UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i	INF 3320 — Metoder i grafisk databehandling og diskret geometri
Eksamensdag:	4. desember 2003
Tid for eksamen:	9.00 – 12.00
Oppgavesettet er på 5 sider.	
Vedlegg:	Ingen
Tillatte hjelpemidler:	Ingen

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

Oppgave 1 Skyggelegging

Anta at Phong-lysmodellen er i bruk og at en normalvektor er gitt i hver node av et polygon i 3D-rommet. Forklar forskjellen mellom flat, Gouraud, og Phong skyggelegging.

Løsning. I flat-shading får hele polygonet den samme fargen. Fargen er bestemt av Phong-lysmodellen basert på normalen til en vilkårlig node i polygonet.

I Gouraud-shading er Phong-modellen brukt til å bestemme fargen i hver node, basert på normalen der. Deretter er fargen i hvert punkt (hver pixel) av polygonet bestemt som en passende blanding av node-fargene.

I Phong-shading beregner man fargen fra Phong-modellen i hvert punkt (pixel) av polygonet. Normalen der er en passende blanding av node-normalene.

Oppgave 2 Viewing

Anta at kameraet ligger i vanlig posisjon, d.v.s., at det er sentrert i origo, og ser langs den negative z-aksen, og at 'opp'-retningen er y-aksen. Finn en rekke transformasjoner som setter kameraet slik at det er sentrert i punktet $(3, 2, 0)$ og ser i retningen $(1, 1, 0)$, og at 'opp'-retningen er z-aksen. Skriv en OpenGL sekvens som setter opp den assosierte model-view matrisen.

Løsning. En mulig rekke transformasjoner er $R_x(90)$, og så $R_z(-45)$, og så $T(3, 2, 0)$. Når vi setter model-view matrisen bruker vi den inverse transformasjonen:

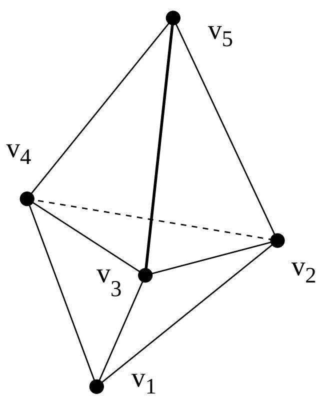
```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
```

(Fortsettes på side 2.)

```
glRotatef(-90, 1.0, 0.0, 0.0);
glRotatef(45, 0.0, 0.0, 1.0);
glTranslatef(-3.0, -2.0, -0.0);
```

Oppgave 3 Trekant-mesh

Anta at M er trekant-meshet i rommet i Figur 1. Meshet danner overflaten til et konvekst polyeder og har seks trekanter: $[v_1, v_2, v_3]$, $[v_1, v_3, v_4]$, $[v_5, v_2, v_3]$, $[v_5, v_3, v_4]$, $[v_1, v_2, v_4]$, $[v_5, v_2, v_4]$.



3a

Hva er Euler-karakteristikken χ til M?

Løsning. Euler-karakteristikken er

$$\chi = v - e + f.$$

I meshet M har vi $v = 5$, $e = 9$, og $f = 6$. Dermed er $\chi = 2$.

3b

Vil en half-edge kollapse fra (i) v_3 til v_5 eller (ii) v_3 til v_2, føre til et ikke-degenerert ('gyldig') trekant mesh?

Løsning. (i) En half-edge kollapse fra v_3 til v_5 er gyldig fordi de nye trekantene vil bli $[v_1, v_2, v_4]$, $[v_5, v_2, v_4]$ (som før) og $[v_1, v_2, v_5]$ og $[v_1, v_4, v_5]$ og de danner overflaten til tetrahederen $[v_1, v_2, v_4, v_5]$.
(ii) En half-edge kollapse fra v_3 til v_2 er ikke gyldig fordi de nye trekantene vil bli $[v_1, v_2, v_4]$ (to ganger) og $[v_5, v_2, v_4]$ (to ganger) og det nye meshet er degenerert.

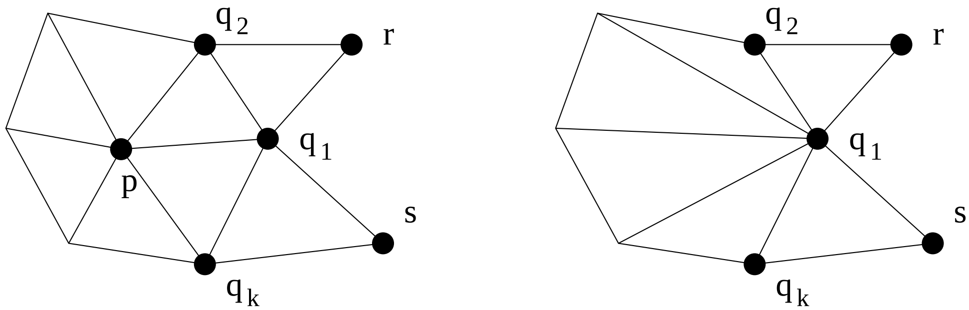
3c

Anta at et vilkårlig lukket trekant-mesh M er representert med en Indexed Face Set, utvidet slik at hver trekant har pekere til de tre nabotrekantene og hver node har en peker til en vilkårlig trekant som inneholder den.

(Fortsettes på side 3.)

Skisser hvordan du implementere en (lovlig) half-edge kollapse. Anta at noden er p og de k naboene til p er $q_1, q_2, \ldots, q_k$ i en rekkefølge rundt p og at kollapsen går fra p til q_1. Pass på at data-strukturen etter kollapsen er gyldig.

Løsning.



Med den gitte data-strukturen må vi fjerne p fra node-arrayen, og de to trekantene $[p, q_1, q_2]$ og $[p, q_1, q_k]$ fra trekant-arrayen. Og så skal trekanten $[p, q_i, q_{i+1}]$ endres til $[q_1, q_i, q_{i+1}]$, for all $i = 2, 3, \ldots, k-1$.

Og så er det pekerne. La $[r, q_1, q_2]$ være trekanten i M som deler kanten $[q_1, q_2]$ med $[p, q_1, q_2]$, og la $[s, q_1, q_k]$ være trekanten i M som deler kanten $[q_1, q_k]$ med $[p, q_1, q_k]$. I det nye meshet M' blir de to trekantene $[q_3, q_1, q_2]$ og $[q_1, q_2, r]$ naboer og må peke til hverandre. På samme måte blir nå de to trekantene $[q_1, q_{k-1}, q_k]$ og $[s, q_1, q_k]$ naboer og må peke til hverandre.

Til slutt er det trekant-pekerne til nodene q_1, q_2, og q_k som muligens må oppdateres. Hvis q_1 eller q_2 pekte til $[p, q_1, q_2]$ i M kan de nå endres til å peke til $[q_3, q_1, q_2]$. Hvis q_1 eller q_k pekte til $[p, q_k, q_1]$ i M kan de nå endres til å peke til $[q_{k-1}, q_k, q_1]$.

Oppgave 4 Bezier-kurver

Beskriv algoritmen til de Casteljau for en plan kubisk Bezier-kurve, gitt ved

$$P(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3, \qquad t \in [0,1].$$

Anta at kontroll-punktene er $P_0 = (1,0)$, $P_1 = (0,1)$, $P_2 = (1,1)$, $P_3 = (0,0)$. Hvis man representerer kurvestykket $Q(t) := P(t/2)$, $0 \le t \le 1$, som en Bezier-kurve hva er kontroll-punktene til Q?

Løsning. de-Casteljau algoritmen er en metode for å evaluere et punkt $P(t)$ på kurven, gitt en parameter-verdi t. Man setter $P_i^0 := P_i$ og så beregner man

$$P_i^1 := (1-t)P_i^0 + tP_{i+1}^0, \qquad i = 0, 1, 2,$$
$$P_i^2 := (1-t)P_i^1 + tP_{i+1}^1, \qquad i = 0, 1,$$
$$P_i^3 := (1-t)P_i^2 + tP_{i+1}^2, \qquad i = 0.$$

Man kan vise at $P(t) = P_0^3$.

(Fortsettes på side 4.)

Kontroll-punktene til

$$Q(t) = (1-t)^3 Q_0 + 3t(1-t)^2 Q_1 + 3t^2(1-t) Q_2 + t^3 Q_3,$$

kan hentes direkt fra de-Casteljau algoritmen:

$$Q_0 = P_0^0 = P_0,$$

$$Q_1 = P_0^1 = (P_0 + P_1)/2,$$
$$Q_2 = P_0^2 = (P_0 + 2P_1 + P_2)/4,$$
$$Q_3 = P_0^3 = (P_0 + 3P_1 + 3P_2 + P_3)/8.$$

En annen måte å finne disse er å utrykke $P(t/2)$ som en lineær kombinasjon av basisfunksjonene

$$(1-t)^3, \quad 3t(1-t)^2, \quad 3t^2(1-t), \quad t^3.$$

Oppgave 5 Koordinater og transformasjoner

5a

Skriv ned de tre 4×4 matrisene som beskriver (i) en translasjon i $\mathbb{R}^3$ langs en vilkårlig vektor i $\mathbb{R}^3$, (ii) en rotasjon i $\mathbb{R}^3$ rundt z-aksen, og (iii) en uniform skalering i $\mathbb{R}^3$.

Løsning. Translasjon langs vektor (a, b, c), rotasjon rundt z-aksen med vinkel θ, og uniform skalering med faktor λ er:

$$T(a,b,c) = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$S(\lambda) = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

5b

La $M_1, M_2, \ldots, M_n$ være en rekke transformasjoner hvor hver M_k enten er en rotasjon rundt z-aksen eller en vilkårlig translasjon i $\mathbb{R}^3$. Vis at

$$M_1 M_2 \cdots M_n = TR,$$

hvor T er en translasjon og R er en rotasjon rundt z-aksen.
Lykke til!

Løsning. Rekken kan ta for eksempel formen

$$R_1 R_2 T_1 R_3 T_2 R_4 T_3 T_4 T_5 \ldots \ldots$$

(Fortsettes på side 5.)

Siden

$$T(a,b,c)R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & a \\ \sin\theta & \cos\theta & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{1}$$

og

$$R_z(\theta)T(a,b,c) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & a\cos\theta - b\sin\theta \\ \sin\theta & \cos\theta & 0 & a\sin\theta + b\cos\theta \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{2}$$

har vi at

$$R_z(\theta)T(a,b,c) = T(a',b',c)R_z(\theta),$$

hvor

$$a' = a\cos\theta - b\sin\theta, \qquad b' = a\sin\theta + b\cos\theta.$$

Dermed kan vi rekursivt bytte hvert ordenede par $R_i T_j$ med et nytt par $T'_j R_i$. Når vi ikke kan bytte lenger har den nye rekken formen

$$T_1 \dots T_m R_1 \dots R_n.$$

Siden to translasjoner kan uttrykkes som en translasjon, og to rotasjoner rundt z-aksen kan uttrykkes som en rotasjon rundt z-aksen reduseres rekken videre til formen

$$TR.$$