

Hemtentamen 980814
t980814a
Programmeringsmetodik DV2, 4/5p

Cons T Åhs
Institutionen för Datalogi
Uppsala Universitet

19 oktober 1998

Administrivia

Inlämning

- Tentamen skall lämnas in fredag 14/8, under tiden kl. 16–1630 (prick) personligen till Cons i rum 1338. Tentor som ej lämnats in inom utsatt tid kommer inte att rättas.
- Tentamen får ej lämnas in på något annat sätt.
- På tentamen skall finnas namnunderskrift.
- Kårleg skall uppvisas vid inlämning.

Utförande

- Tentamen skall utföras och redovisas enskilt. Samarbeta är ej tillåtet.
- Eventuella oklarheter reds ut med Cons enligt nedan. Telefon säkrast. Eventuellt finns jag på mitt rum ett tag efter utlämning.

	inst	hem	mobitelefon	email
Cons	4711 046	46 16 36	0708-11 47 42	cons@csd.uu.se
- Tillåtna hjälpmedel är allt utom andra människor (direkt eller indirekt).
- Cons betraktas inte som människa (i detta fall).
- Ca 8p behövs för godkänt resultat. Maxpoäng är 17p.

Översikt

Uppgifterna nedan beskriver olika sätt att utöka er ray tracer. Utökningarna kan handla om metoder som gör programmet bättre genom att antingen bli effektivare eller modellera en större del av verkligheten. Det är meningen att beskrivningen nedan skall vara tillräcklig för att förstå uppgiften. Om ni tycker att den inte är det är det lämpligt att be om ytterligare förklaring. Syftet med tentamen är *inte* att ni skall behöva brottas med att förstå själva problemet.

Lösningen på varje uppgift skall vara en redogörelse för hur man skall förfara för att utöka *er* ray tracer enligt beskrivningen. Vi är *inte* intresserade av en allmän beskrivning av hur det kan göras.

Redovisning

Följande punkter är värda att beakta angående redovisningen:

- Redogörelse för vilka nya/ändrade ADTer som behövs skall finnas med. Redogörelsen skall innehålla nya/ändrade komponenter och primitiver.
- Redogörelse för ändringar i kod skall finnas. Här skall beskrivas sådant som nya funktioner, nya/ändrade argument och ändringar av beräkningssätt och algoritmer.
- I många fall kommer det att behövas ändringar i lexikalisk analys och/eller parser. Ni behöver *inte* beskriva sådana ändringar.
- Samtliga punkter skall åtföljas av motivering och referenser till existerande kod.
- En referens till den källkod ni använder skall finnas med.
- Framställningen skall ligga på en sådan nivå att man kan lämna er existerande kod och dokumentation tillsammans med era lösningar till en kursare som sedan skall kunna genomföra ändringarna utan några större problem. Det krävs sålunda en viss fullständighet i framställningen.
- Referenser till existerande kod görs lämpligen med utdrag ur existerande kod och indikation om var ändringar skall ske.
- Ny kod behöver inte skrivas, men funktion och syfte med ny kod som behövs skall vara väl beskrivna.
- Redovisningen skall vara lättläst, välstrukturerad och imponerande.

Uppgifter

1 Transformation och sammansatta objekt

För att få större frihet vid modellering vill man på ett enkelt sätt kunna skala om, flytta och vrida objekt enskilt eller flera på en gång. Detta löses genom att man till varje objekt har ett eget lokalt koordinatsystem och transformerar till och från det när man gör själva strålföljningen.

Bilagat finns ett utdrag ur *3D Computer Graphics* beskrivs hur man kan använda en matris för att beskriva transformationer på ett objekt och hur omvandling mellan det lokala och globala koordinatsystemet går till. Läs översiktligt för att förstå principen – när du löser uppgiften kan du anta att du har funktioner för matrismultiplikation mm till hands.

1.1 Sammansatta objekt (3p)

Beskriv vilka ändringar du behöver göra för att kunna ha sammansatta objekt, dvs ett objekt som består av flera andra objekt. Den stora poängen med sammansatta objekt är att kunna behandla det som ett objekt och t ex göra transformationer på det (utan att själv behöva räkna ut nya koordinater för varje ingående objekt).

1.2 Transformation (4p)

Beskriv vilka ändringar du behöver göra i din kod för att kunna transformera objekt.

2 Test (3p)

Vid all programutveckling måste man övertyga sig själv (och andra) om att koden faktiskt fungerar. Beskriv rent allmänt hur man skall gå tillväga för att testa sin kod och använd föregående uppgift som underlag för mer specifika exempel.

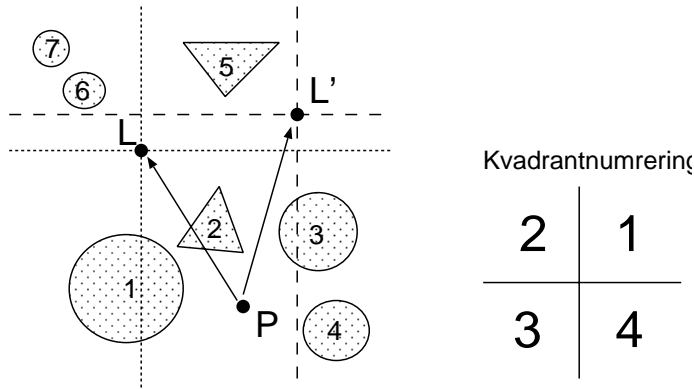
3 Light buffer (7p)

Vid beräkning av skuggning i en "ray tracer" förekommer det totalt sett väldigt många träffpunktsberäkningar. Om vi ska beräkna skuggning (illuminationen) i en punkt ska vi för varje ljuskälla undersöka om det finns ett objekt emellan ljuskällan och punkten. Givet N objekt och M ljuskällor kommer detta grovt sett att resultera i $N * M$ träffpunktsberäkningar.

För att reducera antalet träffpunktsberäkningar kan vi använda en s k "light buffer". En light buffer innebär att vi för varje ljuskälla utför en rymdpartitionering i oktanter (8 öppna kuber) med avseende på ljuskällans origo. Vidare tilldelar vi varje oktant precis de objekt som oktanten innehåller. En stråle från en ljuskälla till vilken punkt som helst kommer att befinna sig i precis en av ljuskällans oktanter och därmed behöver vi endast titta på de objekt som den aktuella oktanten innehåller. Om objekten är jämnt utspridda behöver vi utföra ca. $\frac{N * M}{8}$ beräkningar, vilket ungefär motsvarar en uppsnabbning på en magnitud.

Exempel

Lättare att föreställa sig är det två dimensionella fallet illustrerad i figur 3. (I det tredimensionella fallet rymdpartitionerar vi i oktanter i stället för kvadranter.)



Figur 1: Rymdpartitionering i Light buffer

Ljuskälla	Kvadrant	Objekt
L	1	{5}
L	2	{6, 7}
L	3	{1}
L	4	{1, 2, 3, 4}
L'	1	\emptyset
L'	2	{5, 6, 7}
L'	3	{1, 2, 3}
L'	4	{3, 4}

Figur 2: Objektmängder i rymdpartitioneringen

Om vi ska beräkna skuggningen i punkten P kommer vi att för strålen $P \rightarrow L$ endast undersöka om några objekt träffas i kvadrant nummer 4 för L , dvs $\{1, 2, 3, 4\}$ och för strålen $P \rightarrow L'$ objekt i kvadrant nummer 3 för L' , dvs $\{1, 2, 3\}$. Notera att samma objekt kan förekomma i flera kvadranter samtidigt.

Illustrera hur din ray tracer kan modifieras för att implementera *light buffers*.