

Omtentamen

TNM077 3D-datorgrafik och animering

2005-08-20 kl 14-18

Inga hjälpmedel

Denna tentamen innehåller 6 uppgifter som tillsammans kan ge maximalt 44 poäng.

För betyget G (registreras som sifferbetyg 3) krävs minst 21 poäng.

För betyget VG (registreras som sifferbetyg 5) krävs minst 32 poäng.

Uppgifterna står inte i någon särskild ordning, så läs igenom samtliga uppgifter innan du börjar. Hoppa inte över en hel uppgift bara för att du inte löst a-uppgiften, delfrågorna bygger inte alltid på varandra. Svara kortfattat men rimligt uttömmande på frågorna. Rita tydliga figurer där det gör framställningen klarare. Förklara införda beteckningar och motivera eventuella beräkningar så att de är lätta att följa. Dina svar skall visa på förståelse, inte bara faktakunskaper.

Ett par uppgifter kräver beräkningar. Någon uppgift eller två kräver dessutom ett visst mått av eftertanke och egna idéer med teorin som grund. Samtliga uppgifter kan lösas med endast de grundkunskaper som kursen avsett att förmedla, självklart med förkunskaperna som stöd.

Om något är oklart, fråga mig. Jag kommer in vid flera tillfällen under tentamenstiden för att svara på eventuella frågor.

Lycka till!

Stefan Gustavson

Uppgift 1 (6 p)

Inom datorgrafiken används olika slags modeller för interaktionen mellan ljus och objekt. En tydlig indelning kan göras mellan lokala ljusmodeller och globala ljusmodeller (local and global illumination models). Förklara kortfattat skillnaderna, och ge typiska exempel på vilket slags verkliga ljusreflexioner som kan modelleras med den ena respektive den andra sortens ljusmodell! (6 p)

Uppgift 2 (9 p)

En så kallad "Catmull-Rom spline" är en interpolerande spline, vilket innebär att den till skillnad från exempelvis en Béziér-kurva går igenom alla sina kontrollpunkter. Kurvan är en sekvens av kubiska parametriska delsegment med följande ekvation:

$$\bar{p}(u) = \sum_{i=0}^3 C_i(u)\bar{p}_i, \quad 0 \leq u < 1$$
$$C_0(u) = \frac{1}{2}(-u^3 + 2u^2 - u), \quad C_1(u) = \frac{1}{2}(3u^3 - 5u^2 + 2)$$
$$C_2(u) = \frac{1}{2}(-3u^3 + 4u^2 + u), \quad C_3(u) = \frac{1}{2}(u^3 - u^2)$$

- Visa att segmentets ändpunkter ligger i de två mittersta kontrollpunkterna. (2 p)
- Visa att tangenten i slutpunkten är parallell med sträckan mellan den andra och den fjärde kontrollpunkten. (Likaså gäller att tangenten i startpunkten är parallell med sträckan mellan den första och den tredje kontrollpunkten, men det behöver du inte visa.) (4 p)
- Antag att man har samtliga kontrollpunkter för ett segment definierade. Hur skall kontrollpunkterna för nästa segment väljas för att kurvan skall hänga ihop och ha en mjuk övergång i skarven mellan de två segmenten? Motivera med en tydlig figur eller en beräkning! (3 p)

Uppgift 3 (6 p)

Rita en scengraf för en kontorsstol på fem hjul. Hjulen skall kunna vrida sig var för sig i olika riktningar, och hjulen skall kunna rulla. Underredet på kontorsstolen skall kunna flytta sig och snurra i golvet plan. Stolens sits skall kunna höjas och sänkas samt snurra mot underredet, och ryggstödet skall kunna lutas framåt och bakåt i förhållande till sitsen.

Välj roten i din graf så att stolen enkelt kan animeras på ett realistiskt sätt.

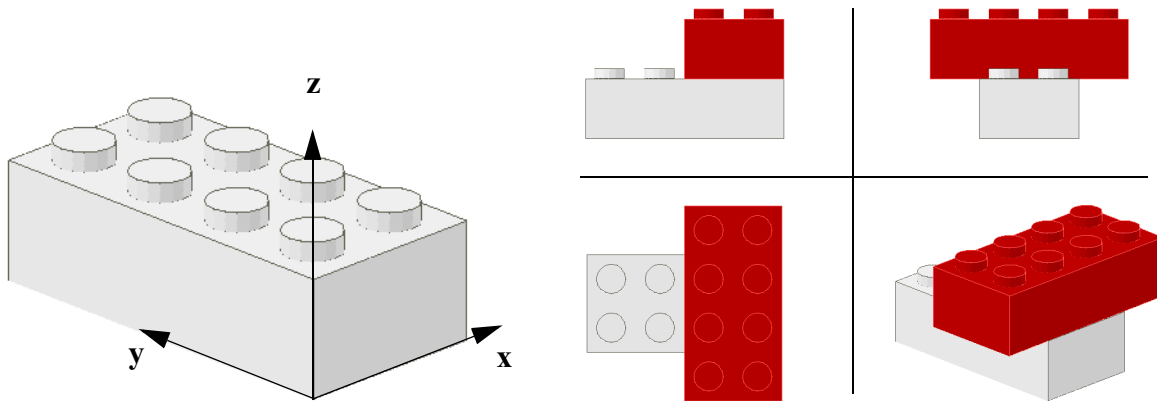
När du ritar din scengraf, ange tydligt vilken nod som styr vilken rörelse, och använd gärna de symboler som används för Java3D. Om du använder någon annan notation, förklara den tydligt så att det klart framgår hur du har tänkt.

En verklig kontorsstol finns för konkret åskådning i tentamenslokalen, men observera att du inte behöver modellera alla de rörelser som den kan utföra. Den verkliga stolens sits kan lutas, ryggstödet kan inte bara lutas utan även höjas och sänkas, och den har höj- och sänkbara armstöd. Detta kan du strunta i, endast de rörelser som beskrivs i stycket ovan behöver modelleras.

Uppgift 4 (6 p)

Du jobbar med att utveckla ett 3D-grafikgränssnitt för mobiltelefoner. Än så länge är det ganska torftigt, och kräver att man uttryckligen anger transformationsmatrisen för varje objekt. Du tänker naturligtvis snarast skriva programkod för att hantera matrismultiplikationer och skapa matriser för grundläggande operationer som translation, rotation och skalning, men du vill i alla fall testa att gränssnittet fungerar så här långt innan du går vidare.

En av dina kollegor i projektet är Lego-fantast och har som testobjekt modellerat en klassisk 2×4 knoppars Legobit. Dess utseende och lokala koordinatsystem framgår av figuren nedan till vänster. Som test vill du rita två ihopsatta Legobitar enligt figuren till höger.



Den understa biten ligger otransformerad i världens koordinater, men den översta biten måste flyttas och vridas för att hamna rätt i förhållande till den undre. Var du väljer att lägga origo för respektive bit väljer du själv, men de två bitarna skall sitta ihop som bilden visar.

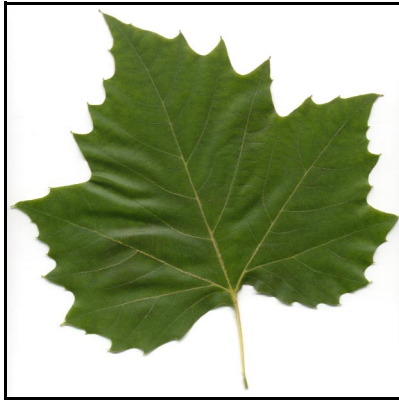
Längdenheterna i koordinatsystemet är valda så att 1 enhet i sidled (x och y) motsvarar en knapp, och 1 enhet i höjded (z) motsvarar höjden på en bit (exklusive knapparnas höjd).

Ange den fullständiga transformationsmatrisen för att utföra rätt koordinatstransformation för den översta Legobiten! Ange svaret som *en* matris, inte som en multiplikation av delmatriser. Motivera även hur du kom fram till lösningen! Ett rätt svar utan motivering ger inga poäng, men ett lätt felaktigt svar med en tydlig förklaring av tankegången kan ge nästan full poäng.

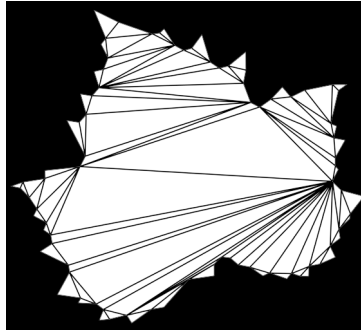
Om du behöver åskådningsmaterial finns det självklart Legobitar i tentasalen.

Uppgift 5 (10 p)

En person i din närhet verksam inom 3D-grafik försöker ersätta skicklighet med hårt arbete, något som ibland fungerar men oftast slår slint. Personen har modellerat ett lönnlöv från ett fotografi enligt nedan. Modellen fungerar och ser OK ut om man använder bilden av lövet som textur, men den innehåller nästan trehundra trianglar. Detta löv skall upprepas i 1000 kopior för att bilda ett lövverk på ett träd, och 300.000 trianglar bara för löven är på tok för mycket för att rymmas inom ramen för vad som är möjligt i den aktuella tillämpningen.



Verklig bild på ett löv



*Triangelmodell av lövet
(skapad med bilden som mall)*

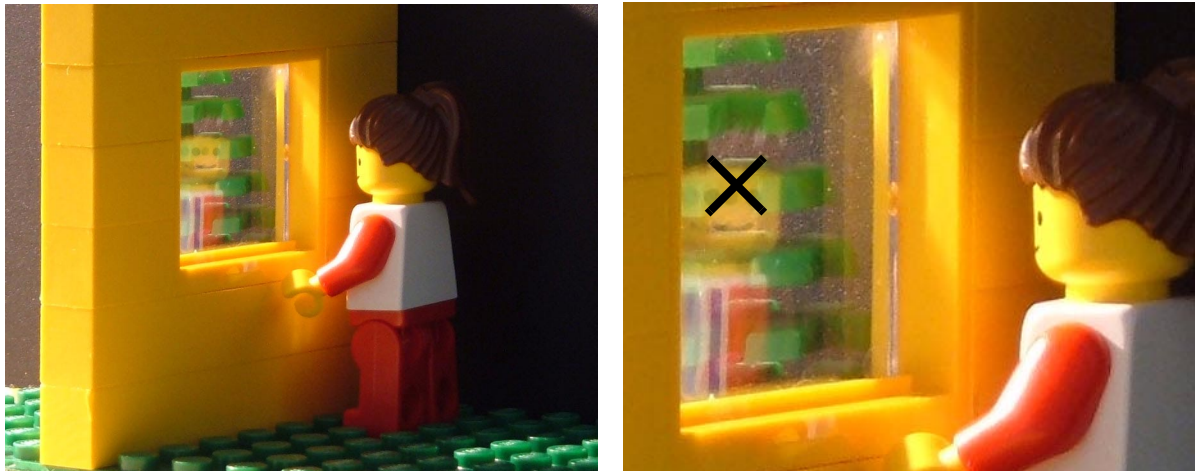


*Modellen renderad med
bilden som textur*

- Föreslå en alternativ modell för lövet som ser bra ut även på rimligt nära håll, men som klarar sig med väsentligt färre polygoner. Förklara i detalj hur din alternativa modell byggs upp, och ange hur många polygoner den har! (3 p)
- På riktigt nära håll ser lövets yta för platt och tråkig ut. Ett verkligt löv är blankt och litet skrovligt när man studerar det riktigt noga. Upphöjningarna är bara någon millimeter breda och höjdskillnaderna är ytterst små, men man finner att även den digitala modellen behöver ge ett intryck av skrovlighet på nära håll. För detta kan man använda så kallad "bump mapping". Förklara kortfattat men rimligt utförligt hur bump mapping fungerar! (3 p)
- Texturen för "bump mapping" av lövet kan göras antingen bildbaserad eller procedurell. Förklara skillnaden och diskutera några generella för- och nackdelar med respektive metod. Ange dessutom om du tycker den ena eller den andra passar bäst i just detta fall, eller om båda kan ha sina fördelar. Motivera ditt svar. (4 p)

Uppgift 6 (7 p)

En Lego-figur står i sitt Lego-hus vid sitt Lego-fönster och tittar ut på ett Lego-träd i sin Lego-trädgård. Ansiktet speglas stämningsfullt i fönsterrutan.



Detta är en verklig bild, men samtliga Legobitar finns modellerade som 3D-objekt på nätet, och det finns gratis program på nätet för att rendera virtuella Lego-modeller. Antag att man i stället hade beslutat sig för att rendera denna bild med raytracing, vilket är en ganska vanlig hobby bland vuxna Lego-freaks.

- Förklara vilka beräkningar som skulle ha utförts av en raytracing-algoritm för att rendera en pixel mitt emellan ögonen på spegelbilden av figuren (vid krysset i den förstörade delbilden till höger). I den punkten ser man både en punkt på trädet utanför fönstret och en spegling av en punkt på figurens ansikte. (4 p)
- Om man tittar noga ser man att spegligen av ansiktet blir dubbel. Det är i själva verket två spegelbilder som är litet förskjutna i sidled. Den ena är svagare, men den är fullt synlig. Förklara vad denna dubbelbild beror på, och förklara hur man kan simulera även detta med raytracing! Använd en förklarande figur. (3 p)