Day8 Ex: openGL (GLUT) example showing how to use stencils to draw outlined polygons.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <GL/glut.h>

static int stencilOn = 1;

/\* function declarations \*/

void

  drawScene(void), setMatrix(void), animation(void), resize(int w, int h),

  keyboard(unsigned char c, int x, int y), menu(int choice), drawWireframe(int face),

  drawFilled(int face);

/\* global variables \*/

float ax, ay, az;       /\* angles for animation \*/

/\* coords of a cube \*/

static GLfloat cube[8][3]

= {0.0, 0.0, 0.0,

  1.0, 0.0, 0.0,

  1.0, 0.0, 1.0,

  0.0, 0.0, 1.0,

  1.0, 1.0, 0.0,

  1.0, 1.0, 1.0,

  0.0, 1.0, 1.0,

  0.0, 1.0, 0.0};

static int faceIndex[6][4] =

{0, 1, 2, 3,

  1, 4, 5, 2,

  4, 7, 6, 5,

  7, 0, 3, 6,

  3, 2, 5, 6,

  7, 4, 1, 0};

int

main(int argc, char \*\*argv)

{

  glutInit(&argc, argv);

  glutInitWindowSize(400, 400);

  glutInitDisplayMode(GLUT\_RGB | GLUT\_STENCIL | GLUT\_DOUBLE | GLUT\_DEPTH);

  glutCreateWindow("Stenciled hidden surface removal");

  ax = 10.0;

  ay = -10.0;

  az = 0.0;

  glutDisplayFunc(drawScene);

  glutReshapeFunc(resize);

  glutCreateMenu(menu);

  glutAddMenuEntry("Motion", 3);

  glutAddMenuEntry("Stencil on", 1);

  glutAddMenuEntry("Stencil off", 2);

  glutAttachMenu(GLUT\_RIGHT\_BUTTON);

  glutKeyboardFunc(keyboard);

  glutMainLoop();

  return 0;             /\* ANSI C requires main to return int. \*/

}

void

drawWireframe(int face)

{

  int i;

  glBegin(GL\_LINE\_LOOP);

  for (i = 0; i < 4; i++)

    glVertex3fv((GLfloat \*) cube[faceIndex[face][i]]);

  glEnd();

}

void

drawFilled(int face)

{

  int i;

  glBegin(GL\_POLYGON);

  for (i = 0; i < 4; i++)

    glVertex3fv((GLfloat \*) cube[faceIndex[face][i]]);

  glEnd();

}

void

drawScene(void)

{

  int i;

  glEnable(GL\_DEPTH\_TEST);

  glDepthFunc(GL\_LEQUAL);

  glClearColor(0.0, 0.0, 0.0, 0.0);

  glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

  glPushMatrix();

  glRotatef(ax, 1.0, 0.0, 0.0);

  glRotatef(-ay, 0.0, 1.0, 0.0);

  /\* all the good stuff follows \*/

  if (stencilOn) {

    glEnable(GL\_STENCIL\_TEST);

    glClear(GL\_STENCIL\_BUFFER\_BIT);

    glStencilMask(1);

    glStencilFunc(GL\_ALWAYS, 0, 1);

    glStencilOp(GL\_INVERT, GL\_INVERT, GL\_INVERT);

  }

  glColor3f(1.0, 1.0, 0.0);

  for (i = 0; i < 6; i++) {

    drawWireframe(i);

    if (stencilOn) {

      glStencilFunc(GL\_EQUAL, 0, 1);

      glStencilOp(GL\_KEEP, GL\_KEEP, GL\_KEEP);

    }

    glColor3f(0.0, 0.0, 0.0);

    drawFilled(i);

    glColor3f(1.0, 1.0, 0.0);

    if (stencilOn) {

      glStencilFunc(GL\_ALWAYS, 0, 1);

      glStencilOp(GL\_INVERT, GL\_INVERT, GL\_INVERT);

    }

    glColor3f(1.0, 1.0, 0.0);

    drawWireframe(i);

  }

  glPopMatrix();

  if (stencilOn)

    glDisable(GL\_STENCIL\_TEST);

  /\* end of good stuff \*/

  glutSwapBuffers();

}

void

setMatrix(void)

{

  glMatrixMode(GL\_PROJECTION);

  glLoadIdentity();

  glOrtho(-2.0, 2.0, -2.0, 2.0, -2.0, 2.0);

  glMatrixMode(GL\_MODELVIEW);

  glLoadIdentity();

}

int count = 0;

void

animation(void)

{

  /\* animate the cone \*/

  ax += 5.0;

  ay -= 2.0;

  az += 5.0;

  if (ax >= 360)

    ax = 0.0;

  if (ay <= -360)

    ay = 0.0;

  if (az >= 360)

    az = 0.0;

  glutPostRedisplay();

  count++;

  if (count >= 60)

    glutIdleFunc(NULL);

}

void

menu(int choice)

{

  switch (choice) {

  case 3:

    count = 0;

    glutIdleFunc(animation);

    break;

  case 2:

    stencilOn = 0;

    glutSetWindowTitle("Stencil Disabled");

    glutPostRedisplay();

    break;

  case 1:

    stencilOn = 1;

    glutSetWindowTitle("Stencil Enabled");

    glutPostRedisplay();

    break;

  }

}

/\* ARGSUSED1 \*/

void

keyboard(unsigned char c, int x, int y)

{

  switch (c) {

  case 27:

    exit(0);

    break;

  default:

    break;

  }

}

void

resize(int w, int h)

{

  glViewport(0, 0, w, h);

  setMatrix();

}

OutPut:

