**Lab Taks-5**

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.
* Must submit within the announced time.
* Must include resources for all the section in the table

|  |
| --- |
| **Question-1**  Create an animation using two box that will move in the opposite direction. |
| **Graph Plot (Picture)-**  **[Not needed]** |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **using namespace std;**  **float \_move = 0.0f;**  **float \_move2 = 0.0f;**  **void initGL() {**  **glClearColor(0.0f, 0.0f, 0.0f, 0.0f);**  **}**  **void drawScene() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glColor3d(255, 255, 255);**  **glLoadIdentity();**  **glMatrixMode(GL\_MODELVIEW);**  **GLfloat x1=0.2f, x2=-0.2f;**  **GLfloat y1=-0.1f, y2=-0.4f;**  **GLfloat mov= \_move;**  **for(int i=0; i<2; i++)**  **{**  **glPushMatrix();**  **glTranslatef(mov, 0.0f, 0.0f);**  **glBegin(GL\_QUADS);**  **glVertex2f(x1, y1);**  **glVertex2f(x2, y1);**  **glVertex2f(x2, y2);**  **glVertex2f(x1, y2);**  **glEnd();**  **glPopMatrix();**  **y1=-y1, y2=-y2;**  **mov=\_move2;**  **}**  **glutSwapBuffers();**  **}**  **void update(int value) {**  **\_move += .02;**  **if(\_move > 1.4)**  **{**  **\_move = -1.0;**  **}**  **\_move2 -= .02;**  **if(\_move2 < -1.4)**  **{**  **\_move2 = 1.0;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update, 0);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(800, 800);**  **glutCreateWindow("Box Animation");**  **glutDisplayFunc(drawScene);**  **glutTimerFunc(20, update, 0);**  **initGL();**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-2**  Design a car which will have rotating wheels. |
| **Graph Plot (Picture)-**  **[Not needed]** |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **#include<math.h>>**  **# define PI 3.14159265358979323846**  **using namespace std;**  **float \_move = 0.0f;**  **float \_move2 = 0.0f;**  **void initGL() {**  **glClearColor(0, 0, 255, 1.0f);**  **}**  **void drawScene() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glLoadIdentity();**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(\_move, 0.0f, 0.0f);**  **/////Car Body/////**  **glColor3d(255,255,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.2f, 0.2f);**  **glVertex2f(-0.2f, 0.2f);**  **glVertex2f(-0.5f, 0.0f);**  **glVertex2f(0.4f, 0.0f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(0.4f, 0.0f);**  **glVertex2f(-0.5f, 0.0f);**  **glVertex2f(-0.5f, -0.3f);**  **glVertex2f(0.4f, -0.3f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.5f, 0.0f);**  **glVertex2f(-0.85f, -0.05f);**  **glVertex2f(-0.85f, -0.3f);**  **glVertex2f(-0.5f, -0.3f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(0.7f, -0.05f);**  **glVertex2f(0.4f, 0.0f);**  **glVertex2f(0.4f, -0.3f);**  **glVertex2f(0.7f, -0.3f);**  **glEnd();**  **/////Windows/////**  **glColor3d(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.17f, 0.18f);**  **glVertex2f(-0.18f, 0.18f);**  **glVertex2f(-0.46f, 0.0f);**  **glVertex2f(0.35f, 0.0f);**  **glEnd();**  **/////Wheels/////**  **glColor3d(0,0,0);**  **GLfloat a=-0.5f; GLfloat b=-0.3f; GLfloat radius2 =.12f;**  **int triangleAmount2 = 100;**  **GLfloat twicePi2 = 2.0f \* PI;**  **for(int j=0;j<2;j++)**  **{**  **glBegin(GL\_TRIANGLE\_FAN);**  **glVertex2f(a, b);**  **for(int i = 0; i <= triangleAmount2; i++) {**  **glVertex2f(**  **a + (radius2 \* cos(i \* twicePi2 / triangleAmount2)),**  **b + (radius2 \* sin(i \* twicePi2 / triangleAmount2))**  **);**  **}**  **glEnd();**  **a=b+0.7f;**  **}**  **glPopMatrix();**  **glutSwapBuffers();**  **}**  **void update(int value)**  **{**  **\_move += .02;**  **if(\_move > 1.3)**  **{**  **\_move = -1.0;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update, 0);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(1200, 900);**  **glutCreateWindow("Car Animation");**  **glutDisplayFunc(drawScene);**  **glutTimerFunc(20, update, 0);**  **initGL();**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-3**  Now move your car of question-2 from left to right in a loop. |
| **Graph Plot (Picture)-**  **[Not needed]** |
