

Omtentamen

TNM077 3D-datorgrafik och animering
samt TNM008 3D-datorgrafik och VR

2004-08-23 kl 8-12

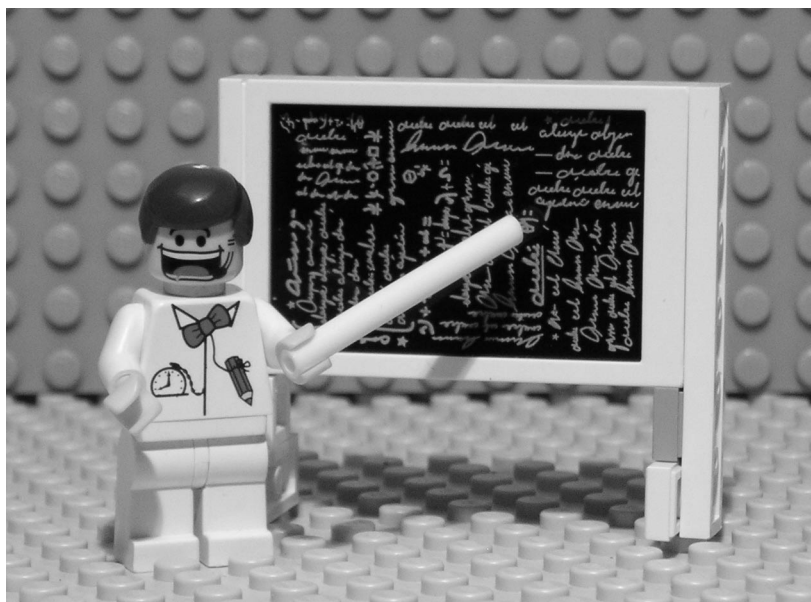
Inga hjälpmedel

Denna tentamen innehåller fem uppgifter som tillsammans kan ge maximalt 30 poäng.
För betyget G (registreras som sifferbetyg 3) krävs minst 16 poäng.
För betyget VG (registreras som sifferbetyg 5) krävs minst 22 poäng.

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.

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 (5 p)

Legobitar är pressgjutna i hårdplast, närmare bestämt akrylonitril-butadien-styren, förkortat ABS. Tillverkningsmetoden gör att ytan blir väldigt blank. ABS-plasten i sig är i stort sett färglös, men man tillsätter olika pigment i plastmassan för att få fram de typiska Lego-färgerna. Färgen är alltså inte målad på ytan, utan finns inne i plasten, vilket gör att det speglade blänket från ytan blir färglöst, men den diffusa reflexionen blir färgad av pigmentet. Phongs reflexionsmodell är en ganska bra modell för sådana material. En vanlig form för denna modell är:

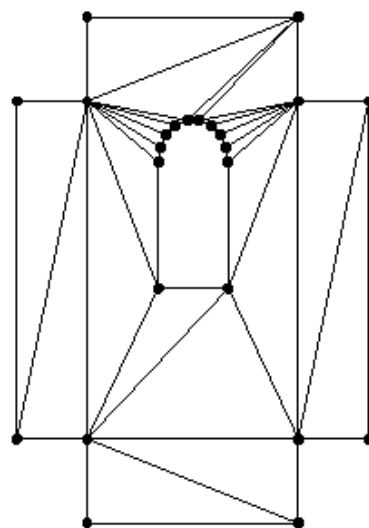
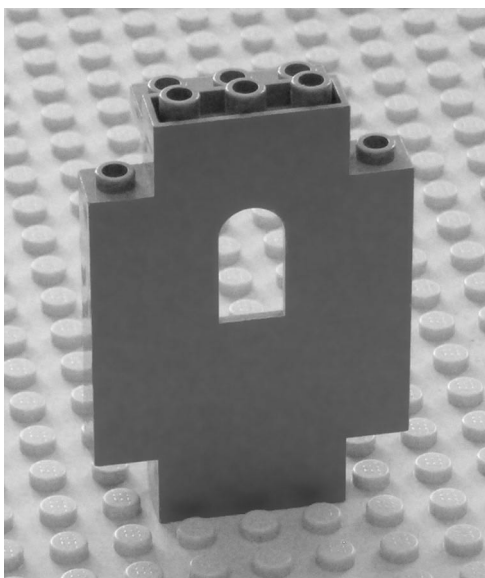
$$I = I_a k_a + I_d k_d (N \cdot L) + I_s k_s (R \cdot V)^n$$

- Ange lämpliga ungefärliga värden för materialparametrarna i ekvationen för att modellera reflexionsegenskaperna för en grå Legobit. (3 p)
- Om Legobiten i stället skall vara vit, vilken eller vilka parametrar behöver ändras? (1 p)
- När Legobitar blir slitna av många års lek blir de alltmer matta i ytan. Vilken eller vilka parametrar skulle förändras för att modellera en Legobit som blivit litet slitna, men ännu inte helt matt? (1 p)

Uppgift 2 (5 p)

Den klassiska serien av slott och borgar från Lego (från 1980- och 1990-talen) använde sig av ett stort väggelement för att minska antalet bitar till rimliga nivåer. Det såg ut enligt bilden nedan till vänster. En person du känner vill modellera detta väggelement för sina ambitiösa planer på att göra ett interaktivt 3D-grafikspel kring sin barndoms favoritleksak, en stor ljusgrå Lego-borg. Hålet i väggen, och framför allt den rundade överkanten av hålet, gör dock att en polygonmodell av den stora platta väggytan blir ganska komplicerad med många hörnpunkter och trianglar, något i stil med bilden nedan till höger. Du har bättre koll på teorin än din kamrat, och du föreslår att hålet kanske i stället kan göras med en texturmappningsmetod, med något som kallas "opacity map" eller "transparency map" i många 3D-program.

- Förklara kortfattat hur en sådan "opacity map" fungerar, och skissa hur texturbilden för detta ändamål skulle se ut. (2 p)
- Man sparar på detta sätt en hel del polygoner, men ditt förslag var egentligen inte så himla smart. Metoden har nämligen i detta fall flera nackdelar. Vilka? (Ange minst tre någorlunda tydliga nackdelar för full poäng.) (3 p)



Uppgift 3 (7 p)

Ryktet har gått i bekantskapskretsen att du är bra på datorgrafik, och att du dessutom vid det här laget har svarat på en massa frågor om Lego-animeringar. En äldre släkting till dig (som snart skall fylla 40) har bestämt sig för att med 3D-grafik återskapa sin barnoms favoritleksak: ett litet vitt Lego-hus med rött tak och en sportbil i en carport. Modellen visas i bilden här bredvid, och finns även att beskåda i verkligheten i tentamenslokalen. (Din släkting har god smak, denna modell är från 1969 och är en av de första riktigt välgjorda husmodellerna, en riktig kultklassiker enligt Lego-samlare världen över.)



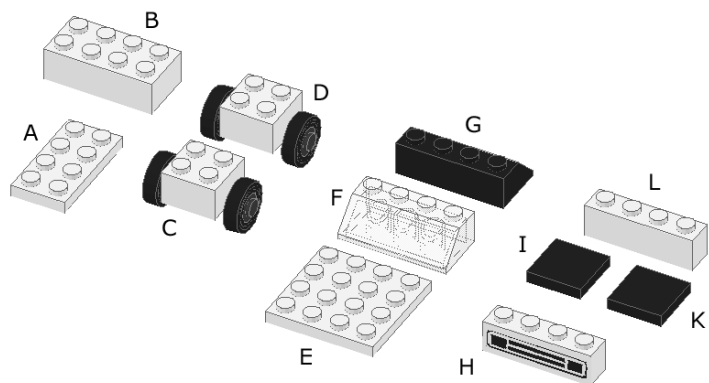
a) Modellen skall bara ses på ganska långt avstånd, och glaset i fönstren behöver inte vara genomskinligt utan kan göras svart. (Det är ändå mörkt inne i ett Lego-hus.) Förklara någorlunda detaljerat hur man i så fall lämpligen kan modellera husets fasader, alltså de vita väggarna med de röda fönstren och dörrarna. (3 p)

b) Taket består av många små bitar, men man kan förenkla det till några få plana ytor. Den sneda ytan på "tegelpannorna" är däremot skrovlig, inte slät som allt annat på huset, så de ytorna bör ges ett annat utseende. (Observera att plastytan på taket är lika *blank* som övriga ytor, men inte *slät*.) Föreslå hur man kan modellera detta för att ses på nära håll (1 p) och på långt håll (1 p).

c) För taket kan man behöva använda antingen en bildbaserad textur eller en procedurrell textur. Förklara skillnaden, och ange fördelar och nackdelar med de båda metoderna! (2 p)

Uppgift 4 (5 p)

Du vill göra en personlig 40-årspresent åt den äldre släkting som nämndes i uppgiften ovan, samtidigt som du lär dig programmera litet bättre. Du bestämmer dig för att ge bort en animerad scen i Java3D med den lilla vita sportbilen i husmodellen från uppgift 3, som han tycker så mycket om. Bilen består av 11 bitar som du efter några timmars modellerande har skapat som färdiga Shape-noder enligt bilderna A-L här bredvid. Varje nod har sitt eget lokala origo, placerat någonstans inom själva Legobiten.



Du tänker dig animeringen enligt följande: alla bitarna skall först ligga på golvet, sedan skall de sväva upp en i taget, snurra litet i luften och landa på rätt plats i modellen. Till slut skall hela den färdigbyggda bilen köra iväg. För att inte bli förvirrad i onödan bestämmer du dig för att alla transformationsgrupper antingen skall vara rena rotationer eller rena translationer. Om både rotation och translation skall utföras så skall du alltså använda två transformationsgrupper.

Rita en scengraf som enkelt möjliggör hela denna animering i Java3D! Rita hela grafen ända från roten. Tag med alla Shape-noderna, ange typen för alla noder, och ange tydligt vilka noder som styr vad i animeringen. (5 p)

Uppgift 5 (8 p)

Du skall nu animera den sekvens där bilen du skapade i uppgift 4 kör iväg. För att göra åkturen litet lagom slingrig och mer intressant bestämmer du dig för att låta den följa en kort parametrisk kurva. Du väljer ett segment av en kubisk Béziérkurva, eftersom det verkar vara något slags standardval inom datorgrafik om man inte har en bättre idé. Du hittar lätt ekvationen för kurvan i en bok:

$$\bar{p}(u) = \sum_{i=0}^3 B_i(u)\bar{p}_i \quad \begin{aligned} B_0(u) &= (1-u)^3, & B_1(u) &= 3u(1-u)^2 \\ B_2(u) &= 3u^2(1-u), & B_3(u) &= u^3 \end{aligned}$$

Bilen skall åka på ett plant golv, så du nöjer dig med en Béziérkurva i två dimensioner (x, y) .

a) Positionen för bilen ges direkt av kurvans ekvation. Teckna ekvationerna för denna position som funktion av kontrollpunkterna \bar{p}_i , på en form som du enkelt kan översätta till programkod i Java. Teckna alltså ekvationerna för x och y separat, använd inte vektornotation. (1 p)

b) För att få bilen att åka längs kurvan på ett övertygande sätt så måste du även rotera bilen så att den åker med fronten i körriktningen. Körriktningen (och även hastigheten) ges av kurvans derivata med avseende på u :

$$\bar{v}(u) = \frac{d}{du}\bar{p}(u) = \frac{d}{du} \sum_{i=0}^3 B_i(u)\bar{p}_i$$

Teckna ekvationerna för att beräkna denna vektor $\bar{v}(u)$ som pekar i körriktningen. Ange även dessa ekvationer på en form som rimligt lätt låter sig översättas till programkod i Java, använd alltså inte vektornotation i slututtrycket. (3 p)

c) För att rotera bilen i Java3D behöver du ange en rotationsvinkel. Ange hur du skulle beräkna den ur vektorn $\bar{v}(u)$! (1 p)

d) För att göra bilens färd ännu mer övertygande vill du även luta den litet åt sidan när den svänger, alltså när den accelererar i sidled. Bilens accelerationsvektor ges av andraderivatet med avseende på u av den parametriska kurvan $\bar{p}(u)$. Bilens acceleration i sidled är sedan den del av accelerationsvektorn som är ortogonal mot körriktningen. Ange hur du skulle beräkna accelerationen i sidled! Det räcker med ett slututtryck i vektorform, du behöver inte skriva svaret på en form som lätt låter sig översättas till programkod. (3 p)