3D Grafikus Rendszerek Vizsga

2017-06-11, 10:00 – 12:00

1. Feladat (11 pont)

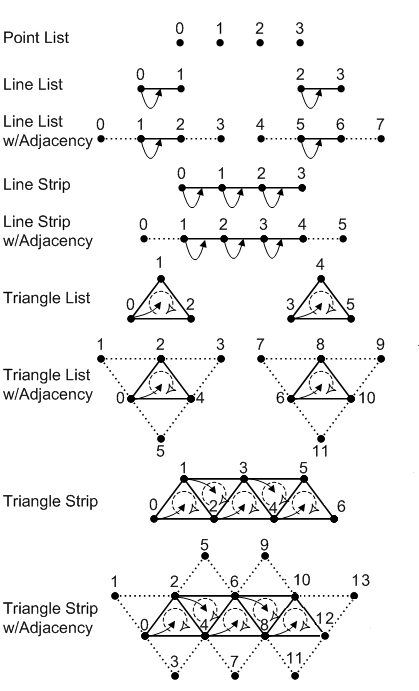
Mi az a tulajdonság-centrikus játékobjektum-modell? Mi az előnye és a hátránya az objektum-centrikushoz képest? Milyen adatszerkezetben érdemes tárolni a tulajdonságleírókat?

1. Feladat (17 pont)

Írja át a következő HLSL geometry shadert úgy, hogy a 3D modell helyett annak árnyéktestét jelenítse meg. A fényforrás irányfényforrás, aminek az iránya konstansként (uniform változó, effekt konstans) rendelkezésre áll. A színtér egy 1000 egység átmérőjű gömbben elfér.

[maxvertexcount(3)]

void gsIdle( triangle VsosTrafo input[3], inout TriangleStream<GsosIdle> outputStream )

{

GsosIdle output = (GsosIdle)0;

output.pos = input[0].pos;

outputStream.Append( output );

output.pos = input[1].pos;

outputStream.Append( output );

output.pos = input[2].pos;

outputStream.Append( output );

}

Lehetséges primitívtípusok listája: point, line, triangle, lineadj, triangleadj.

Lehetséges folyamkimenet-típusok listája: PointStream, LineStream, TriangleStream.

Adja meg továbbá az index buffer tartalmát, amennyiben egyetlen tetraéder árnyéktestét szeretnénk rajzolni, és „háromszögszalag szomszédossággal” primitívtopológiát használunk. Segítségként a dokumentáció releváns ábráját feltüntettük.

1. Feladat (17 pont)

Vázoljon egy megoldást késleltetett árnyalás (deferred shading) megvalósítására. Milyen erőforrásokra van szükség? Milyen menetekből áll a feladat megoldása? Tetszőleges árnyalónyelvet használva adja meg a szükséges árnyalókat. Ahol nincs eltérés a klasszikus forward rendering sémától (MVP transzformáció a csúcspontárnyalóban, BRDF-alapú per-pixel árnyalás a képpontárnyalóban), ott elég ezt jelezni, nem kell az egészet leírni.