

3. Fase - Kameraren kontrola

Borja Calvo

Abstract

Proiektuaren 3. fasean gure programari kameraren kontrola gehituko diogu. Orain arte objektuak soilik alda zitekeen, baina fase honetan kamera ere mugitzeko aukera izango dugu. Horrez gain, objektuak bistaratzean perspektiba proiektzioa erabiliko dugu. Dokumentu honetan helburu hori lortzeko hartu behar diren pauso guztiak aztertuko ditugu.

1 Helburua

Orain arte kamera ortografikoa besterik ez geneukan. Fase honetan ondoko bi kamera hauek gehitu beharko dituzue:

- **Perspektiba kamera.** Kamera honen proiektzioa perspektibakoa izango da. Horrez gain, kamera beste edozein objektu moduan trata daiteke, hots, mugitu eta biratu ahalko da, bere erreferentzi sisteman zein munduko erreferentzi sisteman.
- **Kamera ibiltaria** Kamera hau perspektiba proiektzioa erabili beharko du, baina bere funtzionamendua zertxobait ezberdina da. Helburua, zenbait bideojokotan erabiltzen diren kamerak imitatzea da. Horretarako, mugimenduak murriztuta egon behar dira. Kamera ibiltaria beti haren erreferentzi sisteman mugituko da, eta aukera bakarrak aurrera eta atzera mugitzea eta bere Y ardatzaren inguruan biratzea izango dira. Horrez gain, kamera bere X ardatzarekiko mugitzeko aukera izango dugu, *begirada* gora eta behera mugitzeko, baina mugimendu hori mugatuta egongo da, biraketa ezin baita izan $-90/90^\circ$ baino txikiagoa/handiagoa.

1.1 Kameraren kontrola: teoria

Aurreko fasean objektuen aldaketari ekin diogu; kamerari aldaketak eragiteko pausoak, kontzeptualki, berdinak dira. Izan ere, aldaketak kudeatzeko erabiltzen den estrategia eta matrizeak berdinak dira.

Alabaina, bi ezberdintasun izango ditugu. Alde batetik, uneko matrizea objektu erpinei aplikatu beharrean kameraren ezaugarriari aplikatuko diegu; ezaugarri horiek, gero ikusiko dugun bezala, kameraren kokapena, kamerak nora begiratzen duen eta *gora* adierazten duen bektoreak dira. Bestetik, kamera ibiltarian, aldaketa batzuk bakarrik aplikatu behar dira eta, horietako bat, begiradaren aldaketa, zertxobait ezberdina da.

2 Implementazioari buruz

Kameraren kontrola inplementatzeko hiru puntu aztertu behar dira: proiektzioa, kameraren kokapena eta aldaketen kudeaketa.



2.1 Perspektiba proiektzioa

Orain arte geneuka kamera (hau da, ortografikoa) mantenduko dugu. Beste bi kamerak perspektiba proiektzioa erabiliko dute. *OpenGL*n, ikusi dugun bezala, proiektzioa definitzeko `GL_PROJECTION` matrizea manipulatu behar dugu. Proiektzio ortografikoa definitzeko `glOrtho` funtzioa erabili dugu; perspektiba proiektzioa definitzeko, berriz, `gluPerspective` funtzioa erabili beharko duzue.¹

2.2 Kameraren kokapena

Aurreko fasean objektuen aldaketak sisteman sartzeko `GL_MODELVIEW` matrizea erabili behar dela ikusi dugu. Matrizearen izenak adierazten duen bezala, matrize honek ez du bakarrik objektuak (`MODEL`) kontrolatzen, baita kamera (`VIEW`) ere. Beraz, kameraren aldaketak, nolabait, matrize honetan sartu beharko ditugu.

Hau egiteko biderik errazena `gluLookAt` funtzioa erabiltzea da, funtzio horrek gure kamera kokatzeko erabil baitezakezu. Funtzioaren parametroak ulertzeko, bila ezazu interneten funtzioaren laguntza.

Kontutan hartu funtzioa kamera kokatzeko erabil dezakegu, eta honek `GL_MODELVIEW` matrizea aldatuko du. Beraz, *matrize hori identitate matrizearekin hasieratzen dugun bakoitzean, funtzioa erabili beharko duzue kamera kokatzeko.*

2.3 Matrizeen kudeaketa eta erabilera

Aldaketa-matrizeen kudeaketa aurreko fasean erabilitakoaren berbera da, baina ez matrizeen erabilera. Kamera sisteman sartzeko `gluLookAt` funtzioa erabiliko dugu. Bertan, kameraren zenbait parametro erabili behar dira; parametro horiek balio lehenetsiak izan arren, benetan erabili behar ditugun balioak ezberdinak izango dira, aldaketa matrizearen arabera, alegia.

¹Interneten funtzioaren laguntza bila ezazue funtzioaren parametroak ulertzeko eta nola definitzen diren jakiteko (kontuz `float`, `double` eta horrelakoekin ...).