Institutionen för teknisk databehandling
Bo Nordin (bosse@cb.uu.se, 183470) 941220

Lösningsförslag till
Tentamen i datorgrafik MN1, 941220

---

1. Tre steg a) translation till (0,0)

$$\text{Matris } T(-2,-2): \begin{matrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{matrix}$$

b) rotation 90 grader

$$\text{Matris } R(90): \begin{matrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{matrix}$$

c) translation till (6,2)

$$\text{Matris } T(6,2): \begin{matrix} 1 & 0 & 6 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{matrix}$$

$$\text{Sammanlagd matris } T(6,2)R(90)T(-2,-2): \begin{matrix} 0 & -1 & 8 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{matrix}$$

OBS: Förutsätter postmultiplikation av kolumnvektorer. Om man i stället vill premultiplicera radvektorer blir matriserna transponerade.

---

2. server - ett program som kontrollerar displayen, har hand om display-resurser, etc.

Motif - widget toolkit som har menyer, knappar, etc.

widget - en gränssnitts-byggsten (med ett fönster) för toolkits (ex. pushbutton, drawingarea, list).

callback - en funktion (skriven av applikationsprogrammeraren) som anropas som resultat av en händelse. Programmeren "kopplar ihop" callbacks med händelser genom att sätta widget-resurser.

---

3. Rastrering i detta sammanhang innebär att linjer, cirklar, etc, översätts från en beskrivning i matematiska termer (t.ex linje från P_1 till P_2 eller cirkel med centrum i P_1 och radien r) till pixlar. Rastrering måste göras eftersom (de flesta) skärmar och hardcopy-enheter baseras på rasterteknik medan de grafiska primitiverna vanligtvis anges i matematiska termer.

Midpoint line algorithm: Beskriver här för linjer med lutning mellan 0 och 1. Använder integeraritmetik för att från nuvarande punkt välja endera av punkten E (österut) eller NE (nordost). Man kontrollerar på vilken sida av mittpukten (M) mellan E och NE som linjen går. Metoden använder sig av inkrementella skillnader som är beräknade från $F(x,y) = dy*x - dx*y + B*dx = 0$ (B är linjens skärningspunkt med y-axeln).
(*se vidare sidorna 73-75 i Foley et al*)

punktordning: har betydelse endast när linjen går exakt genom M. Man erhåller symmetry om man väljer E i det fall att linjen går från vänster till höger men SW (sydväst) om linjen går i andra riktningen (*se vidare sidan 78 i Foley et al*)

klippning: Fungerar om man ser till att beslutsvariabeln initieras korrekt (d.v.s. sätts till det värde den skulle haft om hela linjen hade ritats). Man måste se till att den klippta linjen får samma lutning som den oklippta. Problemet är mer komplicerat om linjen skär en horisontell klippkant än om en vertikal skärs (*se vidare sidan 79 i Foley et al*).

---

4. **Cohen-Sutherland**
1. för trivial reject/accept, förläng klippkanterna och inför koder enlig (t ex)::
bit 1 - ovanför övre kanten
bit 2 - nedanför nedre kanten
bit 3 - till höger om den högra kanten
bit 4 - till vänster om den vänstra kanten

1001 1000 1010
0001 0000 0010
0101 0100 0110

om *and* mellan koderna för linjens två ändpunkter är skilt från 0 kan linjen förkastas.
om koderna för ändpunkterna båda är 0 accepteras linjen.
om inget av ovanstånde: beräkna skärningen mellan aktuell klippkant och linjen.

(*se vidare sidorna 103-107 i Foley et al*, eller om du hellre vill beskriva "**Cyrus-Beck parametriska linjeklippningsalgoritm**" på *sidorna 107-111, se dessa*)

---

5. Det parametriska är hur kurvorna representeras: x=x(t), y=y(t) och z=z(t) där parametern t varierar från (normalt) 0 till 1. Tredjegradskurvor används eftersom de ger "lämplig" flexibilitet: lägre gradtal ger inte möjlighet att interpolera (passera genom) två punkter med specificerad derivata i punkterna; högre gradtal kräver fler villkor och resulterar lätt i okontrollerbara oscillationer. Med tredjegradskurovr har man har fyra obekanta per dimension. I fallet Bezierkurvor bestäms dessa genom specifikation av ändpunkter plus derivatan (genom vektorerna P_2-P_1 respektive P_4-P_3) om punkterna anges som P_i med P_1 och P_4 för de två ändpunkterna).

