

Tentamen

TNM061 3D-datorgrafik och animering
tillika omtentamen för

TNM077 3D-datorgrafik och animering

2007-06-05 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 22 poäng.

För betyget VG (registreras som sifferbetyg 5) krävs minst 33 poäng.

Uppgifterna står inte i någon särskild ordning, så läs igenom samtliga uppgifter innan du börjar. 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 (9 p)

Phongs reflexionsmodell är en enkel men ofta använd modell för ljusreflexion. En vanlig form för denna modell är:

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

- a) Förklara vad var och en av de tre termerna avser att modellera, och förklara hur det givna uttrycket kan modellera en sådan effekt! Förklara även vad de ingående storheterna står för. Rita en tydlig förklarande figur. (6 p)
- b) L , N och V i ekvationen ovan är alltid givna av scenens geometri, men R behöver beräknas. Härled ett uttryck för R ur de kända vektorerna. Rita figur och motivera alla steg i beräkningen. (3 p)

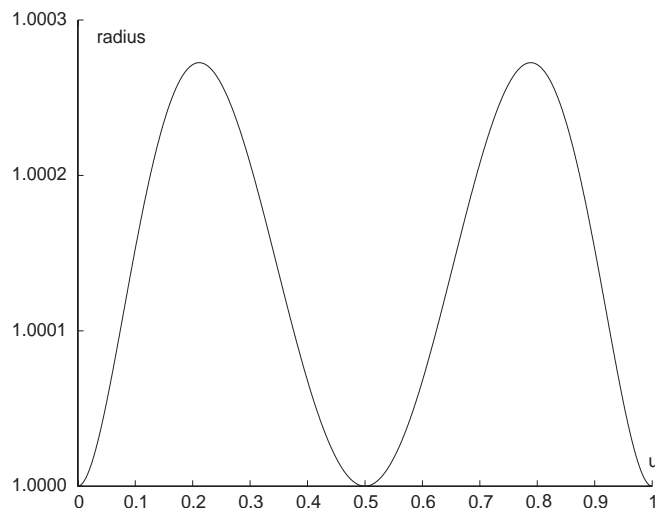
Uppgift 2 (6 p)

Cirklar och cirkelbågar kan inte representeras exakt med Béziér-kurvor, men man kan göra väldigt goda approximationer. Det vanligaste är att man approximerar en kvartscirkel med ett kubiskt kurvsegment och använder fyra segment för att rita en hel cirkel. Kontrollpunkterna för en vanlig sådan approximation av en kvartscirkel i första kvadranten i (x, y) -planet ($x \geq 0$, $y \geq 0$) med centrum i origo och radien 1 kan tecknas:

$$\bar{p}(u) = (1-u)^3 \bar{p}_0 + 3u(1-u)^2 \bar{p}_1 + 3u^2(1-u) \bar{p}_2 + u^3 \bar{p}_3, \quad 0 \leq u \leq 1$$

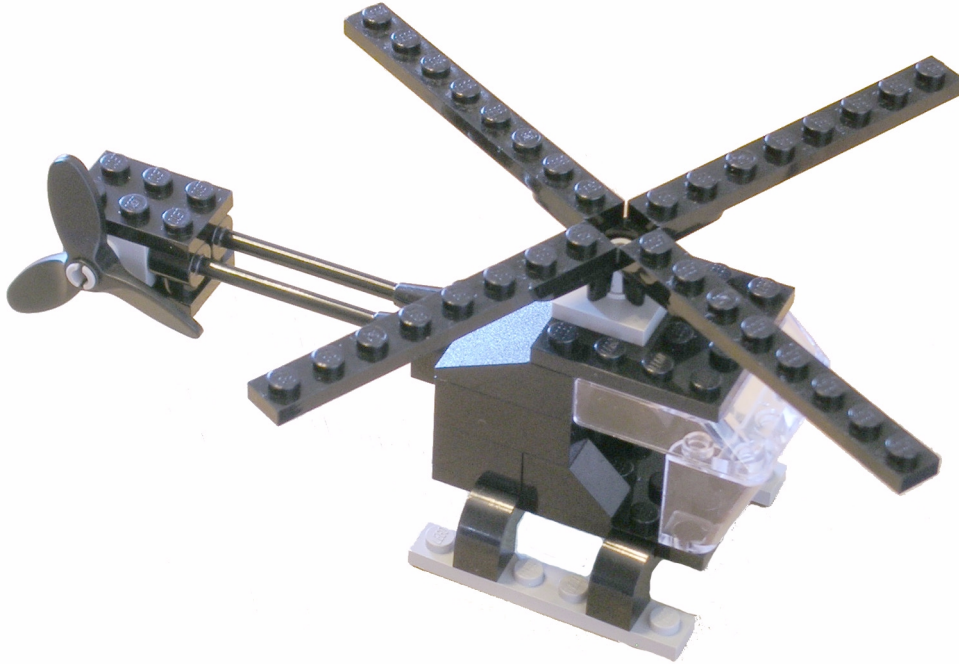
$$\bar{p}_0 = (0, 1), \bar{p}_1 = (a, 1), \bar{p}_2 = (1, a), \bar{p}_3 = (1, 0), \quad a = \frac{4}{3}(\sqrt{2} - 1) \approx 0,5523$$

- a) Visa att denna approximation har egenskapen att kurvans mittpunkt går genom punkten $(\sqrt{2}/2, \sqrt{2}/2)$, vilket även är cirkelbågens mittpunkt. (3 p)
- b) Approximationens radie är inte riktigt konstant 1, den varierar något utefter kurvan. Radien som funktion av parametern u visas i grafen nedan. Ange den fullständiga ekvationen för funktionen i grafen! (3 p)



Uppgift 3 (10 p)

En helikopter är en fascinerande och komplicerad flygmaskin. Utan att gå in alltför noga på hur en helikopter egentligen rör sig i luften och varför den ser ut som den gör är din uppgift här att skapa en scengraf i Java3D för en enkel virtuell modell av en helikopter av det slag som har en stor huvudrotor upptill och en mindre stjärtrotor på en bom baktill. Om du känner dig osäker på hur en sådan helikopter ser ut så finns i tentamenslokalen en Lego-modell som visar hur de två rotorerna är placerade och rör sig. En bild på modellen ser du här nedan.



a) Skapa en enkel men fungerande scengraf som tillåter rätt slags rörelser för helikoptern. Huvudrotorn och stjärtrotorn skall kunna snurra, helikoptern skall kunna röra sig fritt i världen och dessutom kunna svänga och luta framåt, bakåt och åt sidorna. Scengrafen skall vara sådan att det finns separata transformationer som på ett enkelt sätt kan associeras med “riktning”, “lutning framåt/bakåt” och “lutning i sidled”. Ange hela scengrafen från roten i världen, tala om vilka transformationer som görs i varje nod, deras egenskaper och vad de kontrollerar, och rita enkla skisser av dina Shape-noder och hur deras koordinatsystem är orienterade. (5 p)

b) Antag att man vill få helikoptern att åka framåt på konstant höjd över marken. Riktningen “framåt” är dock definierad i helikopterns lokala koordinatsystem och kan ha en godtycklig orientering. Hur räknar man ut vilken riktning “framåt på konstant höjd” motsvarar i världen? Referera till din scengraf i förklaringen. (3 p)

c) En välmående men oinformerad person föreslår att man skall göra animationen enligt b) enklare genom att lägga till en translation i helikopterns lokala koordinater, så att “framåt” får en enkel innebörd i grafen. Tyvärr är detta ingen bra idé. Varför inte? (2 p)

Uppgift 4 (8 p)

Klassisk raytracing (invers raytracing, “Whitted ray tracing”) är en global belysningsmodell som ger en rekursiv algoritm för rendering.

a) Förklara översiktligt, men på ett tydligt sätt, hur algoritmen fungerar och varför den blir rekursiv. Rita en tydlig figur och använd den i din förklaring. (4 p)

b) Förklara hur diffus reflexion beräknas i den klassiska raytracingalgoritmen! (2 p)

c) Förklara hur skuggor beräknas med klassisk raytracing! (2 p)

Uppgift 5 (5 p)

De flesta persondatorer har i dag särskilda enheter för att rita rörlig 3D-grafik. För att rita sådan grafik tillräckligt snabbt används i princip samma fuskmetoder som man använde för tjugo år sedan för att rendera animerade filmsekvenser. Man använder till exempel strikt lokala belysningsmodeller.

- Varför kan man inte använda mer moderna och avancerade globala belysningsmodeller för detta slags rendering? (3 p)
- Att använda en speciell enhet för 3D-grafik kan i extrema fall ge flera hundra gånger snabbare rendering jämfört med att göra exakt samma beräkningar på huvudprocessorn i datorn. Hur kan det komma sig? (2 p)

Uppgift 6 (6 p)

Spel och annan interaktiv 3D-grafik blir allt bättre med åren, men fortfarande är det viktigt att spara in på modellernas komplexitet för att få renderingen att gå rimligt snabbt. Nedan ser du två bilder av en figur ur rollspelet Oblivion från Bethesda Softworks. Ansiktet är automatiskt genererat utifrån ett antal parametrar med en komplicerad algoritm för att man skall få lagom variation mellan de flera hundra individerna i spelet utan att behöva modellera var och en för hand.

- Mellan de två bilderna har en modellparameter ändrats som får figuren att se äldre ut. Det har tillkommit många veck och rynkor i ansiktet, och huden ser ojämn ut. Den högra bilden har dock inte fler polygoner än den vänstra, och de rent geometriska skillnaderna mellan de två modellerna är inte särskilt stor alls. Förklara hur denna förändring i hudens utseende har gjorts, och förklara även översiktligt hur det fungerar rent tekniskt! (3 p)
- Håret och frisyren är exakt detsamma i båda bilderna, endast färgen har ändrats från brunt till grått. Däremot är håret modellerat på ett speciellt sätt. Det skulle bli alldeles för resurskrävande att modellera enskilda hårstrån, men man vill ändå ha ett halvgenomskinligt, stripigt utseende för nedhängande hårtestar, och man vill att kanten på håret skall vara lätt genomskinlig och taggig så att inte alla figurer i spelet ser ut att gå runt med perfekta frisyrer hela tiden. Beskriv hur detta kan uppnås utan att man ägnar särskilt många polygoner åt håret! Svara inte bara med ett namn på metoden, utan förklara översiktligt hur den fungerar. (3 p)

