

TÖL203M Tölvugrafík



Prófdagur og tími: 06.05.2014 13:30-16:30

Prófstaður:

Aðalbygging - A225 (fjöldi: 1)

VR-2 - V152 (fjöldi: 15)

VR-2 - V155 (fjöldi: 16)

VR-2 - V156 (fjöldi: 12)

VR-2 - V261 (fjöldi: 19)

HÁSKÓLI ÍSLANDS

Iðnaðarverkfræði-, vélaverkfræði- og tölvunarfræðideild

Skriflegt próf (Vægi: 70%)

Skráðir til prófs: 63

Kennari:

Hjálmtýr Hafsteinsson (hh@hi.is / S: 4932 / GSM: 8958772) Umsjónarkennari

Kennslumisseri: Vor 2014

Úrlausnir skulu merktar með nafni

Prófbók/svarblöð:

Línustrikuð prófbók

Hjálpargögn:

Vasareiknir

Tölva með textaminni

Öll skrifleg hjálpargögn eru leyfileg.

Önnur fyrirmæli:

Prófverkefnið er á ensku fyrir þá sem óska þess.

Aðgangur að prófverkefni að loknu prófi:

Kennslusvið sendir eintak í prófasafn

Einkunnir skulu skráðar í Uglu eigi síðar en 20.05.2014.

AHUGIÐ að einhverjar úrlausnir úr fjölmönnum prófum geta verið í þunnum umslögum sem auðvelt er að yfirsjá. GÓÐ VINNUREGLA er að byrja á því að opna öll umslög, telja úrlausnir og athuga hvort fjöldi stemmir við uppgefinn fjölda sem kvittað var fyrir.

Prentað: 29.04.14

Samkvæmt 60. grein Reglna fyrir Háskola Íslands skulu einkunnir birtar í síðasta lagi tveimur vikum eftir hvert próf, nema eftir desemberpróf, þá eftir þrjár vikur. Einkunnir skulu skráðar í Uglu.

TÖL203M Tölvugrafík

Lokapróf

Kennari: Hjálmtýr Hafsteinsson

6. maí, 2014

kl. 13³⁰-16³⁰

Öll dæmin hafa sama vægi. Aðeins þarf að leysa 5 dæmi af 6. Fimm bestu dæmin gilda. Öll skrifleg hjálpargögn og reiknivél leyfileg.

1. a) Tvívíðu varpanirnar kvörðun $S(s_x, s_y)$ og snúningur $R(\theta)$ eru almennt ekki víxlnar (þ.e. $S(s_x, s_y) * R(\theta) \neq R(\theta) * S(s_x, s_y)$). Þær eru þó víxlnar ef ákveðnar skorður eru settar á kvörðunina og/eða snúningin. Tilgreinið slíka skorðu og rökstyðjið að hún geri varpanirnar víxlnar með því að sýna fylkin.
- b) Við höfum tvívíðan hlut með miðju í (2, 2). Við viljum varpa honum þannig að miðja hans sé núna í (-2, -2). Það er hægt að gera það á nokkra vegu:
- i. snúa um 180°
 - ii. hliðra um (-4, -4)
 - iii. spegla um $x=-y$ ásin.
- Eru þessar þrjár aðferðir algerlega jafngildar? Ef ekki, að hvaða leyti eru þær ólíkar? Rökstyðjið.

2. a) Lát P_1, P_2, \dots, P_n vera n punkta í þrívídd. Lýsið aðferð sem ákvarðar hvort þeir liggja allir í sömu sléttu (*plane*). Rökstyðjið að aðferðin sé rétt.
- b) Bakhliðareyðing (*back-face culling*) er yfirleitt ekki notuð í geislasömölun (*ray tracing*). Hvers vegna ekki? Hvaða vandræðum getur það valdið í geislasömölun sem koma ekki upp í venjulegri OpenGL grafík? Útskýrið.

