Kommentar:
Svar kort og konsist. Husk at eksamen har tre oppgaver. Poengene for hver (del-) oppgave bør gi en indikasjon på hvor mye tid som bør benyttes per oppgave.

Oppgave 1: Forskjellige emner (40 poeng)

a. Hva er de inverse transformasjonene avfølgende tre transformasjoner T, R og S:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & \boldsymbol{t_x} \\ 0 & 1 & 0 & \boldsymbol{t_y} \\ 0 & 0 & 1 & \boldsymbol{t_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{R} = \begin{bmatrix} \cos(\boldsymbol{\theta}) & -\sin(\boldsymbol{\theta}) & 0 & 0 \\ \sin(\boldsymbol{\theta}) & \cos(\boldsymbol{\theta}) & 0 & 0 \\ 0 & & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{S} = \begin{bmatrix} \boldsymbol{s_x} & 0 & 0 & 0 \\ 0 & \boldsymbol{s_y} & 0 & 0 \\ 0 & 0 & \boldsymbol{s_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fasit:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & -\boldsymbol{t_x} \\ 0 & 1 & 0 & -\boldsymbol{t_y} \\ 0 & 0 & 1 & -\boldsymbol{t_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{R} = \begin{bmatrix} \cos(\boldsymbol{\theta}) & \sin(\boldsymbol{\theta}) & 0 & 0 \\ -\sin(\boldsymbol{\theta}) & \cos(\boldsymbol{\theta}) & 0 & 0 \\ 0 & & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{S} = \begin{bmatrix} \frac{1}{\boldsymbol{s_x}} & 0 & 0 & 0 \\ 0 & \frac{1}{\boldsymbol{s_y}} & 0 & 0 \\ 0 & 0 & \frac{1}{\boldsymbol{s_z}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b. Hvilke to egenskaper har affine transformasjoner? Er skjærtransformasjonen affin?

Fasit:
Affine transformasjoner har egenskapene at linjer som er parallelle før transformasjonen også er parallelle etter transformasjonen. I tillegg vil punkter som er endelige forbli endelige etter transformasjonen.

Skjærformasjonen er dermed affin.

c. Hva er grunnen til at man spesifiserer et klippevindu i visningsprosessen og hva skiller et klippevindu fra en *viewport*?

Fasit:
Man spesifiserer et klippevindu for å klippe bort objekter og primitiv som ikke skal vises på skjermen. Dermed slipper man unna med kun å lyssette og tegne overflaten til de delene av objekter som vises på skjermen, noe som er svært tidsbesparende siden lyssetting og overflategning kan være svært arbeidskrevende.

Mens klippevinduet spesifiserer hvilke komponenter skjermbildet skal bestå av, spesifiserer *viewport* hvor på skjermen skjermbildet skal vises.

d. Hvorfor klipper man ofte i normaliserte koordinater?

Fasit:
Man venter ofte med klippingen til etter normaliseringen fordi man da har muligheten til å slå sammen alle operasjoner som utføres før klippingen til en matrise. Jo færre matrisemultiplikasjoner som må utføres for hvert bilde som genereres, jo mer effektiv blir visningsprosessen.

e. Hva er stereoskopisk visning, og hvordan oppnår man stereoskopi i grafisk programvare?

Fasit:
Stereoskopi visning er å gi en illusjon av dybde ved å vise litt forskjellige todimensjonale bilder for hvert av øynene. Bildene må være av den samme scenen bare tatt med litt horisontal avstand. Man oppnår stereografisk effekt i grafisk programvare ved å generere et bilde til hvert øye med to forskjellige kameraer som er plassert litt i fra hverandre i horisontal retning.

f. Hvordan kan man finne ut om et punkt er foran eller bak en polygon som ligger i planet beskrevet av en implisitt ligning med parametrene A, B, C og D? Angi en formel og forklar de ulike parametrene.

Fasit:
Planlikningen kan uttrykkes ved hjelp av normalvektoren N og et punkt P:

$$\boldsymbol{N} \cdot \boldsymbol{P} = -\boldsymbol{D}$$

hvor normalvektoren, N, kan uttrykkes ved hjelp planparametrene A, B, og C:

$$\boldsymbol{N} = \left[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\right]$$

For å finne ut om punktet P' ligger foran eller bak et bestemt plan, regner man ut skalarproduktet mellom normalvektoren til planet, N, og punktet P'. Dersom $\boldsymbol{N} \cdot \boldsymbol{P}' < -\boldsymbol{D}$, befinner punktet P' seg bak planet og dersom $\boldsymbol{N} \cdot \boldsymbol{P}' > -\boldsymbol{D}$, befinner punktet P' seg foran planet.

g. Pikselfasing (pixel phasing) skiller seg grunnleggende fra de andre antialiaseringsmetodene som er beskrevet i boken. Hvordan?

Fasit:
I pikselfasing korrigerer man aliaseffekten ved å forskyve pikslene som vises på skjermen, mens intensiteten til pikslene ikke forandres. De andre teknikkene nevnt i boken, justerer intensiteten i stedet for plasseringen til pikselen.

h. Hva er sammenhengen mellom kartesiske og homogene koordinater.

Fasit:
$P_h = (x_h, y_h, z_h, w)$ er den homogene representasjonen av det kartesiske koordinaten $P = (x, y, z)$. Den homogene parameteren w må være ulik null. Vi har da at:

$$x = \frac{x_h}{w}, y = \frac{y_h}{w}, z = \frac{z_h}{w}$$

i. Nevn og forklar *kortfattet* tre problemer innen datagrafikk bruken av homogene koordinater løser.

Fasit:
Homogene koordinater muliggjør enhetlig behandling av transformasjoner da også translasjon kan utføres ved matrisemultiplikasjon. I tillegg unngås problemet med at vektorer og punkter blir representert på samme måte ettersom den homogene parameteren er lik null for vektorer og ulik null for punkter. En tredje fordel med homogene koordinater er at perspektivprojeksjon kan utføres som en matrisemultiplikasjon i det homogene rommet.

j. Kvaternionrepresentasjonen av punktet p er P = (0, p). Vis hvordan man kan rotere punktet p θ grader rundt linjen som går gjennom origo og kan representeres ved hjelp av kvaternionen q = (s, **v**). Angi formlene for s, **v** og P’, hvor P’ er kvaternionrepresentasjonen av punktet p’ som rotasjonen av P resulterer i.

Fasit:

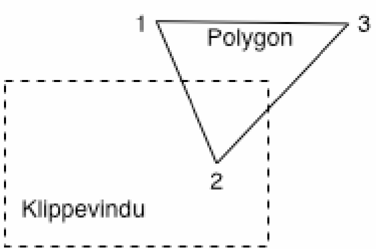
$$s = \cos\left(\frac{\theta}{2}\right),$$

$$v = u\sin\left(\frac{\theta}{2}\right),$$

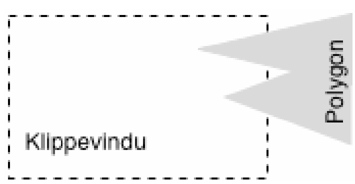
$$P' = qPq^{-1} = (0, p')$$

(Poengfordeling 4 poeng for hver deloppgave)

Oppgave 2: Klipping i to dimensjoner (35 poeng)



Figur 1



Figur 2

Ta utgangspunkt i klipping i to dimensjoner når du svarer på denne oppgaven. Det vil forenkle svaret ditt.

a. Hvorfor er ikke vanlige linjeklippingsalgoritmer tilstrekkelig ved klipping av polygoner?

Fasit:
Klipping av en polygon med en linjeklippingsalgoritme kan resultere i at det klippede polygon ikke lenger er et polygon, men en samling polylinjer som ikke henger sammen. Dette vil vanskeliggjøre fargelegging av polygonene som skal vises ettersom flatene som skal fargelegges ikke lengre har noen klart definerte grenser.

b. Forklar algoritmen for klipping av polygoner utarbeidet av Sutherland og Hodgeman. *Skriv gjerne pseudokode i stedet for en forklaring om du føler det er lettere. Pseudokode er ikke et krav og vil ikke belønnes i større grad enn en tilsvarende god forklaring.* Bruk algoritmen på situasjonen beskrevet i figur 1, og vis de ulike stegene.

Fasit:
Sutherland-Hodgmans linjeklippingsalgoritme tar utgangspunkt i en samlebåndsarkitektur med fire steg, hvor hvert av stegene utfører klipping mot ett av klippevinduets kanter. Hvert steg behandler en kant om gangen. Klippealgoritmen blir matet med en liste av punkter, og resultatet etter klipping vil være en liste av punkter.

For hver kant som blir matet til ett av stegene blir utkommet basert på resultatet av de fire testene under:

1. Dersom første punkt i kanten er utenfor klippevinduet og andre punkt er innenfor, vil skjæringspunktet mellom kanten og klippevinduet og punkt to sendes videre.
2. Dersom begge punktene er på innsiden av klippevinduet sendes kun det siste av punktene videre.
3. Dersom første punkt befinner seg på innsiden av klippevinduet og andre punkt befinner seg på utsiden, sendes kun skjæringspunktet videre.
4. Dersom begge punktene i kanten ligger på utsiden av klippevinduet, sendes ingen punkter videre.

Den siste klipperen i samlebåndet genererer punktlisten for det klippede polygonen.

Figur 1 brukt på algoritmen
Kantene blir matet i følgende rekke følge (1, 2), (2, 3) og (3, 1) til et samlebånd som klipper i følgende rekkefølge: Venstre klipper (steg 1), høyre klipper (steg 2), bunnklipper (steg 3), toppklipper (steg 4).

Venstre:
(1,2) => (inne, inne) gir følgende resultat => (2)
(2,3) => (inne, inne) gir følgende resultat => (3)
(3,1) => (inne, inne) gir følgende resultat => (1)
Høyre:
(2,3) => (inne, ute) gir følgende resultat => (2')
(3,1) => (ute, inne) gir følgende resultat => (3', 1)
(1,2) => (inne, inne) gir følgende resultat => (2)

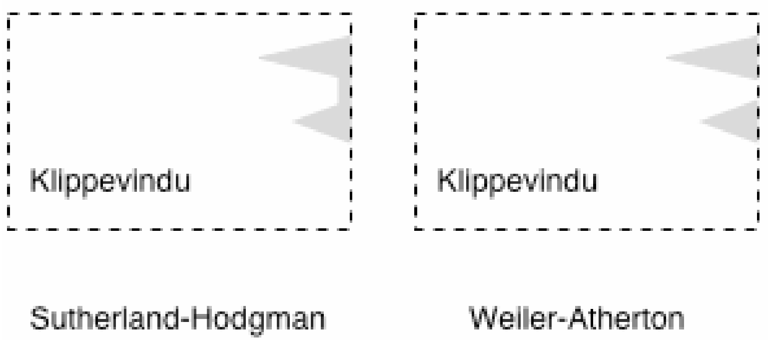
Bunn:
(2',3') => (inne, inne) gir følgende resultat => (3')
(3',1) => (inne, inne) gir følgende resultat => (1)
(1,2) => (inne, inne) gir følgende resultat => (2)
(2,2') => (inne, inne) gir følgende resultat => (2')

Topp:
(3',1) => (ute, ute) gir følgende resultat =>
(1,2) => (ute, inne) gir følgende resultat => (1',2)
(2,2') => (inne, inne) gir følgende resultat => (2')
(2', 3') => (inne, ute) gir følgende resultat => (3'')

Resultat:
(1', 2, 2', 3''), hvor 1' er skjæringspunktet mellom klipplinjen topp og kanten (1,2), 2' er skjæringspunktet mellom kanten (2,3) og høyre klipplinje og 3' er skjæringspunktet mellom kanten (3,1) og de to klipplinjene høyre og topp.

c. Hva blir resultatet om standardversjonen av Sutherland-Hodgeman-algoritmen, som er beskrevet i boken, blir benyttet for å klippe polygonen med klippevinduet som er illustrert i *figur 2*? Enn om man benytter Weiler-Atherton-algoritmen? Illustrer resultatet med to figurer, en for hver algoritme.

Fasit:



Klipping av e konkav polygon med Sutherland-Hodgman-algoritmen beskrevet i boken vil kunne resultere i en polygon når det egentlig skal klippes i to (eller flere biter). Slike artefakter vil ikke oppstå med algoritmen utarbeidet av Weiler og Atherton. W-A vil gi et visuelt korrekt resultat. Effekten av S-H er overdrevet for å tydeliggjøre poenget.

d. Sutherland-Hodgman kan benyttes til å klippe mot et hvilket som helst konvekst, polygonformet klippevindu. Hvorfor foretrekkes å bruke et rektangulært klippevindu i sammenheng med denne algoritmen.

Fasit:
Ved å la kantene til klippevinduet være parallelle med aksene til visningskoordinatsystemet kan klippeprosessen forenkles til å kun bestå av tester om x-, y-, og z- verdiene ligger mellom null og en. Om det er tilfellet beholdes punktet, ellers klippes det bort.

Dersom kantene i klippevinduet ikke ligger parallelt med aksene i koordinatsystemet (noe som vil være tilfellet for minst en av sidene om klippevinduet ikke rektangulært) blir man nødt til å sjekke om hvordan et punkt forholder seg til en linje som beskrives av en funksjon i stedet for en konstant. En slik test er mer krevende.

(Poengfordeling: 6, 15, 6, 8)

Oppgave 3: Projeksjon (25 poeng)

Grafikkpakker tar som regel utgangspunkt i en vektor V_{skjev} for å spesifisere en skjev parallellprojeksjon av et punkt P i projeksjonsplanet. La oss kalle projeksjonen av P for P_{proj}.

a. Hva uttrykker vektoren V_{skjev}?

   Fasit:
   V_{skjev} uttrykker retningen til den skjeve parallellprojeksjonen relativt projeksjonsplanet.

b. Hva mangler i beskrivelsen over for å definere projeksjonen fullstendig?

   Fasit:
   Plasseringen av projeksjonsplanet er ikke spesifisert.

c. Uttrykk komponentene til P_{proj} ved hjelp av komponentene til P og V_{skjev}. Benytt så uttrykkene du har utledet til å sette opp en transformasjonsmatrise for skjev parallellprojeksjon. Illustrer oppsettet slik at betydningen til alle variabler kommer frem.

   Fasit:
   figur

Vi plasserer projeksjonsplanet vinkelrett på z-aksen i visningskoordinatsystemet, slik at det planet er definert ved at z-koordinaten til alle punkter i planet har verdien z_{pp}. Og vi definerer $V_{skjev} = (V_x, V_y, V_z)$.

Vi tar utgangspunkt i figuren over. Ved sammenligning av like triangler kan vi sette opp følgende ligninger:

$$\frac{x_p - x}{z_{pp} - z} = \frac{V_x}{V_z}$$

$$\frac{y_p - y}{z_{pp} - z} = \frac{V_y}{V_z}$$

Uttrykker så projeksjonskoordinatene ved hjelp av projeksjonsvektoren:

$$x_p = x + (z_{pp} - z)\frac{V_x}{V_z}$$

$$y_p = y + (z_{pp} - z)\frac{V_y}{V_z}$$

Ligningene over uttrykkes ved hjelp av en matrise:

$$M_{skjev} = \begin{bmatrix} 1 & 0 & -\frac{V_x}{V_z} & z_{vp}\frac{V_x}{V_z} \\ 0 & 1 & -\frac{V_y}{V_z} & z_{vp}\frac{V_x}{V_z} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(Poengfordeling: 5, 5, 15)