



1 a)

A = USANT	G = USant
B = Sant (sant)	H = <del>USant</del> Sant USANT
C = Ja (sant)	I = USant
D =	J = Sant
E = Ja (sant)	K =
F = USant	L = USant

2 a)

1. Beskriv en translasjon i  $y$  på  $x$ ,  $z$  på  $z$  og  $z$  i  $z$  retning.

2. Beskriv en rotasjon om  $z$  akse med vinkelen på  $90$  grader

3. Beskriv en Homogen matrise (som resten) med skaleringen  $6$  i alle "aksen". med det Homogene koordinatet på  $2$ . Skaleringen vil være  $3$  uten det homogene koordinatet

$$x = x_h/h \quad y = y_h/h \quad z = z_h/h$$

b)

#: Vi har 1. Rotasjon 2. Skalering 3. Translasjon som Basistransformasjoner.



2 c

Generelt sett kan vi si at vi roterer rundt den akse som står still under rotasjonen. Dette ser vi også i rotasjonsmatrisene der:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotasjon om Z-aksen (Z står stille)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotasjon om X-aksen (X står stille)

Dette skjer også for Y-aksen.

Derfor kan vi lett gjennkjenne en rotasjon v.h.f. i matrisene.

3 a

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ m \\ Y \end{bmatrix}$$

Om jeg har forstått oppgaven riktig vil den fram til formelen der jeg dytter om C og m (da også R og G)

I såfall får vi:

1. Grønn
2. Rødt
3. Blått (ingen forandring) Her
4. Magenta
5. Cyan
6. Yellow
7. Black (k)
8. Hvit

I dette denne problemstillingen

Det kan tenkes at jeg misforstår Hver som skjer med f.eks. Blått. Men om denne da er uavhengig av C eller m blir denne også godt fremstilt. Generelt sett blir alle farger som er uavhengig av C/m - Galt



Emnekode : DAt 200  
Kandidatnr. : 358  
Dato : 4/12-13  
Ark nr. : 3 av 12

3 B)

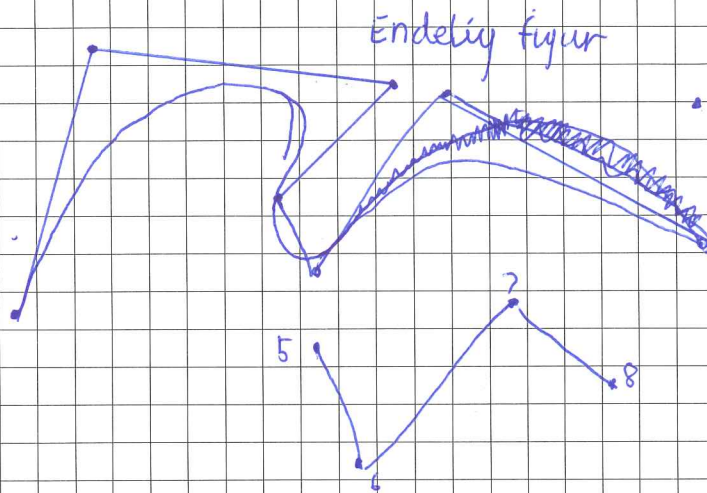
Z-Buffer <sup>har</sup> ~~er~~ ~~en~~ dybdebufferens. Algoritmen baseret  
seg på at vi har Z-Buffer kaldt Ramme-Buffer  
og Dybdebuffer. Dybdebufferet holder på dybde information  
om hver piksel mens rammebufferet har det endelige  
Billedet (i piksler). Når dybdebufferet opdateres  
bliver ny data lagret i Ramme Bufferet (overskrives  
"gammel" piksel information). Det er her Z-Buffer  
kommer til kort, dette på at eben du ikke kan  
repræsentere transparente billeder korrekt.  
Lucas Arts har imidlertid forbedret dette ved  
A-Buffer. Z-Buffer er en Billederens algoritme.

