();  }  void house\_1(){  //black area under traffic light poll  //upper portion  glColor3f(1, 0.7, 0);  glBegin(GL\_POLYGON);  glVertex2f(-455, 50);  glVertex2f(-555, 50);  glVertex2f(-650, -25);  glVertex2f(-490, -25);  glVertex2f(-440, 0);  glEnd();  //lowerportion  glColor3f(0.9, 0.9, 0.9);  glBegin(GL\_POLYGON);  glVertex2f(-620.0788546, -25);  glVertex2f(-619.975899, -78);  glVertex2f(-486, -78);  glVertex2f(-486, -25);  glEnd();  //extream lower portion  glColor3f(1, 1, 1);  glBegin(GL\_POLYGON);  glVertex2f(-636, -78);  glVertex2f(-636, -100);  glVertex2f(-455, -100);  glVertex2f(-455, -78);  glEnd();  //door  if (state == 1)  glColor3f(1, 0.5, 0);  else if (state == 2)  glColor3f(0, 0, 0);  glBegin(GL\_POLYGON);  glVertex2f(-570, -30);  glVertex2f(-570, -70);  glVertex2f(-540, -70);  glVertex2f(-540, -30);  glEnd();  }  void grass(){  glColor3f(0, 1, 0);  glBegin(GL\_POLYGON);  glVertex2f(-1000, -75);  glVertex2f(-1000, -100);  glVertex2f(-710, -140);  glVertex2f(-765, -180);  glVertex2f(-425, -230);  glVertex2f(-445, -250);  glVertex2f(-435, -265);  glVertex2f(-380, -275);  glVertex2f(-315, -280);  glVertex2f(-150, -300);  glVertex2f(50, -300);  glVertex2f(50, -40);  glEnd();  }  void tree(){  glBegin(GL\_POLYGON);  glColor3f(0.0f, 1.0f, 0.0f);  glVertex2f(-140, 30.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-168, 16.5);  glVertex2f(-179, 27);  glVertex2f(-180, 40);  glVertex2f(-189, 44);  glVertex2f(-197.2, 53.4);  glVertex2f(-200.2, 68);  glVertex2f(-197.5,80.5);  glVertex2f(-192.5,87.5);  glVertex2f(-184.3,91.3);  glVertex2f(-182.6,101.1);  glVertex2f(-176,109);  glVertex2f(-166,111.5);  glVertex2f(-155.8,111.85 );  glVertex2f(-150.8,110.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-140.4,114 );  glVertex2f(-129.8,117.6 );  glVertex2f(-119.05,117.35 );  glVertex2f(-111.1,113.7 );  glVertex2f(-104.4,106.2 );  glVertex2f(-100,100 );  glVertex2f(-92,100.6 );  glVertex2f(-82,97.4 );  glVertex2f(-76.1,88.8);  glVertex2f(-76,78.8);  glVertex2f(-78.45, 71.9);  glVertex2f(-73.6, 67.8);  glVertex2f(-70.2, 61.4);  glVertex2f(-71, 51);  glVertex2f(-75.86, 43.68);  glVertex2f(-73.04, 37.78);  glVertex2f(-70, 30);  glVertex2f(-73, 19);  glVertex2f(-85, 8.5);  glVertex2f(-99, 7.2);  glVertex2f(-112, 10);  glVertex2f(-120, 24.5);  glEnd();  glColor3f(0.5f, 0.3f, 0.0f);  glBegin(GL\_POLYGON);  /\*  glVertex2f(-120, 24.5);  glVertex2f(-140, 24.5);  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, -24.5);  \*/  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, 24.5);  glVertex2f(-140, 24.5);  /\*  \*/  glEnd();  glBegin(GL\_POLYGON);  glVertex2f(-120, 24.5);  glVertex2f(-107.7, 35);  glVertex2f(-113.7, 42);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 43);  glVertex2f(-140.2, 50.6);  glVertex2f(-144.7, 42);  glVertex2f(-140, 37);  glVertex2f(-140, 24.5);  glEnd();  }  void outline\_tree(){  glColor3f(0, 0, 0);  glLineWidth(1.5);  glBegin(GL\_LINES);  glVertex2f(-140, -65);  glVertex2f(-162, -83);  glVertex2f(-162, -83);  glVertex2f(-108, -83);  glVertex2f(-108, -83);  glVertex2f(-120, -65);  glVertex2f(-120, -65);  glVertex2f(-120, 24.5);  glVertex2f(-120, 24.5);  glVertex2f(-107.7, 35);  glVertex2f(-107.7, 35);  glVertex2f(-113.7, 42);  glVertex2f(-113.7, 42);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 37);  glVertex2f(-119.7, 61);  glVertex2f(-119.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 61);  glVertex2f(-131.7, 43);  glVertex2f(-131.7, 43);  glVertex2f(-140.2, 50.6);  glVertex2f(-140.2, 50.6);  glVertex2f(-144.7, 42);  glVertex2f(-144.7, 42);  glVertex2f(-140, 37);  glVertex2f(-140, 37);  glVertex2f(-140, 24.5);  glVertex2f(-140, 24.5);  glVertex2f(-140, -65);  //round green leafs  glVertex2f(-140, 30.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-152.5, 17.5);  glVertex2f(-168, 16.5);  glVertex2f(-168, 16.5);  glVertex2f(-179, 27);  glVertex2f(-179, 27);  glVertex2f(-180, 40);  glVertex2f(-180, 40);  glVertex2f(-189, 44);  glVertex2f(-189, 44);  glVertex2f(-197.2, 53.4);  glVertex2f(-197.2, 53.4);  glVertex2f(-200.2, 68);  glVertex2f(-200.2, 68);  glVertex2f(-197.5,80.5);  glVertex2f(-197.5,80.5);  glVertex2f(-192.5,87.5);  glVertex2f(-192.5,87.5);  glVertex2f(-184.3,91.3);  glVertex2f(-184.3,91.3);  glVertex2f(-182.6,101.1);  glVertex2f(-182.6,101.1);  glVertex2f(-176,109);  glVertex2f(-176,109);  glVertex2f(-166,111.5);  glVertex2f(-166,111.5);  glVertex2f(-155.8,111.85 );  glVertex2f(-155.8,111.85 );  glVertex2f(-150.8,110.2 );  glVertex2f(-150.8,110.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-145.4,109.2 );  glVertex2f(-140.4,114 );  glVertex2f(-140.4,114 );  glVertex2f(-129.8,117.6 );  glVertex2f(-129.8,117.6 );  glVertex2f(-119.05,117.35 );  glVertex2f(-119.05,117.35 );  glVertex2f(-111.1,113.7 );  glVertex2f(-111.1,113.7 );  glVertex2f(-104.4,106.2 );  glVertex2f(-104.4,106.2 );  glVertex2f(-100,100 );  glVertex2f(-100,100 );  glVertex2f(-92,100.6 );  glVertex2f(-92,100.6 );  glVertex2f(-82,97.4 );  glVertex2f(-82,97.4 );  glVertex2f(-76.1,88.8);  glVertex2f(-76.1,88.8);  glVertex2f(-76,78.8);  glVertex2f(-76,78.8);  glVertex2f(-78.45, 71.9);  glVertex2f(-78.45, 71.9);  glVertex2f(-73.6, 67.8);  glVertex2f(-73.6, 67.8);  glVertex2f(-70.2, 61.4);  glVertex2f(-70.2, 61.4);  glVertex2f(-71, 51);  glVertex2f(-71, 51);  glVertex2f(-75.86, 43.68);  glVertex2f(-75.86, 43.68);  glVertex2f(-73.04, 37.78);  glVertex2f(-73.04, 37.78);  glVertex2f(-70, 30);  glVertex2f(-70, 30);  glVertex2f(-73, 19);  glVertex2f(-73, 19);  glVertex2f(-85, 8.5);  glVertex2f(-85, 8.5);  glVertex2f(-99, 7.2);  glVertex2f(-99, 7.2);  glVertex2f(-112, 10);  glVertex2f(-112, 10);  glVertex2f(-120, 16.5);  //glVertex2f(-120, 24.5);  glEnd();  }  void two\_hills(){  // first hill  glColor3f(0, 0.8, 0);  glBegin(GL\_POLYGON);  glVertex2f(-1000, -45.5);  glVertex2f(-900, 50);  glVertex2f(-883, 61);  glVertex2f(-870, 61);  glVertex2f(-850, 51);  glVertex2f(-812.251844, 26.727090);  glVertex2f(-895, -75);  glVertex2f(-1000, -75);  glEnd();  //second hill  glBegin(GL\_POLYGON);  glColor3f(0, 0.5, 0);  glVertex2f(-895, -75);  //glVertex2f(-812.251844, 26.727090);  glVertex2f(-786, 59);  glVertex2f(-780, 65);  glVertex2f(-773, 65);  glVertex2f(-764, 61);  glVertex2f(-580, -75);  glEnd();  }  void sun(){  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<360;i++)  {  glColor3f(1,0.9,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r= 85.3901037;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x - 824.535315,y + 53.8050655 );  }  //glVertex2f(0.3f,0.4f);  //glVertex2f(0.1f,0.4f);  glEnd();  glLineWidth(5);  glColor3f(1, 0.9, 0);  glBegin(GL\_LINES);  glVertex2f(-968, 48);  glVertex2f(-926, 56);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-980, 88);  glVertex2f(-918, 70);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-954, 124);  glVertex2f(-922, 104);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-954, 124);  glVertex2f(-922, 104);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-925, 150);  glVertex2f(-896, 124);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-888,158);  glVertex2f(-874,138);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-852, 194);  glVertex2f(-850, 150);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-814, 170);  glVertex2f(-816, 152);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-758, 172);  glVertex2f(-780, 142);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-744, 142);  glVertex2f(-760, 128);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-680, 150);  glVertex2f(-740, 110);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-680, 100);  glVertex2f(-730, 90);  glEnd();  glBegin(GL\_LINES);  glVertex2f(-728, 66);  glVertex2f(-674, 58);  glEnd();  }  void boat(){  //sitting area  glBegin(GL\_POLYGON);  glColor3f(0.5, 0.3, 0);  glVertex2f(-936, -218);  glVertex2f(-824, -218);  glVertex2f(-800.6849, -179.6123);  glVertex2f(-972, -180);  glEnd();  //shade  glBegin(GL\_POLYGON);  glColor3f(1, .9, .6);  glVertex2f(-880, -194);  glVertex2f(-836, -194);  glVertex2f(-817.9758973, -186.61688);  glVertex2f(-826.261432, -169.25207);  glVertex2f(-850, -148);  glVertex2f(-907, -148);  glVertex2f(-894, -157);  glVertex2f(-885.1914705, -180.2796);  glEnd();  //shade left  glBegin(GL\_POLYGON);  glColor3f(1, .9, .6);  glVertex2f(-936.8605, -180.1132086);  glVertex2f(-885.1914705, -180.2796);  glVertex2f(-894, -157);  glVertex2f(-907, -148);  glVertex2f(-924, -156);  glVertex2f(-934, -174);  glEnd();  //stick  glLineWidth(6);  glBegin(GL\_LINES);  glColor3f(0, 0, 0);  glVertex2f(-846, -240);  glVertex2f(-806, -106);  glEnd();  //black lower area  glBegin(GL\_POLYGON);  glColor3f(0, 0, 0);  glVertex2f(-936, -218);  glVertex2f(-824, -218);  glVertex2f(-782, -172);  glVertex2f(-836, -194);  glVertex2f(-930, -194);  glVertex2f(-972, -180);  glEnd();  }  /\*void change\_day\_night(int value){  if (state== 1){  state = 2;  }  else if (state == 2){  state = 1;  }  glutPostRedisplay();  glutTimerFunc(2000,change\_day\_night, 0 );  }\*/  void handleKeypress(unsigned char key, int x, int y) {  switch (key) {  case 'd' :  state = 1;  cout << "d pressed" << endl;  glutPostRedisplay();  break;  case 'D' :  state = 1;  cout << "D pressed" << endl;  glutPostRedisplay();  break;  case 'n' :  state = 2;  cout << "n pressed" << endl;  glutPostRedisplay();  break;  case 'N' :  state = 2;  cout << "N pressed" << endl;  glutPostRedisplay();  break;  }  }  void display() {  //cout << state << endl;  if (state == 1)  glClearColor(0.0f, 0.6f,0.90f, 1.0f); // Set background color to black and opaque  else if (state == 2)  glClearColor(0.0f, 0.0f,0.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT);  if (state == 1)  sun();  two\_hills();  grass(); // Clear the color buffer (background)  tree();  house\_1();  khor\_pala();  house\_2();  boat();  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInitWindowSize(1020, 520); // Set the window's initial width & height  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title  glutDisplayFunc(display); // Register display callback handler for window re-paint  gluOrtho2D(-1000, +50, -300, 200);  //glutTimerFunc(2000,change\_day\_night, 0 );  glutKeyboardFunc(handleKeypress);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |