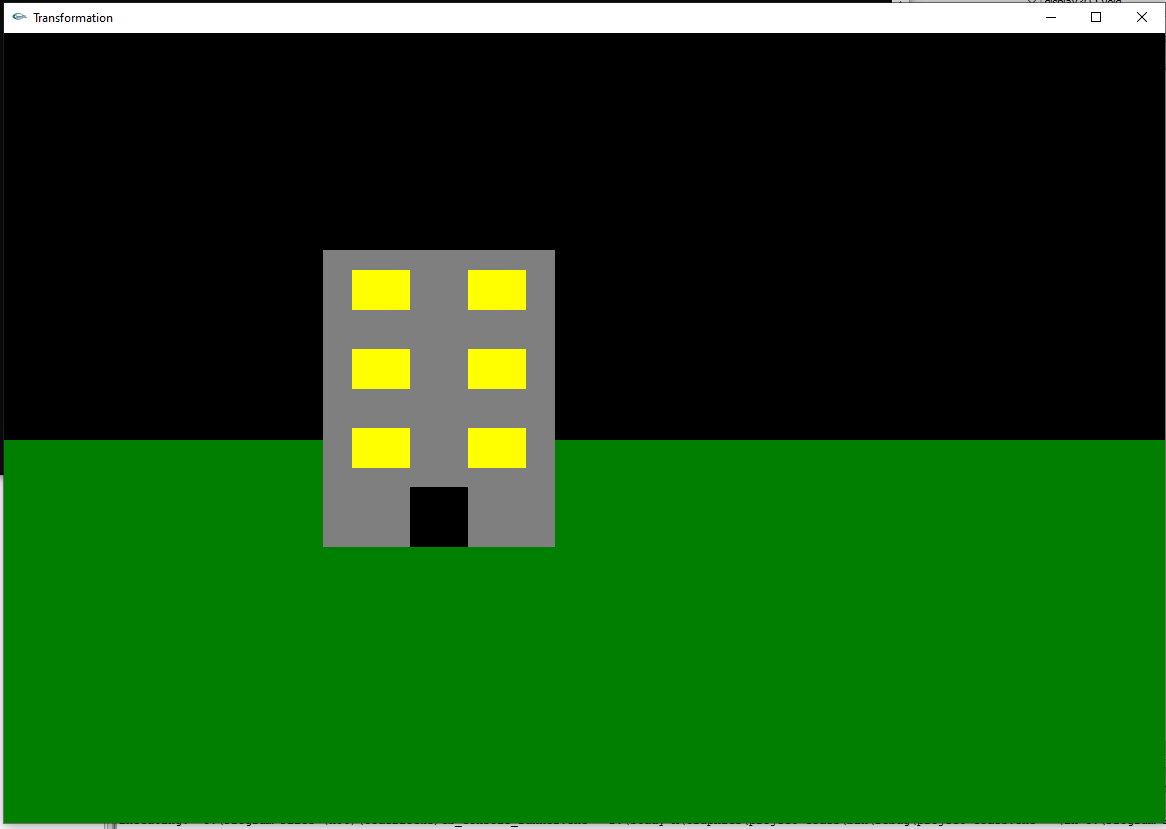
**Lab Practice-7**

Submission Guidelines-

* Rename the file to your id only. If your id is 18-XXXXX-1, then the file name must be 18-XXXXX-1.docx.