3 C)

1. Konstant intensitet
2. Phong shading
3. Gouraud shading



4a)



Siden jeg tydeligvis ikke er en abstrakt lepper jeg ved forklaring :  
Ud fra det første kurveregmentet retter vi på en ny kurve  
denne kurven skal ha en unik tangent uterom i skjærpunktet  
kurvene skal også være sammenhengende.

4B)

a) Kontinuitet krever at punktene er på samme Bunket (co)

Samt at tangenten er lik i Begge kurvene (for de deraste  
må være lik).

b) Kontinuitet må ikke forstedemeste være lik over  
hele kurven, men lik i skjærpunktet.

c)

D er definet fra 2 til  $n+1$  hvor  $P_k$  er et Bunket,  $P_{k+1}$

er neste Punkt D er da antall Punkt så  $D P_k$  er

fels  $2 P_k$  . mens du det siste Punktet er  $D P_{k+1}$

Så de 2 første Punktene er Bepynnelsen man

de to neste er Slutten på kurven . Vi tar også

Inversen av dette

NESTE Side



4c

Vi kan tenke oss Hermitiske kurver som en Serie Samling av forskjellige tangenter. Det er vesl å sette en tangent på enden at vi utvikler kurvesegmentet.

4d.

Her kan vi se et typisk tilfelle hvor anti-aliasing blir Brukt for å rette opp pikselerte kurver.

Anti-aliasing brukes mye innenfor ~~3D~~ Grafikk for å gjøre et dårlig resultat "Bra", det krever dessverre mye maskinerkraft.

4e.)

Isometrisk projeksjon tilhører Parallelprojeksjon.

Denne kjennetegnes at avstanden fra origo er lik fra alle Sider av figuren Dette gir oss derfor et "overblikk" av figuren fra alle ~~3D~~ sider. Isometrisk blir Brukt ofte i strategi spill som Age of Empires 2 eller Starcraft 1.



Emnekode : BAT 200  
Kandidatnr. : 358  
Dato : 4/12-13  
Ark nr. : 6 av 12

4F

$$X = X_2(1-P) + X_1P$$

$$Y = Y_2(1-P) + Y_1P$$

$$Z = Z_2(1-P) + Z_1P$$

Setter Z til 0

$$0 = -50(1-P) + 50P$$

$$= -50 + 50P + 50P$$

$$\frac{100P}{100} = \frac{50}{100}$$

$$P = \frac{1}{2}$$

$$X = 0 + 20 \cdot \frac{1}{2}$$

$$Y = 0 + 40 \cdot \frac{1}{2}$$

$$X = 10$$

$$Y = 20$$

$$P = \frac{1}{2}$$

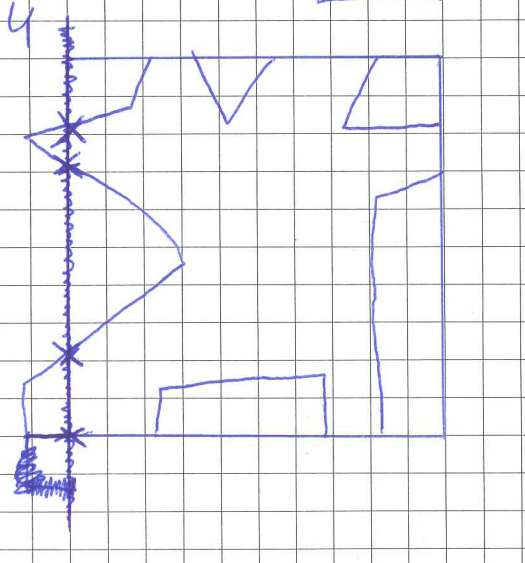
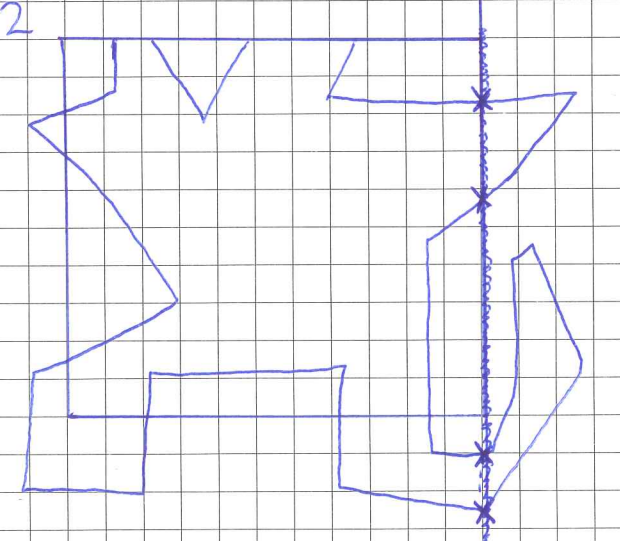
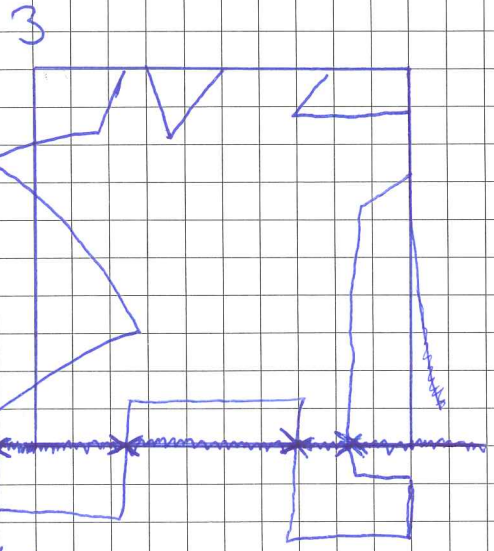
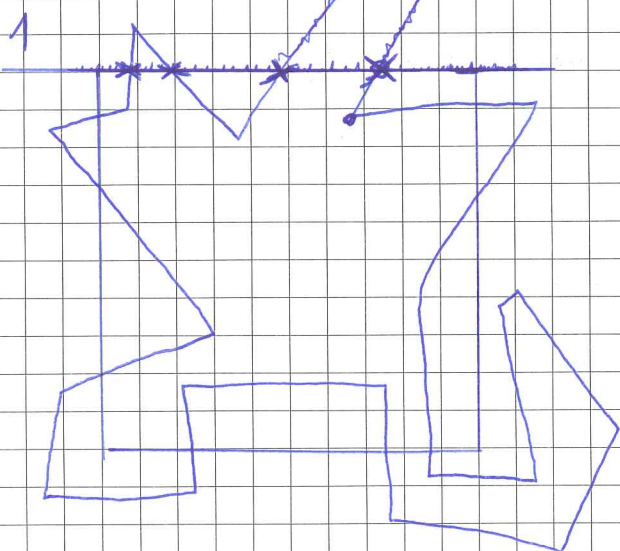
Svar: Svar er  $X = 10$   
 $Y = 20$   
 $P = \frac{1}{2}$



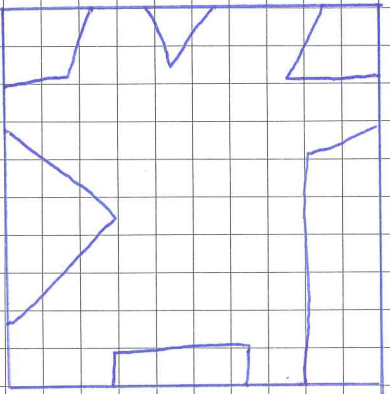
Emnekode : DA2200  
Kandidatnr. : 358  
Dato : 4/12-13  
Ark nr. : 7 av 12

49

1 velger en side og går mellom punktene til den sjøen "Lungen"



Result



Neste Side



Emnekode : DAT 200  
Kandidatnr. : 358  
Dato : 4/12-13  
Ark nr. : 8 av 12

4c

Algoritmen vi ser på forrige side er Sutherland Hough.  
Og er en algoritme som baserer seg på "split og hersk"  
med dette mens at vi tar et "stort" problem og gjør det til  
4 små

5a

Her ville jeg startet med å lage 1 sferer som skal være passende  
størrelse. Jeg ville reetter lage en sirkel (uten masse) og  
duplikert sferen rundt denne. Så langt er kulen på plass.  
Jeg ville deretter lage 4 "smulke" og gjør dem "firkantet"  
~~Zou da~~ Den ene skal være minste mens den andre ypperste  
Jeg setter begge for til passende størrelse og duplikert  
Jeg translere så de duplikerte til rett høyde.  
(Jeg nevner ikke translere da dette er en sekvens.)  
Når dette er på plass er den ferdig og vi kan  
da legge på destruktorer som teksturer osv.  
Stolper som fester sammen sammen ville jeg  
også satt på.





Emnekode : DAT 200  
Kandidatnr. : 358  
Dato : 4/12-13  
Ark nr. : 9 av 12

6a1

1. Legger til mouslutener fra implementert interface.  
(implementerte mouselutener).

Dette er metoden jeg foretrekker på grunn av at f.eks. `click`  
 2 ikke er i global scope. Ulemper med metoden er at du  
 blir tvungen til å implementere alle funksjonene  
 i interfacet her.

2. Denne ligger et nytt objekt av mouse adapter og du  
 vil her måtte følge til adapteret utentia. Fordelen  
 her er at du slipper å implementere funksjoner  
 du ikke trenger.

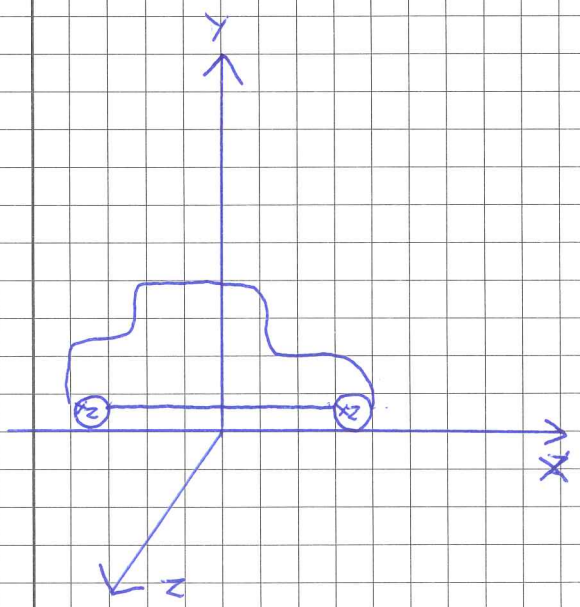
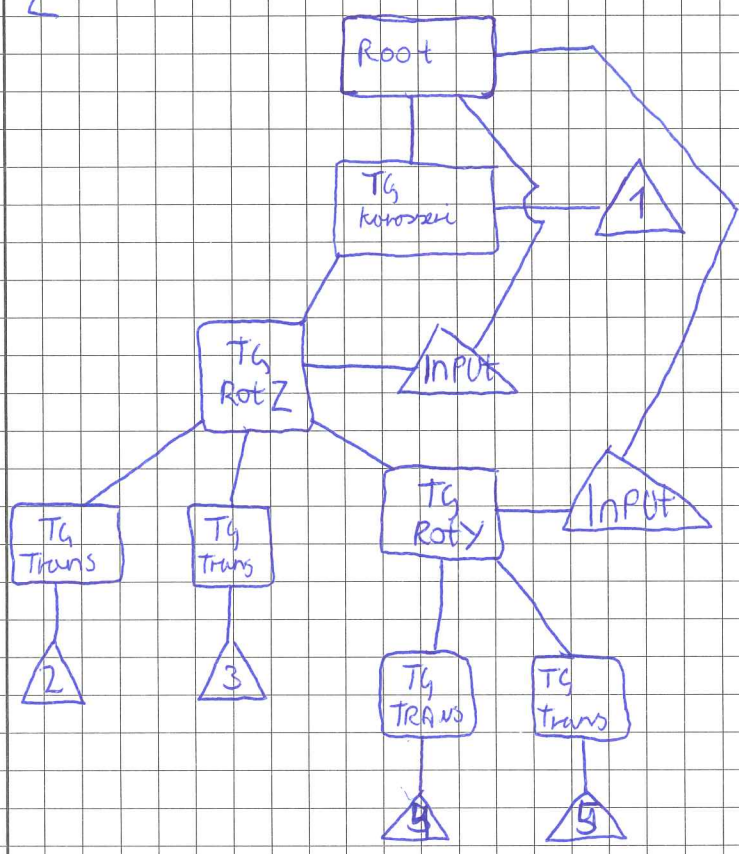
3. Her implementerer vi mouse objekt som er en  
 klasse med mouselutener implementert.  
 mye likt valg 1, men her har ikke klassen kontroll  
 over implementasjonen som ligger i mouse objekt.

