

KbG proiektua: 2. fasea

Oier Irazabal

Jesus Calleja

November 14, 2018

Contents

1	Inplementazio berriak	2
1.1	Objektuen transformazioak	2
1.2	Transformazioen desagitea	3
2	Aldaketak	3
3	Gehikuntzak	4
3.1	Aldaketak berregiteko aukera	4
3.2	Transformazio gehigarriak	4

1 Implementazio berriak

1.1 Objektuen transformazioak

Hiru oinarritzko transformazioak (translazioa, errotazioa eta eskalaketa) inplementatu ditugu, erreferentzia-sistema globala zein lokala izanik. Bai transformazio mota, bai erreferentzia-sistema automata baten bidez kodetuta daude. Automataren egoera batetik beste batera mugitzeko taula hauetan jasotako teklak sakatu behar dira:

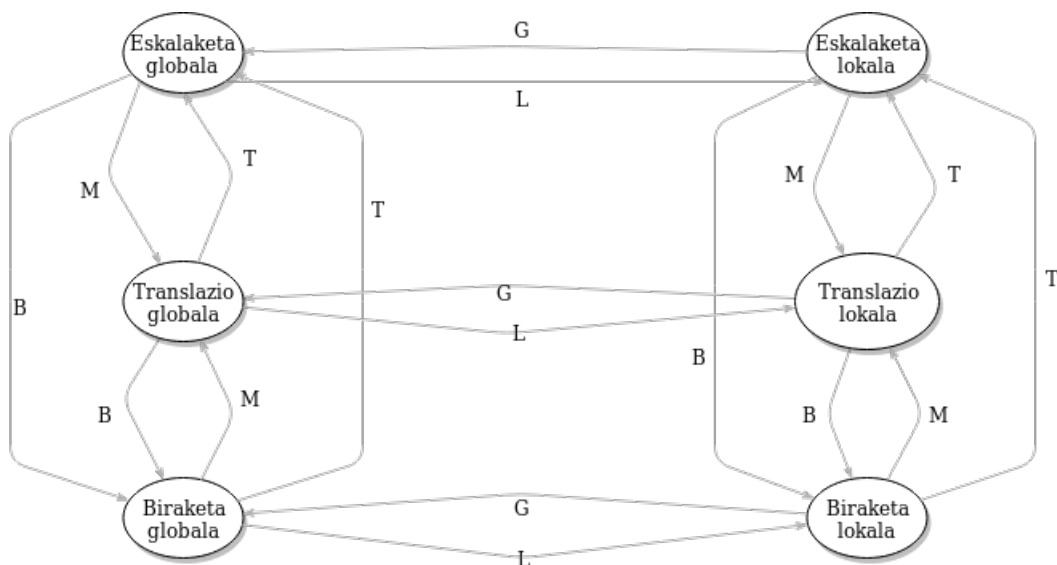
Tekla	Ekintza
M edo m	Translazioa aktibatu
B edo b	Biraketa aktibatu
T edo t	Tamaina aldaketa aktibatu
L edo l	Erreferentzia-sistema lokalera aldatu
G edo g	Erreferentzia-sistema globalera aldatu
+	Ardatz guztietan handitu objektua
-	Ardatz guztietan handitu objektua

1. Taula: Tekla arrunten taula

Tekla	Ekintza		
	Translazioaren norabidea	Biraketaren ardatza	Eskalaketaren norabidea
Goranzko gezia	Mugitu +Y	Biratu +X	Txikitu Y
Beheranzko gezia	Mugitu -Y	Biratu -X	Handitu Y
Ezkerreranzko gezia	Mugitu +X	Biratu +Y	Txikitu X
Eskuineranzko gezia	Mugitu -X	Biratu -Y	Handitu X
REPAG	Mugitu +Z	Biratu +Z	Txikitu Z
AVPAG	Mugitu -Z	Biratu -Z	Handitu Z

2. Taula: Tekla berezien taula

Beraz, automata honela geratuko litzaiguke:



1. Irudia: Egoera-diagrama

Transformazio-mota eta erreferentzia-sistema aldagai globalak *main.c*-n daude definituta eta hasieratuta translazio lokala burutzeko. Gainontzeko fitxategiek extern moduan erabili behar dituzte. Aldagai hauek hartu ditzaketen balioak *definizioak.h*-n daude definituta horrela:

```
#define TRANSLAZIOA      0
#define BIRAKETA        1
#define ESKALAKETA      2

#define LOKALA           0
#define GLOBALA         1
```