|  |
| --- |
| **Question-**  Create a simple day and night scenario that will automatically change from day to night |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **#include <math.h>**  **using namespace std;**  **GLfloat position = 0.0f;**  **//GLfloat position1 = 0.0f;**  **GLfloat speed =0.1f;**  **void dis();**  **void display();**  **void update(int value)**  **{**  **if(position<-1.5)**  **position=1.0f;**  **position -= speed;**  **glutPostRedisplay();**  **glutTimerFunc(100,update,0);**  **}**  **void sky1(){**  **///sky///**  **glPushMatrix();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **glColor3ub(135,206,250);**  **glVertex2f(38.0f,-3.0f);**  **glVertex2f(38.0f,14.0f);**  **glVertex2f(-12.0f,14.0f);**  **glVertex2f(-12.0f,-3.0f);**  **glEnd();}**  **void sky2(){**  **///sky///**  **glPushMatrix();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(38.0f,-3.0f);**  **glVertex2f(38.0f,14.0f);**  **glVertex2f(-12.0f,14.0f);**  **glVertex2f(-12.0f,-3.0f);**  **glEnd();}**  **void init()**  **{**  **glClearColor(0.0f,0.0f,0.0f,1.0f);**  **}**  **void disback(int val)**  **{**  **glutDisplayFunc(display);**  **}**  **void display4(int val)**  **{**  **glutDisplayFunc(display);**  **}**  **void display3()**  **{ glClear(GL\_COLOR\_BUFFER\_BIT);**  **glColor3d(1,0,0);**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-12,38,-19,14);//range**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **glColor3d(0.0,0.5,0.0);**  **glVertex2f(38.0f,-40.0f);**  **glVertex2f(38.0f,40.0f);**  **glVertex2f(-12.0f,40.0f);**  **glVertex2f(-12.0f,-40.0f);**  **glEnd();**  **sky2();**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-2,2,-2,2);//range**  **glBegin(GL\_POLYGON);**  **glColor3f(0.5, 0.5, 0.5);**  **glVertex2f(-.9f, -0.6f);**  **glVertex2f(-.1f, -0.6f);**  **glVertex2f(-.1f, 0.9f);**  **glVertex2f(-.9f, 0.9f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0,0,0);**  **glVertex2f(-.6f, -0.6f);**  **glVertex2f(-.4f, -0.6f);**  **glVertex2f(-.4f, -0.3f);**  **glVertex2f(-.6f, -0.3f);**  **glEnd();**  **//glPopMatrix();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, -0.0f);**  **glVertex2f(-.6f, -0.0f);**  **glVertex2f(-.6f, -0.2f);**  **glVertex2f(-.8f, -0.2f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, 0.2f);**  **glVertex2f(-.6f, 0.2f);**  **glVertex2f(-.6f, 0.4f);**  **glVertex2f(-.8f, 0.4f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, 0.6f);**  **glVertex2f(-.6f, 0.6f);**  **glVertex2f(-.6f, 0.8f);**  **glVertex2f(-.8f, 0.8f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, -0.0f);**  **glVertex2f(-.2f, -0.0f);**  **glVertex2f(-.2f, -0.2f);**  **glVertex2f(-.4f, -0.2f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, 0.2f);**  **glVertex2f(-.2f, 0.2f);**  **glVertex2f(-.2f, 0.4f);**  **glVertex2f(-.4f, 0.4f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, 0.6f);**  **glVertex2f(-.2f, 0.6f);**  **glVertex2f(-.2f, 0.8f);**  **glVertex2f(-.4f, 0.8f);**  **glEnd();**  **glPopMatrix();**  **glPopMatrix();**  **glutTimerFunc(2000,display4,0);**  **glFlush();**  **}**  **void display2(int val)**  **{**  **glutDisplayFunc(display3);**  **}**  **void display()**  **{ glClear(GL\_COLOR\_BUFFER\_BIT);**  **glColor3d(1,0,0);**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-12,38,-19,14);//range**  **glMatrixMode(GL\_MODELVIEW);**  **//sky1();**  **glPushMatrix();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **glColor3d(0,255,0);**  **glVertex2f(38.0f,-19.0f);**  **glVertex2f(38.0f,10.0f);**  **glVertex2f(-12.0f,10.0f);**  **glVertex2f(-12.0f,-19.0f);**  **glEnd();**  **sky1();**  **glPushMatrix();**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-2,2,-2,2);//range**  **glBegin(GL\_POLYGON);**  **glColor3f(0.5, 0.5, 0.5);**  **glVertex2f(-.9f, -0.6f);**  **glVertex2f(-.1f, -0.6f);**  **glVertex2f(-.1f, 0.9f);**  **glVertex2f(-.9f, 0.9f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0,0,0);**  **glVertex2f(-.6f, -0.6f);**  **glVertex2f(-.4f, -0.6f);**  **glVertex2f(-.4f, -0.3f);**  **glVertex2f(-.6f, -0.3f);**  **glEnd();**  **//glPopMatrix();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, -0.0f);**  **glVertex2f(-.6f, -0.0f);**  **glVertex2f(-.6f, -0.2f);**  **glVertex2f(-.8f, -0.2f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, 0.2f);**  **glVertex2f(-.6f, 0.2f);**  **glVertex2f(-.6f, 0.4f);**  **glVertex2f(-.8f, 0.4f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, 0.6f);**  **glVertex2f(-.6f, 0.6f);**  **glVertex2f(-.6f, 0.8f);**  **glVertex2f(-.8f, 0.8f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, -0.0f);**  **glVertex2f(-.2f, -0.0f);**  **glVertex2f(-.2f, -0.2f);**  **glVertex2f(-.4f, -0.2f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, 0.2f);**  **glVertex2f(-.2f, 0.2f);**  **glVertex2f(-.2f, 0.4f);**  **glVertex2f(-.4f, 0.4f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, 0.6f);**  **glVertex2f(-.2f, 0.6f);**  **glVertex2f(-.2f, 0.8f);**  **glVertex2f(-.4f, 0.8f);**  **glEnd();**  **glPopMatrix();**  **glPopMatrix();**  **glutTimerFunc(2000,display2,0);**  **glFlush();**  **}**  **void dis()**  **{**  **sky1();**  **glutDisplayFunc(display);**  **//sky();**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitWindowSize(2000, 1500);**  **glutCreateWindow("Transformation");**  **glutDisplayFunc(dis);**  **init();**  **gluOrtho2D(-12,38,-19,14);**  **glutTimerFunc(100, update, 0); //Add a timer**  **glutMainLoop();**  **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. |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **#include <math.h>**  **using namespace std;**  **GLfloat position = 0.0f;**  **//GLfloat position1 = 0.0f;**  **GLfloat speed =0.1f;**  **void sky1(){**  **///sky///**  **glPushMatrix();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **glColor3ub(135,206,250);**  **glVertex2f(38.0f,-3.0f);**  **glVertex2f(38.0f,14.0f);**  **glVertex2f(-12.0f,14.0f);**  **glVertex2f(-12.0f,-3.0f);**  **glEnd();}**  **void sky2(){**  **///sky///**  **glPushMatrix();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **glColor3f(0.0f, 0.0f, 0.0f);**  **glVertex2f(38.0f,-3.0f);**  **glVertex2f(38.0f,14.0f);**  **glVertex2f(-12.0f,14.0f);**  **glVertex2f(-12.0f,-3.0f);**  **glEnd();}**  **void init()**  **{**  **glClearColor(0.0f,0.0f,0.0f,1.0f);**  **}**  **void display3()**  **{ glClear(GL\_COLOR\_BUFFER\_BIT);**  **glColor3d(1,0,0);**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-12,38,-19,14);//range**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **glColor3d(0.0,0.5,0.0);**  **glVertex2f(38.0f,-40.0f);**  **glVertex2f(38.0f,40.0f);**  **glVertex2f(-12.0f,40.0f);**  **glVertex2f(-12.0f,-40.0f);**  **glEnd();**  **sky2();**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-2,2,-2,2);//range**  **glBegin(GL\_POLYGON);**  **glColor3f(0.5, 0.5, 0.5);**  **glVertex2f(-.9f, -0.6f);**  **glVertex2f(-.1f, -0.6f);**  **glVertex2f(-.1f, 0.9f);**  **glVertex2f(-.9f, 0.9f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0,0,0);**  **glVertex2f(-.6f, -0.6f);**  **glVertex2f(-.4f, -0.6f);**  **glVertex2f(-.4f, -0.3f);**  **glVertex2f(-.6f, -0.3f);**  **glEnd();**  **//glPopMatrix();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, -0.0f);**  **glVertex2f(-.6f, -0.0f);**  **glVertex2f(-.6f, -0.2f);**  **glVertex2f(-.8f, -0.2f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, 0.2f);**  **glVertex2f(-.6f, 0.2f);**  **glVertex2f(-.6f, 0.4f);**  **glVertex2f(-.8f, 0.4f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, 0.6f);**  **glVertex2f(-.6f, 0.6f);**  **glVertex2f(-.6f, 