

## Grafikkprogrammering – kamera

Denne uken skal du implementere perspektiver og en generell kameramodell.

### Lek med perspektiver

Les om ortho og perspective funksjonene til mat4 i glmatrix biblioteket.

Inkluder begge disse to metodene i web-applikasjonen. Send projeksjonsmatrisen til verteks-shaderen slik at alle vertekser blir transformert etter dette (du har sannsynligvis dette på plass allerede). Gi bruker mulighet til å manipulere de ulike parameterne til perspektiv og ortografisk projeksjon.

### Generell kameramodell

Opprett et JavaScript script for en kameramodell.

Les om lookAt funksjonen til mat4 i glmatrix biblioteket.

Implementer et objekt/konstruktør/funksjon for et generelt kamera som inneholder en view transformasjonsmatrise og som bruker lookAt funksjonen til å tilby generell funksjonalitet for et kamera. Det vil si, implementer metoder som er naturlige for et generelt kamera. Eksempler på slike metoder er:

- moveForward / moveBackward / moveLeft / moveRight / moveUp / moveDown
- setPosition / setLookDirection / setLookAt
- pitch / yaw
- rollLeft / rollRight

Test metodene med log til konsoll og deretter inkluder funksjonaliteten i webapplikasjonen. Transformasjonene til model og view må slås sammen før de sendes til verteks-shaderen (pga. hvordan transformasjoner for normalvektorer vil håndteres senere). Det vil si, multipliser view-matrisen med modell-matrisen og send dette resultatet som modelViewMatrix.

Map hendelser (events) som er naturlige for manipulasjon av kamera (f.eks. wasd for bevegelse, qe for roll etc.) til metodene til kameraobjektet.

Du skal nå ende opp med et UI der bruker kan enkelt bevege seg rundt i den virtuelle verden, samt å kunne transformere objektet som er lastet inn.