

# **KONPUTAGAILU BIDEZKO GRAFIKOAK**

## **1. FASEA**

Xabier Lecumberri  
Iker Ortiz

### 1. Errorea

Azalpena:

Programa hasieratzean, TAB tekla ematean Segmentation Fault errorea jauritzen du.

Konponketa:

Arazoa io.c fitxategiko *keyboard()* funtzioan dago, 'case 9' kasuan (TAB tekla sakatzean, alegia).

Bertan, beti saiitzen zen kargatutako objektuen lista zeharkatzen. Hala ere, hasieran

*\_selected\_object* (egungo objektua) eta *\_first\_object* (kargatutako objektuen lehenengo elementua)

0-ra hasieratuta daude. Beraz, arazo hau konpontzeko baldintza bat jarri dugu *\_first\_object* '0'

balioa duenean; kasu horretan errore-mezu bat pantailaratuko du.

```
if (_first_object == 0){  
    printf("Ezin da aukeratutako objektua aldatu, oraindik ez  
    delako objekturik kargatu\n");  
} else {  
    ... kode originala ...  
}
```

### 2. Errorea

Azalpena:

Objeturik kargatuta ez badago eta <SUPR> tekla sakatzen badugu Segmentation Fault errorea jauritzen du.

Konponketa:

Arazoa io.c fitxategiko *keyboard()* funtzioan dago, 'case 127' kasuan (SUPR tekla sakatzean, alegia).

Bertan, beti saiitzen zen aukeratutako objektua ezabatzen. Objekturik kargatuta ez

dagoenenan *\_selected\_object* (egungo objektua) 0-ra hasieratuta dago, beraz, Segmentation Fault

konpontzenko hurrengo if-a jarri dugu:

```
if (_selected_object == 0){  
    printf("Ezin da objekturik ezabatu ez dagoelako bat ere ez  
    kargatuta\n");  
} else {  
    ... kode originala ...  
}
```

### 3. Errorea

Azalpena:

CTRL eta '-' bistaratze eremua handitu arren, CTRL eta '+' sakatzean eremua ez da txikitzen.

Konponketa:

Arazoa io.c fitxategiko *keyboard()* funtzioan dago, 'case +' kasuan (+ tekla eta CTRL tekla

sakatzean, alegia). Bertan, eremua txikitzea inplmentatu beharko izan dugu. Eremua handitzearen

kode berdina da; aldatzen den gauza bakarra da proiektzio plano aldatzeko, *KG\_STEP\_ZOOM*-ekin

zatitu beharrean, horrekin biderkatu beharko dugu:

```
/*Decrease the projection plane; compute the new dimensions*/  
wd=(_ortho_x_max-_ortho_x_min)*KG_STEP_ZOOM;  
he=(_ortho_y_max-_ortho_y_min)*KG_STEP_ZOOM;
```

#### 4. Atzeko planoaren kolorea:

Atzeko planoaren R, G, B eta  $\alpha$  kanalak definitions.h fitxategian daude, KG\_COL\_BACK\_R, KG\_COL\_BACK\_G, KG\_COL\_BACK\_B eta KG\_COL\_BACK\_A definizioetan gordetzen direnak (hurrenez hurren). Balio hauek main.c fitxategiko *initialization()* funtziotik irakurtzen da, lerro honetan:

```
glClearColor(KG_COL_BACK_R, KG_COL_BACK_G, KG_COL_BACK_B,  
KG_COL_BACK_A);
```

#### 5. Leihoaren izena

Leihoaren izena definitions.h fitxategian dago, KG\_WINDOW\_TITLE definizioan gordeta. Balio hau *main()* funtziotik deitzen da, lerro honetan:

```
glutCreateWindow(KG_WINDOW_TITLE);
```

#### 6. Leihoaren tamaina

Leihoaren dimentsioak definitions.h fitxategian daude, KG\_WINDOW\_WIDTH eta KG\_WINDOW\_HEIGHT (zabalera eta altuera hurrenez hurren) definizioetan gordeta. Balio hauek main.c fitxategiko *main()* funtziotik deitzen da, lerro honetan:

```
glutInitWindowSize(KG_WINDOW_WIDTH, KG_WINDOW_HEIGHT);
```

Balio hauek ere fitxategi berdineko *initialization()* funtziotik deitzen dira, lerro honetan:

```
_window_ratio = (GLdouble)KG_WINDOW_WIDTH / (GLdouble) KG_WINDOW_HEIGHT;
```

#### 7. Ardatzak idatzi

Ardatzak idazteko, definitons.h fitxategiko KG\_MAX\_DOUBLE definizioa 10E25 balioa 1000-rekin aldatu behar izan dugu.