

Virtualna Okruženja

Vježba 1

Programiranje grafičkog sklopovlja

1. Preslikavanje neravnina

Vježbu sam riješio u DirectX/HLSL inačici.

Kako bi se ostvarilo preslikavanje neravnina potreban je jedan prolaz kroz grafički cjevovod sa odgovarajućim programima za sjenčanje vrhova i fragmenata. Na izlazu iz sjenčanja vrhova potrebni su (uz uobičajene parametre) vektor prema svjetlu i prema oku.

U programu za sjenčanje fragmenata trebalo je dodati sljedeće linije koda:

Dohvat normale iz teksture normala	<pre>float3 n = normalize(tex2D(normalMap,IN.texCoord)*2 + float3(-1,-1,-1));</pre>
Računanje polu-vektora h	<pre>float3 l = normalize(IN.lightDir); float3 v = normalize(IN.viewDir); float3 h = normalize(IN.lightDir + IN.viewDir);</pre>
Računanje faktora refleksije prema pojednostavljenoj formuli refleksije	<pre>float nSpecL = pow(dot(n, h), material.shininess);</pre>
Računanje ukupne boje svjetla na temelju materijala i izračunatih faktora svjetla (difuznog i reflektiranog)	<pre>float4 color = (material.ambient * light.ambient) + (material.specular * light.specular * nSpecL) + (material.diffuse * light.diffuse * nDotL);</pre>

Na slici 1. nalazi se scena bez preslikavanja neravnina, a na slici 2. je scena sa preslikavanjem neravnina.



Slika 1. Bez preslikavanja ravnina



Slika 2. Preslikavanje neravnina

2. Preslikavanje sjena

Vježbu sam riješio u DirectX/HLSL inačici.

Za preslikavanje sjena potrebna su dva prolaza kroz grafički cjevovod. U prvom se prolazu scena crta iz perspektive jedinog svjetla u scenu, a kao rezultat se dobiva tekstura sjena koja sadrži podatke o udaljenosti fragmenata vidljivih iz pozicije svjetla.

U drugom se prolazu scena crta iz perspektive kamere i tada se opet računa udaljenost fragmenta od izvora svjetlosti. Novi korak je usporedba dobivene udaljenosti sa onom u teksturi sjena. Ako je udaljenost veća, znači da postoji fragment koji je bliže svjetlu i trenutni se fragment nalazi u sjeni te ga treba potamniti.

U tablici u nastavku su linije koda koje je trebalo dodati u program za sjenčanje:

Početna transformacija viewPosLight vektora	<code>float4 transVPL = (IN.viewPosLight / IN.viewPosLight.w) + float4(1,1,1,1) / 2; transVPL.y = 1.0 - transVPL.y;</code>
Čitanje dubine iz mape sjena	<code>float shadowDepth = tex2D(shadowMap, float2(transVPL.x, transVPL.y)).x;</code>
Ako trenutni fragment ima veću udaljenost nego fragment iz mape sjena, onda je trenutni fragment u sjeni!	<code>float factor = 1.0; if(shadowDepth < (IN.viewPosLight.z/xMaxDepth - 0.01)){ factor = 0.0; }</code>
Gađenje difuznog osvjetljenja koje daje svjetlo	<code>color = (light.ambient + factor * light.diffuse * nDotL) * material;</code>

Na slikama 3. i 4. nalaze se scene bez i sa preslikavanjem sjena.



Slika 3. Bez preslikavanja sjena



Slika 4. Sa preslikavanjem sjena

Do nazubljenosti sjena dolazi zbog male razlučivosti mape sjena i nedostatka interpolacije njenih vrijednosti. Na slici 5. vidimo da rub sjene ima promjenu kvadratnog oblika što je posljedica projekcije teksture sjena niske razlučivosti i odsustva interpolacije. Rješenje tog problema je korištenje veće razlučivosti za teksturu sjena, omogućavanje interpolacije ili implementacijom mekih sjena koje bi bile ovisne o udaljenosti od ruba sjene.



Slika 5. Nazubljeni rub sjene