UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i	INF3320 — Metoder i grafisk databehandling og diskret geometri
Eksamensdag:	3. desember 2010
Tid for eksamen:	14.30 – 18.30
Oppgavesettet er på 5 sider.	
Vedlegg:	Ingen
Tillatte hjelpemidler:	Ingen

Kontroller at oppgavesettet er komplett før du begynner å besvare spørsmålene.

Hver oppgave teller like mye

Oppgave 1 Data-grafikk

Forklar følgende begreper i datagrafikk-sammenheng:

1. Rasterisering.
2. Specular reflection (som komponent i OpenGL lys-modell).
3. Normal mapping.
4. Ambient occlusion.

Løsning.

1. *Rasterisering: prosessen som tar geometriske primitiver (punkter, linjer, trekanter) med assosierte dato som input og genererer fragmenter tilsvarende pixler på skjermen.*
2. *Specular reflection: modell for lysrefleksjon fra en (delvis) reflekterende flate. Andelen lys som reflekteres i en gitt retning er en funksjon av lysretning, synsretning og flatenormal.*
3. *Normal mapping: teknikk for skyggelegging basert på å slå normal-vektorer opp i en tekstur, istedenfor som vertex attributter.*
4. *Ambient occlusion: Teknikk for skyggelegging basert på å beregne hvor mye bakgrunnslys som treffer/reflekteres fra et punkt på en flate. Beregner en faktor for hvor stor del av en kule rundt punktet som er i skygge for objektet selv.*

(Fortsettes på side 2.)

Oppgave 2 Barysentriske koordinater

Vi skal se på en trekant $T = [\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]$.

1. Definer midpunktet til T ved hjelp av barysentriske koordinater. Hva er de barysentriske koordinatene til $\mathbf{v}_0$ og til midtpunktet på segmentet $[\mathbf{v}_0, \mathbf{v}_2]$?
2. Anta vi har fått oppgitt normal-vektorer $\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2$ i hjørnene til T. Uttrykk den lineære interpolanten $\mathbf{n}(\mathbf{x})$ til normal-vektorene ved hjelp av barysentriske koordinater.
3. Hva er den interpolerte normalvektoren $\mathbf{n}(\mathbf{x})$ for $\mathbf{x} = \mathbf{v}_0$? Kan vi anta at $\mathbf{n}(\mathbf{x})$ er en enhets-vektor for et vilkårlig punkt $\mathbf{x}$? Begrunn svaret.
4. Forklar hvordan man i OpenGL kan interpolere normaler over trekantene, dvs. pr fragment. Hvilke type variabler brukes?

Løsning.

1. *Midpunktet til T er gitt ved $\frac{1}{3}\mathbf{v}_0 + \frac{1}{3}\mathbf{v}_1 + \frac{1}{3}\mathbf{v}_2$. Midpunktet til segmentet er gitt ved koordinatene $(\frac{1}{2}, 0, \frac{1}{2})$*
2. $\mathbf{n}(\mathbf{x}) = \sum_{i=0}^{2} \alpha_i(\mathbf{x})\mathbf{n}_i$ *der $(\alpha_i(\mathbf{x}))$ er de barysentriske koordinatene til $\mathbf{x}$ mhp T.*
3. $\mathbf{n}(\mathbf{v}_0) = \mathbf{n}_0$. *Generelt er ikke $\mathbf{n}(\mathbf{x})$ en enhets-vektor fordi lengden til en sum av enhetsvektorer ikke nødvendigvis er lik 1.*
4. *En definerer normalvektorer som vertex attributter, og setter opp vertex- og fragment-shadere med normalvektorene som varying variable slik at de blir interpolert over primitiver.*

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 kuben K synes på skjermen.
4. Skisser OpenGL kode som tegner ut objektet som beskrevet i oppgavene over.

(Fortsettes på side 3.)

Løsning.

