2. Fasea - Objektuen Aldaketak

Abstract

Bigarren fasean objektuen aldaketak inplementatu behar dira; dokumentu honek inplementazioari buruzko zenbait azalpen jasotzen du.

Fase honetan, gutxienez, hiru aldaketa inplementatu behar dira, hala nola, translazioak, biraketak eta tamaina aldaketak. Aldaketak hautaturiko objektuari eragingo zaizkio, eta edozein momentuan aldaketak desegiteko aukera izan behar dugu.

Teorian ikusi duzuen legez, aldaketak egiteko objektuaren koordenatuak 4×4 tamainako matrize batekin biderkatu behar ditugu.

1 OpenGLren matrizeak

`OpenGL`k zenbait matrize erabiltzen ditu. Horietako bat `GL_MODELVIEW` deritzona da. Objektu bat sisteman sartzean, bere koordenatuak matrize horrekin biderkatzen da. Hortaz, objektuen aldaketak kudeatzeko matrize hori erabiliko dugu.

Aurreko fasean objektuen marrazketari ekin genion, ondoko kodea erabiliz:

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

/*First, we draw the axes*/
draw_axes();

/*Now each of the objects in the list*/
while (aux_obj != 0) {
  /* Select the color, depending on whether the current object is the selected one or not */
  ...

  /* Draw the object; for each face create a new polygon with the corresponding vertices */
  glLoadIdentity();
  for (f = 0; f < aux_obj->num_faces; f++) {
    glBegin(GL_POLYGON);
    for (v = 0; v < aux_obj->face_table[f].num_vertices; v++) {
      v_index = aux_obj->face_table[f].vertex_table[v];
      glVertex3d(aux_obj->vertex_table[v_index].coord.x,
             aux_obj->vertex_table[v_index].coord.y,
             aux_obj->vertex_table[v_index].coord.z);
    glEnd
  }
  aux_obj = aux_obj->next;
}
```

Goiko kodean, lehenengo bi lerrotan `glMatrixMode` eta `glLoadIdentity` funtzioak ikusten ditugu. Lehenengoak `OpenGL`ren matrize bat (`GL_MODELVIEW` kasu honetan) aukeratzeko erabiltzen da; bigarrenak aukeratutako matrizean identitatea kargatzen du.

Goiko kodea exekutaturik, erpin bakoitza sartzean bere koordenatuak identitate matrizearekin biderkatzen da eta, ondorioz koordenatuak ez dira aldatzen.

`while` begiztaren barruan, objektu bakoitza marraztu aurretik (aurpegiak sartu aurretik, alegia) berriro ere identitate matrizea kargatzen dugu, matrizean legokeen edozein aldaketa *reseteatzeko*.

Year: 3. Maila
Lecturer: Borja Calvo
Telephone: 943 01 50 13
E-mail: borja.calvo@ehu.es

GIPUZKOAKO CAMPUSA
CAMPUS DE GUIPUZCOA
Paseo Manuel de Lardizabal, 1
20018 Donostia (Gipuzkoa)

Marraztu behar dugun objektuari aldaketarik eragin nahi izanez gero, matrizea hasieratu ondoren guk nahi dugun matrizea sartu beharko genuke, `glMultMatrixd` funtzioa erabiliz; funtzio honek parametro bakarra du, `GLdouble *` bat, non erabili nahi dugun matrizea gordeta izango dugun.

OpenGLn erabiltzen diren matrizeak 16 posizioko `GLdouble` bektoreak dira. Matrize berri bat sortzeko ondoko kodea erabili behar da:

```
GLdouble * matrix = malloc (sizeof(GLdouble)*16);
```

Bektorearen 16 posizio horietan erabili nahi dugun matrizearen balioak sartu behar dira. Hori ondo egiteko, matrizeak zutabez definitzen direla jakin behar da. Hau da, ondoko matrizea sortzeko:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 7 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$$

jarraian dagoen kodea idatziko dugu:

```
GLdouble * m = malloc (sizeof(GLdouble)*16);
m[0]=1;  m[4]=2;  m[8] =3;  m[12]=4;
m[1]=5;  m[5]=6;  m[9] =7;  m[13]=7;
m[2]=9;  m[6]=10; m[10]=11; m[14]=12;
m[3]=13; m[7]=14; m[11]=15; m[15]=16;
```

Behin matrizea definiturik, objektu bat eraldatzeko, `glLoadIdentity` funtzioaren ostean eta aurpegien marrazketarekin hasi aurretik `glMultMatrixd(m);` agindua gehituko diogu kodeari, non `m` aldaketa eragingo duen matrizea den.

Puntu honetan saia zaitezkete, adibidez, hautaturiko objektuaren tamaina aldatzen.

2 Objektu bat, aldaketa pila bat

Kontutan hartu behar da objektu bakoitzak bere aldaketa zerrenda izango duela; are gehiago, zerrenda horretan sartutako aldaketak desegiteko aukera izan behar dugu.

Hori dena inplementatzeko erarik errazena objektu bakoitzari aldaketa zerrenda bat esleitzea da; zerrenda hori datu egitura berria izango da, non eragindako aldaketa guztiak gordeko ditugun.

Dakigunez, aldaketak kateatzeko matrizeak biderkatu behar dira. Beraz, egituran aldaketa matrize indibidualak gordetzen baditugu, aplikatu nahi ditugun bakoitzean biderketak egin beharko ditugu. Horren ordez, zerrendan matrizeak biderkaturik gorde ditzakegu, hori askoz ere eraginkorragoa izango baita. Hau da, T_1, T_2, T_3 eta T_4 aldaketa matrizeak erabili baldin baditugu, zerrendan hauxe izango dugu: $(T_1, T_1T_2, T_1T_2T_3, T_1T_2T_3T_4)$. Azken aldaketa desegin nahi badugu, zerrendaren azken elementua ezabatzea besterik ez dugu egin behar.

3 Interfazearen funtzionamendua

Gure programak egoera-makina moduan funtzionatuko du. Hau da, aldaketa modu bat aktibatzen denean (M, B edo T sakatzen dugunean), hortik aurrera geziekin edota `AVPAG`, `REPAG` teklekin eragindako aldaketak aukeratutako modukoak izango dira. Esate baterako, M (edo m) sakatzen badugu, eskumako geziak objektua mugituko du, baina B (edo b) sakatzen badugu, hortik aurrera gezia objektua biratzeko erabiliko dugu.

Aldaketak aplikatzeko erreferentzi sistemaren aukeraketak ere horrela funtzionatu beharko du; G edo g sakatzen badugu aldaketa guztiak globalak izango dira, L edo l tekla sakatu arte.