3. Hér að neðan er bútur úr `display`-falli OpenGL forrits, sem sýnir teningslaga "jörð" snúast í kringum teningslaga "sól". Þetta er einfaldaður bútur úr forritinu `TveirTen.cpp`, sem sýnt var í námskeiðinu.

```
mat4 mv = Translate(0.0, 0.0, -2.0);
mv *= RotateX( (GLfloat)spinx ) * RotateY( (GLfloat)spiny );
glUniformMatrix4fv( modelview, 1, GL_TRUE, mv );

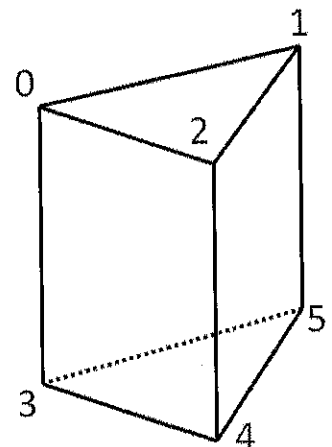
// Teikna "sólina"
glDrawArrays( GL_TRIANGLES, 0, NumVertices );

// Teikna "jörðina"
mv *= RotateY(rotAngle) * Translate(1.5, 0.0, 0.0)
    * Scale(0.2, 0.2, 0.2);
glUniformMatrix4fv( modelview, 1, GL_TRUE, mv );
glDrawArrays( GL_TRIANGLES, 0, NumVertices );
```

- Útskýrið tilgang skipananna hér að ofan, sérstaklega vörpunarskipananna, og lýsið því hvar áhorfandinn virðist staddur miðað við "sólkerfið".
- Breytið bútnum þannig að nú sé eins og áhorfandinn sé staddur á "jörðinni" og sjái "sólina" snúast í kringum sig. Útskýrið nákvæmlega hvernig kóðinn ykkar virkar. Kóði án útskýringa gefur engin stig hér.

4. Prisma (*prism*) er margflötungur með 5 hliðar eins og sést á myndinni hér til hliðar.

- Ef við gerum ráð fyrir að við vinnum aðeins með þríhyrninga sýnið þá PLY-skrá fyrir prismuna sem er á myndinni. Gerið ráð fyrir að botninn liggji í xz -planinu og að hnútur 3 sé með hnit $(0, 0, 0)$, en að öðru leyti getið þið ákveðið hnitin sjálf (svo fremi sem hluturinn sé áfram prisma).
- Sýnið hvernig þið mynduð teikna prismu sem tvo **TRIANGLE_STRIP** búta. Þessi liður er óháður *a*-lið.



5. a) Það hefur verið stungið uppá því að í stað þess að nota umhverfislýsingu (*ambient*) í lýsingarlíkani Phongs að nota frekar annan ljósgjafa sem alltaf er staðsettur í auganu. Þessi viðbótarljósgjafi gefur frá sér dauft ljós í allar áttir, án dofnunar. i) Hvernig er þessi aðferð svipuð og umhverfislýsing? ii) Hvernig verður þessi lýsing frábrugðin?

- b) Munið lýsingarlíkan Phongs fyrir lit punkts P á yfirborði hlutar:

$$I = k_a I_a + k_d I_d (\mathbf{l} \cdot \mathbf{n}) + k_s I_s (\mathbf{v} \cdot \mathbf{r})^\alpha$$

Í líkaninu er gert ráð fyrir því að allir vigrarnir (\mathbf{l} , \mathbf{n} , \mathbf{r} og \mathbf{v}) séu einingarvigrar (*normalized*). Hvernig myndi liturinn á P breytast ef það gleymdist að gera vigrinn \mathbf{l} að einingavigri (stærð hans færi þá eftir fjarlægð ljósgjafa frá P). Útskýrið hvaða áhrif það hefði á einstaka hluta lýsingarlíkansins þegar hluturinn færir til.

6. Einfalt spjald (þ.e. samsett út tveimur þríhyrningum) er litað með lýsingarlíkani Phongs og einum ljósgjafa. Við viljum láta spjaldið líta út eins og bárujárnsplötu.

- a) Útskýrið nákvæmlega hvernig við getum ná því fram með því að breyta aðeins þvervigrum í bútalitara (*fragment shader*).
- b) Hvaða gallar eru á þessu líkani af bárujárnsplötu úr a)-lið? Útskýrið hvernig hægt væri að laga þá galla.

TÖL203M Computer Graphics

Final exam

Teacher: Hjálmtýr Hafsteinsson

May 6th, 2014

Time: 13³⁰-16³⁰

All problems have the same value. You only have to solve 5 problems out of 6. The best five solutions count. All written material and a calculator are allowed.

1. a) The 2D transformations scaling $S(s_x, s_y)$ and rotation $R(\theta)$ do not in general commute (i.e. $S(s_x, s_y) * R(\theta) \neq R(\theta) * S(s_x, s_y)$). However they do commute if certain restrictions are placed on the scaling and/or the rotation. Show such a restrictions and justify that it makes the transformations commute by showing the matrices.
- b) We have a 2D object with its center in $(2, 2)$. We want to transform it so that its center is now in $(-2, -2)$. It can be done in several ways:
 - i. rotate by 180°
 - ii. translate by $(-4, -4)$
 - iii. mirror about the $x=-y$ axis.

Are these three methods exactly equivalent? If not, in what ways are they different? Justify.

2. a) Let P_1, P_2, \dots, P_n be n points in 3D. Describe a method to determine if they all lie in the same plane. Argue that your method works correctly.
- b) Back-face culling is usually not used in ray tracing. Why not? What problems can it cause in ray tracing that do not come up in normal OpenGL graphics? Explain.

3. Below is a piece of the `display`-function of an OpenGL program that show a cube-like "earth" revolving around a cube-like "sun". This is a simplified fragment from the program `TveirTen.cpp`, that was studied in the course.

```
mat4 mv = Translate(0.0, 0.0, -2.0);
mv *= RotateX( (GLfloat)spinx ) * RotateY( (GLfloat)spiny );
glUniformMatrix4fv( modelview, 1, GL_TRUE, mv );

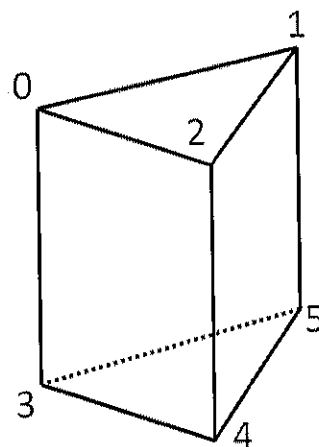
// Teikna "sólina"
glDrawArrays( GL_TRIANGLES, 0, NumVertices );

// Teikna "jörðina"
mv *= RotateY(rotAngle) * Translate(1.5, 0.0, 0.0)
      * Scale(0.2, 0.2, 0.2);
glUniformMatrix4fv( modelview, 1, GL_TRUE, mv );
glDrawArrays( GL_TRIANGLES, 0, NumVertices );
```

- Explain the purpose of the commands above, especially the transformations, and describe where the viewer seems to be positioned relative to the "solar system".
- Change the code fragment so that now it seems like the viewer is positioned on the "earth" and sees that "sun" revolve around him/her. Explain carefully how your code works. Code without explanations gives no points here.

4. A prism is a polytope with 5 sides as you can see on the adjacent figure.

- If we assume that we are only using triangles, show a PLY-file for the prism given in the figure. Assume that the bottom lies in the xz -plane and that vertex 3 has coordinates $(0, 0, 0)$. You can decide the rest of the coordinates yourselves as long as the object remains a prism.
- Show how you could draw a prism as two **TRIANGLE_STRIP** pieces. This part is independent of part a).



5. a) It has been suggested that instead of using ambient lighting in the Phong illumination model we just use another light source that is always positioned in the same point as the viewer. This additional light source emits soft light in all directions, without attenuation. i) How is that method similar to ambient lighting? ii) How will this new lighting be different?
- b) Recall the Phong illumination model to compute the color of point P on the surface of an object:

$$I = k_a I_a + k_d I_d (\mathbf{l} \cdot \mathbf{n}) + k_s I_s (\mathbf{v} \cdot \mathbf{r})^\alpha$$

The model assumes that all the vectors (\mathbf{l} , \mathbf{n} , \mathbf{r} , and \mathbf{v}) are normalized. How would the color of P change if we forgot to normalize the vector \mathbf{l} (then its size would depend on the distance between the light source and P). Explain what effects it would have on individual parts of the illumination model when the object is moved around.

6. A simple card in 3D (i.e. made up of two triangles) is shaded with the Phong illumination model and one light source. We want to have the card look like a sheet of corrugated iron.

- a) Explain carefully how we can accomplish that by only changing the normal vectors in the fragment shader.
- b) What flaws are there in this model of corrugated iron from part a)? Explain how they could be fixed.