(*se vidare sidorna 328-329 och 336-339 i Foley et al*)

---

6. Färgtabeller används, d.v.s. ett färgindex (i intervallet 0-255) pekar ut en RGB-triplett. Man kan visa 256 färger samtidigt med denna uppställning. Anledningen till att man kan ändra färg dynamiskt är helt enkelt att det räcker med att skriva om färgtabellen (ändra RGB-tripletten för de index man vill ändra) medan man i 24-bitars-fallet måste skriva om själva bilden.

RGB är ett additivt system medan CMY är ett subtraktivt. I CMY specificeras färger med vad som dras från vitt medan i RGB specificeras färger med van som adderas till svart.

$$\begin{matrix} C \\ M = \\ Y \end{matrix} \begin{matrix} 1 - R \\ 1 - G \\ 1 - B \end{matrix}$$

---

7. Använd liksidiga trianglar i figuren:
xp/d = x/z Multiplicera med d: xp = (d*x)/z. = x/(z/d)

$$\text{Matrisen} \quad \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{matrix} \quad \text{används för att transformera}$$

punkten (x y z 1) till (x y z z/d).
För att få fram önskad punkt måste man dividera med z/d: (x/(z/d) y/(z/d) d)
Perspektivtransformationen är egentligen bara denna division.

(*se vidare sidorna 213-214 i Foley et al*)

---

8. **Bounding-box testing**: Ett enkelt test på min- och maxvärden i de olika riktningarna för att välja bort polygoner som är helt utanför klippfönstret eller snabbt se att två polygoner ligger helt bredvid varandra (se vidare sidorna 446-447 i Foley et al).

Backface culling: borttagning av ytor som är frånvända, görs enklast genom att testa tecknet på z-komponenten av ytnormaler (negativt tecken medför att polygonen är frånvänd). Fungerar om kanterna specficerats moturs (se vidare sidorna 448-449 i Foley et al).

Warnock: Delar successivt upp skärmen (eller snarare fönstret) i mindre och mindre kvadrater. Vid varje steg har varje polygon ett av fyra relationer till aktuell kvadrat:
1/ omgivande (finns över hela ytan)
2/ skärande
3/ inuti ytan
4/ helt bredvid
I fyra fall behöver ytan inte delas upp i mindre ytor:
1/ alla polygoner utanför
2/ Bara en skärande eller innesluten polygon.
3/ En enda omgivande polyogon
4/ Mer än en skärande, innesluten eller omgivande men en omgivande framför alla andra.

Man fortsätter till pixelupplösning om det behövs. Vid denna upplösning (om inget av 1-4 ovan uppfyllt) tar man t.ex. den närmaste polygonen eller också kan man använda flera polygoner (alla som "syns" i pixeln) som för någon form av antialiasing.

(*se vidare sidorna 469-471 i Foley et al*)

---

9. Riktningen från betraktaren till punkten (4,1,4) är **e** = (-3, -3, -4) / sqrt(3*3+3*3+4*4).
Riktningen från ljuskällan dit är **s** = (11, 5, 10) / sqrt(11*11+5*5+10*10).
Ytnormalens riktning i punkten är givetvis **n** = (0, 0, 1).
Den speglade ljusstrålen för riktningen **s1** = **s** med z-koordinaten omvänd.
Avståndet från ljuskällan till punkten är **d** = sqrt(11*11+5*5+10*10).

Ljusstyrka = $I_b R_b + I_d (R_d \langle s1,n \rangle + R_s (\langle -s1,e \rangle)^{50}) / d =$

$5 * 0.5 + 0.5 * (0.8 * 10 / \text{sqrt}(246) + 0.7 * (88/(\text{sqrt}(246)*\text{sqrt}(34)))^{50}) / \text{sqrt}(246)$

(*se vidare sidorna 478-485 i Foley et al*)

---

10. Transformationer, rastrering och display kan med fördel läggas i hårdvara. De två första stegen (generering och traversering) bör göras i värddatorn om man skall kunna ändra på modellen. Är ändringar ovanliga kan också traversering ske i special-processorer (såsom ibland görs i demonstrationsprogram för att imponera på oss dumma åskådare).