

Oppgave 3 Transformasjoner

I denne oppgaven skal vi se på transformasjoner. Anta du har en funksjon `drawObject()` som tegner ut et objekt som er inneholdt i en kube K med sidelengder 2 og sentrert i origo.

1. Angi en view-transformasjon i form av to elementære affine transformasjonsmatriser som setter kamera i posisjon $(5, 0, 0)$, innrettet i retning $(-1, 0, 0)$ og med oppvektor $(0, 1, 0)$.
2. Angi en model-transformasjon i form av en affin transformasjonsmatrise som speilvender med hensyn på planet definert ved $z = 0$.
3. Bestem et view frustum definert ved to punkter og ett plan som passer med konfigurasjonen over. View frustumet skal være minst mulig og slik at hele kubens K synes på skjermen.
4. Skisser OpenGL kode som tegner ut objektet som beskrevet i oppgavene over.

Oppgave 4 Projeksjoner

Anta at vi har en bounding-box B om et objekt, angitt ved minimumshjørne (l, b, n) og maksimumshjørne (r, t, f) .

1. Anta først en ortografisk kamera-modell. Angi en affin transformasjon bestående av en translasjon og en skalering som transformerer B til det kanoniske view-volumet $[-1, 1]^3$.
2. Utled den resulterende homogene transformasjonsmatrisen. Hvordan settes en slik projection-matrise enklest opp i OpenGL?
3. Hva er en (syntetisk) perspektiv kamera-modell (pin-hole camera)? Skriv ned et uttrykk for hvordan et punkt (x, y, z) projiseres til en skjerm med avstand d fra kameraet.
4. Anta en perspektiv kamera-modell og illustrer med figur hvordan view frustumet transformeres til det kanoniske view-volumet. Er dette en affin transformasjon?

Oppgave 5 Trekant-mesh

I denne oppgaven skal vi se på trekant-mesh.

1. Forklar hva mesh connectivity/topology er, og skisser hvordan dette kan implementeres med en trekant-basert datastruktur.
2. Hvordan beregner man en enhets normal-vektor til en trekant?
3. Gi eksempel på hvordan man kan beregne normalvektorer i vertexene i en triangulering.
4. Hva betyr det at et trekant-mesh har konsistent trekant-orientering og hvilken rolle spiller dette for beregning av normalvektorer.

(Fortsettes på side 3.)