

1. [Gamalt prófdæmi] Þegar við notum MIP vörpun (trilinear) þá eykst minnispláss hvers mynsturs um 33% (tekur 4/3 sinnum meira pláss). Við ójafnhliða (anisotropic) mynsturvörpun fjórfaldast minnisnotkun hvers mynsturs. Við tókum þó eftir því að mörg mynstur minnka aðeins í einni vídd, t.d. mynstur á gólfi minnkar meira í t-víddinni (þ.e. hæð mynstursins), en mynstur á veggjum minnka meira í s-víddinni (þ.e. breidd mynstursins). Það væri því hægt að búa til útgáfu af ójafnhliða mynsturvörpun þar sem mynstrin minnka aðeins í einni vídd, eftir því hvort þau eru notuð á lárétt eða lóðrétt yfirborð. Hve mikil yrði minnisaukningin í þessari útgáfu? Hverjir væru helstu gallar þessarar aðferðar? Rökstyðjið í nokkrum orðum.

Minnisaukning væri tvöföldun á upprunalegri minnisnotkun:  
Lóðrétt yfirborð nota þá bara s-vídd og lárétt bara t-vídd.

Helsti kosturinn er auðvitað helmingi minni minnisnotkun, 2x í staðinn fyrir 4x.

Helstu gallar: Myndi ekki höndla yfirborð á ská vel, þar sem þau þyrftu að nota báðar víddir. Stefna yfirborðanna þyrfti að vera flokkuð áður en ákveðið er hvernig vörpun væri notuð. Ef yfirborð breytist úr lóðréttu í skáhallandi eða lárétt gætu sést gallar. Minni sveigjanleiki, ekki hægt að nota sama mynstur á mismunandi 'oriented' yfirborð.

2. [Próf 2023] Bakhliðareyðing og litaraforritun:

a. Segjum að við litum margflötung (polytope) með endurskinslíkani Phong og Phong litun. Ef það er ljósgjafi fyrir aftan margflötunginn (frá áhorfanda) og ef bakhliðareyðing (back-face culling) er ekki virk, hvernig lit fá þá bakhliðar margflötungsins (ef við sæjum inn í hann)? Rökstyðjið.

Þær væru svartar eða mjög dökkar því þvervigur bakhliðarinnar snýr frá áhorfandanum. Þar sem diffuse og specular gildin eru reiknuð með punktmargfeldi þvervigursins og átt ljósgjafans (diffuse) eða endurvarps-vigursins (specular) myndu þau vera ca 0 og bara ambient lýsingin væri sýnileg.

b. Hvernig breytist svar ykkar við a-lið ef bakhliðareyðing er virk? Rökstyðjið.

Ef bakhliðareyðing er virk eru bakhliðarnar ekki teiknaðar.

c. GLSL bútalitarar hafa innbyggðu breytuna `gl_FrontFacing`. Þetta er breyta af taginu `bool`, sem er sönn ef núverandi bútur er hluti af þríhyrningi sem snýr að áhorfandanum, en ósönn annars. Skrifðu GLSL kóðabút sem veldur því að allar bakhliðar sem sjást eru rauðar, en framhliðar væri litaðar eins og venjulega (þið þurfið ekki að skrifa út þann hluta kóðans!). Hvað myndi gerast í kóðanum ykkar ef bakhliðareyðing væri virk? Útskýrið.

// í bútalitara:

```
void main() {  
    if (!gl_FrontFacing) {  
        gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);  
    }  
    // ...  
}
```

Ef bakhliðareyðing væri sönn myndi `gl_FrontFacing` alltaf skila sönnu og kóðinn í `if` setningunni yrði aldrei keyrður.

3. [Próf 2023] Hér fyrir neðan eru stuttar spurningar úr ýmsu efni námskeiðsins. Svárið hverri spurningu með nokkrum setningum.

a. Berið saman Gouraud litun og Phong litun. Nefnið einn kost og einn galla við hvora aðferð.

Gouraud er ódýrari þar sem lýsing er bara reiknuð í hnútalitara og brúuð á milli hnúta. Útkoman er því ónákvæmari, sérstaklega verr á kúpt yfirborð þar sem notuð er línuleg brúun.

Phong er mun raunverulegri lýsing þar sem hún brúar yfirborðs þvervigrana og reiknar lýsinguna fyrir hvern bít, í bútalitanum.

b. Er hægt að útfæra mynsturvörpun eingöngu í hnútalitara? Rökstyðjið svar ykkar.