Orain, dena teorikoki definituta badaukagula, tekla bakoitza sakatzerakoan exekutatu behar den kodea bete behar da *io.c* fitxategian. 1. taulan jasotako teklek bakarrik esleitzen diote dagokion aldagaiari dagokion balioa, adibidez **m** teklaren kasuan:

```
case 'm':
case 'M':
    transformazio_mota = TRANSLAZIOA;
    break;
```

2. taulako tekla implementatzeko funtzio berri bat definitu behar dugu, orain arte erabilitakoak bakarrik ASCII karaktereak erregistratzen baititu (a-tik z-rako karaktereak CTRL-ekin sakatuz gero ere detektatzen ditu tekla desberdin moduan). Funtzio berri horri `specialKeyboard()` deitu diogu eta `glutSpecialFunc()`[1] funtzioaren callback funtzioa[2] izango da.

```
glutSpecialFunc(special_keyboard);
```

IGUAL HAU GOIKOA KENDU

`specialKeyboard()` funtzioak tekla arruntekin bezala funtzionatuko luke teklaren kodea karaktere batekin konparagarria izango baliteke. Arrazio hau dela-eta *definizioak.h*-n definitu ditugu tekla berezi horien balioak, horrela txukunago geratzen baita kodea.

1.2 Transformazioen desegitea

Aldaketak desegiteko *definizioak.h*-n `object3d` estrukturari atributu bat gehitu diogu: `transformazio_pila`. Pilari esker, aldaketak desegiteko funtzionalitatea errez egingo da. Pilaren erazagupena eta implementazioa *pila.h* eta *pila.c* fitxategietan egin da.

Beraz, aldaketak sartu ahala, pilari `push()` egingo diogu matrize berriekin. Edozein momentuan aldaketaren bat desegin nahi badugu, **CTRL+Z** sakatuz aurreko aldaketara pasatuko gara, eta hala egin dezakegu hasierako objektura heldu edo aldaketa berri bat egin arte.

2 Aldaketak

Aurreko faseko feedback-ean jaso bezala, orain sareta erreferentzia bezala erabiltzeko balio du. Aurreko fasean objektu bezala ageri zen pantailaren lehenengo koadrantean, orain aldiz, `viewport[3]` osoa zati uniformeetan banatzen du eta kameraren transformatua baino lehen marrazten denez, berdin dio kameraren posizioak edota zoom-ak.

3 Gehikuntzak

INFORMAZIOAREKIN BATERA, MOMENTUAN DAUKAN TRANSFORMAZIO-MATRIZEA PRINTEATU.

3.1 Aldaketak berregiteko aukera

CTRL+SHIFT+Z konbinaketa sakatzerakoan azken transformaziotik aurrera desegindako aldaketak berregin ditzake objektuak. Kontuan hartu transformazio bat desegiterakoan beste bat egiten bada, desegite hori galduko dela.

Pila implementatuta daukagun erak erraz ahalbidetu digu aldaketak berregitea. Bakarrik aurreratu behar dugu posizio bat matrizeen zerrenda estekatuan, egin ahal izanez gero.

Arazo posible bakarra memoriaren kutsadura da. Lehen esan bezala, desegite baten ostean transformazio berri bat eginez gero, aurretik pilaren gainean egon diren transformazioak galtzen dira. Hala ere ez dira memoriatik kentzen, beraz, memoria erabilezinaren parte osatzen dute, memoria kutsatuz. Oraingoz ez da arazo bat transformazioek memoria gutxi okupatzen baitute eta gure programak ez baitu memoriaren eskaera handirik burutu behar. Hala ere, etorkizunean arazoa izan liteke eta kontutuan hartu beharko genukeen arazo bat da.

3.2 Transformazio gehigarriak

References

- [1] [glutSpecialFunc\(\)](https://www.opengl.org/resources/libraries/glut/spec3/node54.html):
<https://www.opengl.org/resources/libraries/glut/spec3/node54.html>
- [2] [callback funtzioa](https://en.wikipedia.org/wiki/Callback_%28computer_programming%29):
https://en.wikipedia.org/wiki/Callback_%28computer_programming%29
- [3] [viewport](https://en.wikipedia.org/wiki/Viewport):
<https://en.wikipedia.org/wiki/Viewport>