1. *Roter* -90 *grader om* y*-aksen, translater* -5 *enheter langs* x*-aksen:*

$$RT = \begin{bmatrix} \cos(-90) & 0 & \sin(-90) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-90) & 0 & \cos(-90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. *Skaler* (x, y, z) *med* $(1, 1, -1)$.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. *I kamera koordinater: View frustumet er bestemt av hjørnene* $(-1, -1, -4)$ *og* $(1, 1, -4)$, *far plan settes til -6.*
   I globale koordinater: View frustumet er bestemt av hjørnene $(1, -1, 1)$ *og* $(1, 1, -1)$, *far plan settes til -1.*

4. *Kode:*

```
// Perspective transformasjon
glMatrixMode(GL_PROJECTION);
glFrustum(-1.0,1.0,-1.0,1.0,4.0,6.0);

// View transformasjon
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glRotatef(-90,0.0,1.0,0.0);
glTranslatef(-5,0.0,0.0);

// Model transformasjon
glScalef(1.0,1.0,-1.0);

drawObject();
```

(Fortsettes på side 4.)

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).

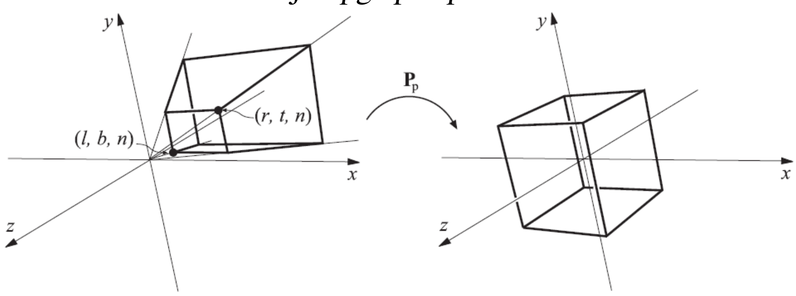
1. Anta først en ortografisk kamera-modell. Angi en affin transformasjon bestående av en translasjon og en skalering som transformerer B til det kanoniske view-volumet $[-1, 1]^3$.
2. Utled den resulterende homogene transformasjonsmatrisen. Hvordan settes en slik projection-matrise enklest opp i OpenGL?
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?

Løsning.

1. *Translater senter av kuben til origo, dvs med vektoren* $-(\frac{l+r}{2}, \frac{b+t}{2}, \frac{n+f}{2})$. *Deretter skalerer vi med faktorene* $(\frac{2}{r-l}, \frac{2}{t-b}, \frac{2}{f-n})$
2. *Kall* `glOrtho(l,r,b,t,n,f)`

$$ST = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{l+r}{2} \\ 0 & 1 & 0 & -\frac{b+t}{2} \\ 0 & 0 & 1 & -\frac{n+f}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{l+r}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{b+t}{t-b} \\ 0 & 0 & \frac{2}{f-n} & -\frac{n+f}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. *I en syntetisk perspektiv kamera-modell projiseres lys fra et objekt mot et punkt (camera center), og treffer en skjerm på veien.* $(x, y, z) \mapsto (dx/z, dy/z, d)$.
4. *Dette er ikke en affin transformasjon pga perspective division.*



(Fortsettes på side 5.)

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 normal-vektorer.

Løsning.

1. *Mesh connectivity/topology er naboskapsrelasjoner mellom noder, kanter og trekanter i et mesh. I en trekant-basert datastruktur er topologien påtrekant-nivå: hver trekant har en peker til hver av dens nabo-trekanter. I tillegg har hver node en peker til en trekant - en såkalt leading triangle.*
2. *Normal-vektor til en trekant* $T = [\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]$ *er gitt ved f.eks* $\tilde{\mathbf{n}} = (\mathbf{v}_1 - \mathbf{v}_0) \times (\mathbf{v}_2 - \mathbf{v}_0)$ *Denne vektoren kan eventuelt normaliseres:* $\mathbf{n} = \tilde{\mathbf{n}}/||\tilde{\mathbf{n}}||$
3. *Man kan f.eks for hver node: ta gjennomsnittet til alle trekanter som inneholder noden, evt normaler vektet med areal eller vinkel.*
4. *Det betyr at alle trekanter kan orienteres slik at par av nabotrekanter har samme orientering av nodene. To nabotrekanter har samme orientering hvis deres felles noder er nummerert i motsatt rekkfølge. Hvis trekantorienteringen ikke er konsistent vil trekantnormalene ikke være orientert på konsistent måte. F.eks kan to trekanter som ligger i samme plan ha normaler som peker i motsatt retning. Et gjennomsnitt basert på inkonsistente normaler gir ikke mening.*