Nei, þá fengum við bara rétt texel gildi á hnútunum sjálfum og línuleg brúun á milli hnúta myndi teygja mynstrið ónáttúrulega yfir bítana þar á milli. Þufrum að 'sampla' mynstrið í hverjum bít með því að nota tex hnitin.

c. Skuggakort (shadow maps) eru ein leið til að fá skugga í WebGL. Lýsið stuttlega hvernig hún virkar og hvaða takmarkanir eru á aðferðinni.

Horfir fyrst frá sjónarhorni ljóssins og geymir dýptar gildin í skuggakorti.

Fyrir hvern búið: ber saman gildi bútarins í skuggakortinu við dýpt bútarins frá áhorfandanum séð. Ef skuggakortið er með minna gildi, þá er búturinn í skugga.

Takmarkanir:

Bara 'harðir' skuggar, feida ekki út. Er háð upplausn, getur búið til ónátturulegar útlínur skugga (ekki smooth). Höndlar ekki gegnsæja hluti. Þarf að rendera tvisvar, fyrst til að búa til skuggakortið. Minnis overhead fyrir skuggakortið.

d. Útskýrið hvaða kosti það hefur að nota jafnþætt (homogeneous) hnit í tölvugrafík.

Gera það kleyft að nota fylkjamargföldun fyrir aðgerðir eins og translation, rotation, projection, ..

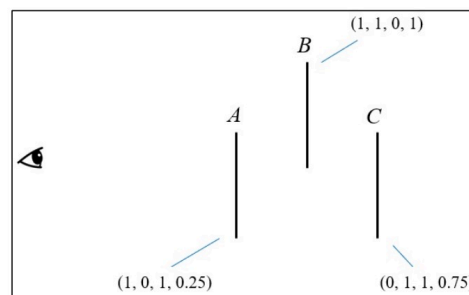
e. Hvers vegna skiptir máli að röð hnúta í þríhyrningi sé rétt (þ.e. rangsælis eða réttsælis)? Útskýrið.

Sjálfgefið notar gl rangsælis, en það væri alveg eins hægt að nota réttsælis ef allt væri stillt samkvæmt því. Aðalatriðið er að nota alltaf sömu röð, því annars myndu framhliðarnar verða túlkðar sem bakhliðar og öfugt, fyrir þann hlut sem er skilgreindur í öfugri röð.

4. [Próf 2020] Gefnir eru þrír þríhyrningar, A, B og C, sem eru mislangt frá áhorfanda. Á myndinni hér fyrir neðan sjást þeir, afstaða þeirra gagnvart áhorfandanum og litagildi hvers þeirra í RGBA-gildum.

a. Í hvaða röð þarf að teikna þríhyrningana ef dýptarminni er virkt og við viljum láta gegnsæi koma fram? Rökstyðjið.

Við teiknum fyrst gegnheila þríhyrninginn B, (með alpha gildi 1.0). Síðan teiknum við þá sem eru að einhverju leiti gegnsærir í röð frá þeim lengst frá áhorfanda til næst, þannig röðin væri B, C, A.



b. Ef þríhyrningarnir eru teiknaðir í röðinni C, B, A, hvaða blöndunarformúlu (þ.e. stuðla) þarf þá að nota við hverja teikningu til að gegnsæi komi fram? Rökstyðjið.

Myndum vanalega nota: `gl.blendFunc(gl.SRC_ALPHA, gl.ONE_MINUS_SRC_ALPHA);`  
En þar sem gegnheili þríhyrningurinn B er ekki teiknaður fyrst getum við notað `gl.ONE`, `gl.ZERO` áður en við teiknum B til þess að hann haldist örugglega gegnheill.

```
gl.blendFunc(gl.SRC_ALPHA, gl.ONE_MINUS_SRC_ALPHA); // C
gl.blendFunc(gl.ONE, gl.ZERO); // B
gl.blendFunc(gl.SRC_ALPHA, gl.ONE_MINUS_SRC_ALPHA); // A
```

5. Breytið `three.js` sýnisforritinu 20-point-light-shadowmap þannig að það séu tveir punktljósgjafar sem báðir gefa skugga. Þeir eiga báðir að vera sjáanlegir (þ.e. vera með `PointLightHelper`) og það á að vera hægt að breyta staðsetningu beggja með valmyndinni. Í sýnisforritinu eru kúla og teningur. Látið þau hreyfast þannig að boltinn fer upp og niður á sama stað (svipað og hinu skuggasýnisforritinu) og teningurinn á að færast fram og til baka. Þið ættuð líka að prófa að breyta upplausn skuggakortsins, sem er stikinn `light.shadow.mapSize` (sjá t.d. sýniskóða). Skilið skjámynd og hlekk á forritið.

<https://kristinnroach.github.io/HI-grafik-assignments/src/H7/shadowmap.html>