Av de 3 midlene er selvfølgelig sett metode 1 best  
 i de generelle tilfellene Pga mer kontroll over  
 implementasjonen.



Emnekode : BAT20  
Kandidatnr. : 358  
Dato : 4/12-13  
Ark nr. : 10 av 12

6 B 2





6 b

1

Tg = Transformgroup

Public class Bil

Public Branch group createScene (float f)

{

Tg Tg Hull-1 new Tg ();  
Tg Tg Hull-2 new Tg ();  
Tg Tg Hull-3 new Tg ();  
Tg Tg Hull-4 new Tg ();  
Tg Tg kross new Tg ();

// Load krossen with Inspector 3D # 1  
objRoot.addChild ( loader.getmodel ();

Tg . RotZ = new Tg (); // Wheel Rotate group

// Load model for each of the wheels # 2  
and add to transformgroup  
Tg.hull.addChild ( loader.getmodel ();

// add all off the wheels to the RotZ Tg.

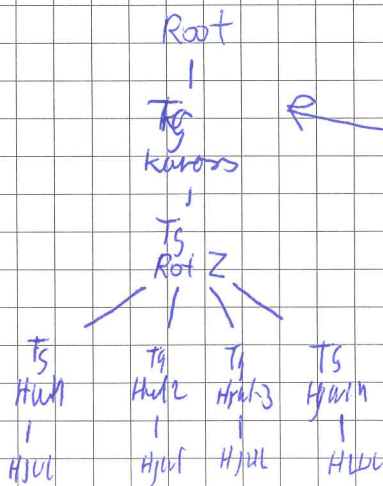
// Translate each of the wheels Tg to the correct  
// positions.

// Return tree  
return objRoot;

}

Treat som vest operator Bil

NB !  
Utbedring pa neste side  
(↑)  
(markert med)  
#



NB  
lick 3 oppg  
2



Emnekode : DAT6 200  
Kandidatnr. : 358  
Dato : 4/12-13  
Ark nr. : 12 av 12

6b 1

Utbedring:

Innlasting av modellen #1

```
Inspector3DS loader = new Inspector3DS("c:/karakterer/3ds");
```

```
loader.Parsert();
```

```
Tg karakter = loader.getModel();
```

```
objRoot.addChild(karakter);
```

Innlasting av Hjul #2

```
Inspector3DS loader2 = new Inspector3DS("c:/Hjul/3ds");
```

```
loader2.Parsert();
```

```
Tg Hjul1 = loader2.getModel();
```

```
TgHjul1.addChild(Hjul1);
```

Vi gjør dette for alle Hjul. Etter dette må jeg ta med på Plumm  
Hjulene via TgHjul1-4 objektet. Jeg henger de så  
på Roteringsgruppen og henger til Slutt på  
Roteringsgruppen på kartet

Jeg kan sende om karakterer være origo.