| **Code-**  **#include <iostream>**  **#include<GL/gl.h>**  **#include <GL/glut.h>**  **#include<math.h>>**  **# define PI 3.14159265358979323846**  **using namespace std;**  **float \_move = 0.0f;**  **float \_move2 = 0.0f;**  **void initGL() {**  **glClearColor(0, 0, 255, 1.0f);**  **}**  **void drawScene() {**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glLoadIdentity();**  **glMatrixMode(GL\_MODELVIEW);**  **glPushMatrix();**  **glTranslatef(\_move, 0.0f, 0.0f);**  **/////Car Body/////**  **glColor3d(255,255,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.2f, 0.2f);**  **glVertex2f(-0.2f, 0.2f);**  **glVertex2f(-0.5f, 0.0f);**  **glVertex2f(0.4f, 0.0f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(0.4f, 0.0f);**  **glVertex2f(-0.5f, 0.0f);**  **glVertex2f(-0.5f, -0.3f);**  **glVertex2f(0.4f, -0.3f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(-0.5f, 0.0f);**  **glVertex2f(-0.85f, -0.05f);**  **glVertex2f(-0.85f, -0.3f);**  **glVertex2f(-0.5f, -0.3f);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glVertex2f(0.7f, -0.05f);**  **glVertex2f(0.4f, 0.0f);**  **glVertex2f(0.4f, -0.3f);**  **glVertex2f(0.7f, -0.3f);**  **glEnd();**  **/////Windows/////**  **glColor3d(0,0,0);**  **glBegin(GL\_QUADS);**  **glVertex2f(0.17f, 0.18f);**  **glVertex2f(-0.18f, 0.18f);**  **glVertex2f(-0.46f, 0.0f);**  **glVertex2f(0.35f, 0.0f);**  **glEnd();**  **/////Wheels/////**  **glColor3d(0,0,0);**  **GLfloat a=-0.5f; GLfloat b=-0.3f; GLfloat radius2 =.12f;**  **int triangleAmount2 = 100;**  **GLfloat twicePi2 = 2.0f \* PI;**  **for(int j=0;j<2;j++)**  **{**  **glBegin(GL\_TRIANGLE\_FAN);**  **glVertex2f(a, b);**  **for(int i = 0; i <= triangleAmount2; i++) {**  **glVertex2f(**  **a + (radius2 \* cos(i \* twicePi2 / triangleAmount2)),**  **b + (radius2 \* sin(i \* twicePi2 / triangleAmount2))**  **);**  **}**  **glEnd();**  **a=b+0.7f;**  **}**  **glPopMatrix();**  **glutSwapBuffers();**  **}**  **void update(int value)**  **{**  **\_move += .02;**  **if(\_move > 1.3)**  **{**  **\_move = -1.0;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(20, update, 0);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**  **glutInitWindowSize(1200, 900);**  **glutCreateWindow("Car Animation");**  **glutDisplayFunc(drawScene);**  **glutTimerFunc(20, update, 0);**  **initGL();**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-4**  Design a windmill with rotating blades |
| **Graph Plot (Picture)-**  **[Not needed]** |
| **Code-**  #include <GL/gl.h>  #include <GL/glut.h>  #include <math.h>  const double PI = 3.141592654;  int frameNumber = 0;  void drawWindmill() {  int i;  glColor3f(2.0f, 0.5f, 1.0f);  glBegin(GL\_POLYGON);  glVertex2f(-0.05f, 0);  glVertex2f(0.05f, 0);  glVertex2f(0.05f, 3);  glVertex2f(-0.05f, 3);  glEnd();  glTranslatef(0, 3, 0);  glRotated(frameNumber \* (180.0/46), 0, 0, 1);  glColor3f(0.4f, 0.4f, 0.8f);  for (i = 0; i < 3; i++) {  glRotated(120, 0, 0, 1); // Note: These rotations accumulate.  glBegin(GL\_POLYGON);  glVertex2f(0,0);  glVertex2f(0.5f, 0.1f);  glVertex2f(1.5f,0);  glVertex2f(0.5f, -0.1f);  glEnd();  }  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT); // Fills the scene with blue.  glLoadIdentity();  glPushMatrix();  glTranslated(5.8,3,0);  glRotated(-frameNumber\*0.7,0,0,1);  glPopMatrix();  glPushMatrix();  glTranslated(3.7,0.8,0);  glScaled(0.7,0.7,1);  drawWindmill();  glPopMatrix();  glPushMatrix();  glTranslated(-3 + 13\*(frameNumber % 300) / 300.0, 0, 0);  glScaled(0.3,0.3,1);  glPopMatrix();  glutSwapBuffers();  }  void doFrame(int v) {  frameNumber++;  glutPostRedisplay();  glutTimerFunc(30,doFrame,0);  }  void init() {  glClearColor(0.0f, 0.0f, 0.0f, 1);  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  glOrtho(0, 7, -1, 4, -1, 1);  glMatrixMode(GL\_MODELVIEW);  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_DOUBLE);  glutInitWindowSize(700,500);  glutInitWindowPosition(100,100);  glutCreateWindow("OpenGL Hierarchical Modeling 2D Example");  init();  glutDisplayFunc(display);  glutTimerFunc(200,doFrame,0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |