

08.72.64 Tölvugrafik

Lokapróf

Kennari: Hjálmtýr Hafsteinsson

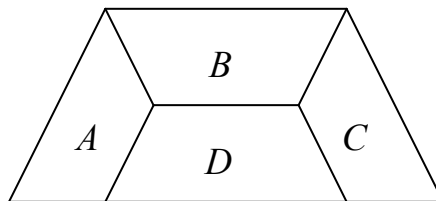
3. maí, 2006

kl. 9⁰⁰-12⁰⁰

Öll dæmin hafa sama vægi. Aðeins þarf að leysa 5 dæmi af 6. Fimm bestu dæmin gilda.

Öll skrifleg hjálpargögn og reiknivél leyfileg.

- Í þessu dæmi nægir að miðað sé við tvívíðar varpanir.
 - Eru eftirfarandi varpanapör víxlin (e. commutative), þ.e. fyrir varpanirnar T_1 og T_2 , er $T_1 T_2 = T_2 T_1$? Rökstyðjið hvert tilfalli.
 - Tvær hliðranir (e. translation)
 - Snúningur og hliðrun
 - Tvær kvarðanir (e. scaling)
 - Speglun (e. reflection) um hnitakerfisás fæst með því að kvarða tilsvareandi hnit með -1 . Sýnið hvernig hægt er að spegla um línu sem gefin er með tvívíðu punktum P_1 og P_2 .
- Á sama hátt og hægt er að teikna þéttlista Sierpinskis með endurteknum þríhyrningum (sjá kafla 2.9 á bls. 77-80 í kennslubók) er hægt að nota jafnarma trapisu, eins og á myndinni hér að neðan, til að teikna brotamynd.



Myndin sýnir eina stóra jafnarma trapisu samsetta úr fjórum minni, A , B , C og D . Allar fjórar minni trapissurnar eru jafnstórar (teikningin er e.t.v. ekki alveg nákvæm!) og hlutfallið á milli lengda samsíða hliða þeirra er 1 á móti $\sqrt{2}$ ($\sim 1.4142\dots$). Sama hlutfall er á milli hliðarlengdanna í stóra samsíðungnum (þ.e. lengd lengri samsíða hliðarinnar er $\sqrt{2}$ -sinnnum meiri en lengd styttri hliðarinnar).

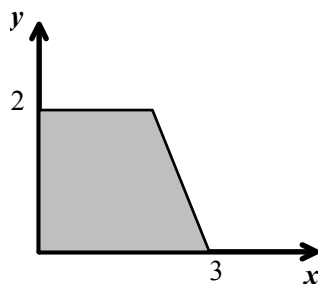
Sýnið helstu föllin í OpenGL forriti sem býr til mynd sem er sambærileg við þéttlista Sierpinskis með því að fylla endurkvæmt upp í trapissurnar A og C . Útskýrið einnig með teikningu og orðum hvernig forritið ykkar vinnur.

3. Þegar ljós endurkastast af vatni að næturlagi þá verður endurskinið langt og mjótt eins og sést á myndinni hér að neðan.

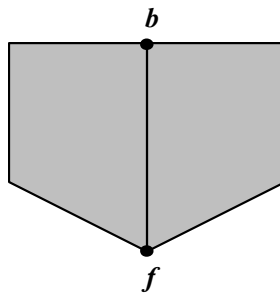


- Útskýrið þetta endurskin út frá því sem þið vitið um sjónhornsvörpun (e. perspective), endurskinslíkan Phong og eiginleika vatns.
- Ef vatnsyfirborðið væri algerlega slétt hvernig yrði þá endurskinið og hvers vegna?
- Nefnið önnur yfirborð þar sem svipaðir endurskinseiginleikar koma fram.

4. Gefið er fallið $Vaengur()$, sem teiknar vænginn á myndinni hér að neðan, sem liggur í $z=0$ planinu.



- Skrifið OpenGL fallið $Fidrilidi(f_x, f_y, f_z, b_x, b_y, b_z)$, sem notar $Vaengur()$ til að teikna fiðrildið hér að neðan. Fallið fær sem viðfang hnit punktanna f , sem er staðsetning á framenda fiðrildisins, og b , sem er staðsetning á afturenda þess. Athugið að stærð fiðrildisins ræðst af fjarlægðinni á milli b og f . Notið $glPushMatrix()$ og $glPopMatrix()$ á réttum stöðum til að hafa ekki áhrif á aðra hluta forritsins.



- Útskýrið hvernig hægt væri að breyta fallinu í $a)$ -lið þannig að vængirnir geti snúist á móti hvor öðrum, allt að 90° um ásinn sem f og b mynda, til að líkja eftir vængjaslætti fiðrilda.

5. Ef notuð er *billboarding* aðferð til að teikna tré, þá er trjá-mynstri er varpað á ferning, með alpha gildi sett sem 1.0 þar sem tréð er í mynstrinu, en 0.0 á öllum öðrum stöðum. Með blöndun (e. blending) þá sjást hlutir sem eru fyrir aftan “tréð” í gegnum það á þeim stöðum þar sem alpha er 0.0. Þessi aðferð virkar ef tréð er teiknað á eftir öllum öðrum hlutum. Ef ógegnisær hlutur sem er fyrir aftan tréð er teiknaður á eftir trénu þá kemur hann ekki rétt út.

a) Útskýrið hvað gerist í þessu tilfelli og hvert vandamálið er.

b) Í OpenGL er hægt að kveikja á svokölluðu *Alpha prófun* (e. alpha test), sem leyfir notanda að birta einstaka skjápunkta eftir alpha-gildi þeirra. Fallið `glAlphaFunc` er notað til að skilgreina skilyrði á alpha-gildi skjápunkta til þess að þeir séu teiknaðir, t.d. `glAlphaFunc(GL_GREATER, 0.5)` birtir aðeins þá punkta marghyrnings með alpha-gildi stærra en 0.5. Útskýrið hvernig hægt væri að nota Alpha prófun til að útfæra billboarding sem teikniröð marghyrninga skiptir ekki máli (þ.e. leysa vandamálið sem lýst er að ofan).

6. Í OpenGL forriti á áhorfandinn að fljúga um líkanið eftir 3ja stigs Bezier ferli og hann á ávallt að snúa fram eftir ferlinum. Áhorfandinn á að hefja ferðina í stýripunktinum P_0 og enda hana í P_3 .

Ef þið fáið gefna stýripunktanna fjóra, útskýrið þá nákvæmlega hvernig þið mynduð staðsetja áhorfandann og ákveða stefnu hans. Notið teikningar, útreikninga og OpenGL skipanir.