0.8f);**  **glVertex2f(-.8f, 0.8f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, -0.0f);**  **glVertex2f(-.2f, -0.0f);**  **glVertex2f(-.2f, -0.2f);**  **glVertex2f(-.4f, -0.2f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, 0.2f);**  **glVertex2f(-.2f, 0.2f);**  **glVertex2f(-.2f, 0.4f);**  **glVertex2f(-.4f, 0.4f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, 0.6f);**  **glVertex2f(-.2f, 0.6f);**  **glVertex2f(-.2f, 0.8f);**  **glVertex2f(-.4f, 0.8f);**  **glEnd();**  **glPopMatrix();**  **glPopMatrix();**  **glFlush();**  **glutSwapBuffers();**  **}**  **void display()**  **{ glClear(GL\_COLOR\_BUFFER\_BIT);**  **glColor3d(1,0,0);**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-12,38,-19,14);//range**  **glMatrixMode(GL\_MODELVIEW);**  **//sky1();**  **glPushMatrix();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **glColor3d(0.0,0.5,0.0);**  **glVertex2f(38.0f,-40.0f);**  **glVertex2f(38.0f,40.0f);**  **glVertex2f(-12.0f,40.0f);**  **glVertex2f(-12.0f,-40.0f);**  **glEnd();**  **sky1();**  **glPushMatrix();**  **glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin**  **glColor3d(0,255,0);**  **glVertex2f(38.0f,-19.0f);**  **glVertex2f(38.0f,10.0f);**  **glVertex2f(-12.0f,10.0f);**  **glVertex2f(-12.0f,-19.0f);**  **glEnd();**  **sky1();**  **glPushMatrix();**  **glLoadIdentity(); //Reset the drawing perspective**  **gluOrtho2D(-2,2,-2,2);//range**  **glBegin(GL\_POLYGON);**  **glColor3f(0.5, 0.5, 0.5);**  **glVertex2f(-.9f, -0.6f);**  **glVertex2f(-.1f, -0.6f);**  **glVertex2f(-.1f, 0.9f);**  **glVertex2f(-.9f, 0.9f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(0,0,0);**  **glVertex2f(-.6f, -0.6f);**  **glVertex2f(-.4f, -0.6f);**  **glVertex2f(-.4f, -0.3f);**  **glVertex2f(-.6f, -0.3f);**  **glEnd();**  **//glPopMatrix();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, -0.0f);**  **glVertex2f(-.6f, -0.0f);**  **glVertex2f(-.6f, -0.2f);**  **glVertex2f(-.8f, -0.2f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, 0.2f);**  **glVertex2f(-.6f, 0.2f);**  **glVertex2f(-.6f, 0.4f);**  **glVertex2f(-.8f, 0.4f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.8f, 0.6f);**  **glVertex2f(-.6f, 0.6f);**  **glVertex2f(-.6f, 0.8f);**  **glVertex2f(-.8f, 0.8f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, -0.0f);**  **glVertex2f(-.2f, -0.0f);**  **glVertex2f(-.2f, -0.2f);**  **glVertex2f(-.4f, -0.2f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, 0.2f);**  **glVertex2f(-.2f, 0.2f);**  **glVertex2f(-.2f, 0.4f);**  **glVertex2f(-.4f, 0.4f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3f(255,255,0);**  **glVertex2f(-.4f, 0.6f);**  **glVertex2f(-.2f, 0.6f);**  **glVertex2f(-.2f, 0.8f);**  **glVertex2f(-.4f, 0.8f);**  **glEnd();**  **glPopMatrix();**  **glPopMatrix();**  **glFlush();**  **glutSwapBuffers();**  **}**  **void handleKeypress(unsigned char key, int x, int y) {**  **switch (key) {**  **case 'd':**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(2000, 1500);**  **glutCreateWindow("Basic Animation");**  **glutDisplayFunc(display);**  **init();**  **glutKeyboardFunc(handleKeypress);**  **break;**  **glutPostRedisplay();**  **case 'n':**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(2000, 1500);**  **glutCreateWindow("Basic Animation");**  **glutDisplayFunc(display3);**  **init();**  **glutKeyboardFunc(handleKeypress);**  **break;**  **glutPostRedisplay();**  **}**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitWindowSize(2000, 1500);**  **//glutInitWindowPosition(50,50);**  **glutCreateWindow("Transformation");**  **glutDisplayFunc(display);**  **glutKeyboardFunc(handleKeypress);**  **init();**  **gluOrtho2D(-12,38,-19,14);**  **glutKeyboardFunc(handleKeypress);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |