



Eötvös Loránd Tudományegyetem
Informatikai Kar
Algoritmusok és Alkalmazásaik Tanszék

Fizikai alapú 3D megjelenítés valósidőben

Valasek Gábor
tanársegéd

Bölöny Zsolt
programtervező informatikus

Dr. Magdics Milán
adjunktus

Budapest, 2016

Tartalomjegyzék

1. Bevezetés	2
2. Felhasználói dokumentáció	4
2.1. kukac	4
3. Fejlesztői dokumentáció	5
3.1. A 3D-s megjelenítés elméleti háttere	5
3.1.1. A megjelenítési egyenlet	5
3.1.2. A megjelenítési egyenlet	5

1. fejezet

Bevezetés

A számítógépes grafikával foglalkozó kutatókat, fejlesztőket napjainkban is foglalkoztatja a kérdés, hogyan lehet a számítógép segítségével minél életszerűbb, *fotorealistikus* megjelenítést létrehozni. A különféle sugárkövetéses technikákkal már régóta képesek vagyunk rendkívül élethű képek létrehozására, de ezek a módszerek nem minden esetben jelentenek megoldást, hiszen a rendelkezésre álló számítási kapacitástól függően a létrehozás folyamata akár napokig is eltarthat. Míg egy animációs film esetében van lehetőség ezt kivárni, addig a valós idejű megjelenítésben nem áll rendelkezésre ennyi idő: a szinte folyton változó, interaktív 3D-s környezetet a másodperc törtrésze alatt kell a képernyőre vetítenünk.

A grafikus kártyák teljesítménye a megjelenésük óta folyamatosan, gyors ütemben fejlődik. Évről évre újabb és újabb technikák jelennek meg, amelyek igyekeznek kihasználni ezt a folyton növekvő teljesítményt annak érdekében, hogy minél látványosabb, minél valóságosabb legyen a virtuálisan létrehozott 3D-s tartalmak megjelenítése. A fizikai alapú megjelenítés is csupán egy, bár annál jelentősebb módszer ezek közül: nem csak a korábban mindenki által használt Blinn-Phong közelítést cseréli le egy, az anyagok valós, fizikai tulajdonságait figyelembe vevő új módszerre, hanem a megjelenítés bemeneteként szolgáló anyagmodellek előállításának folyamatát is átformálja.

A fizikai alapú megjelenítés (hivatalosan *physically based rendering*, a továbbiakban PBR) legfontosabb tulajdonsága az, hogy az anyagok ill. fények mérhető fizikai tulajdonságait veszi alapul, és az adott pontban a kamera által érzékelt fény színet és intenzitását igyekszik ismert, valós fizikai képletek közelítéseinek segítségével meghatározni. Habár a PBR alapelvei és összefüggései létrejötte óta változatlanok, a felhasznált különféle egyenletek minél pontosabb és mindeközben minél kevésbé számításigényes közelítése a mai napig aktív kutatási terület.

A fizikai alapú megjelenítés előnyeinek bemutatására egy egyszerű modellek betöltésére képes megjelenítőt hoztam létre, amelynek segítségével akár egy témában kevésbé járatos felhasználó számára is láthatóvá válnak a módszer előnyei. A

program megalkotása komoly kihívást jelentett, hiszen nem csak a CPU-t, hanem a GPU-t is programoznom kellett árnyalók (*shader*ek) segítségével. A felhasznált technológia újdonsága, az implementáció komplexitása és a várható látványos végeredmény miatt esett választásom erre a feladatra.

2. fejezet

Felhasználói dokumentáció

2.1. kukac

nyehehe

3. fejezet

Fejlesztői dokumentáció

3.1. A 3D-s megjelenítés elméleti háttere

3.1.1. A megjelenítési egyenlet

Tetszőleges térbeli jelenet leképezéséhez alapvetően három dolog szükséges:

- a jelenet geometriájának leírása,
- egy pont a térben, ahonnan "nézzük" a jelenetet, továbbá
- legalább egy fényforrás, ill. annak pozíciója.

Valószerű grafikában a geometriák leírásához háromszöghálókat használunk, mivel a GPU-k térbeli háromszögeken dolgoznak. Domború felületek leírásához ezen háromszögháló felbontását növeljük addig, amíg a végeredmény szempontjából elfogadható közelítést kapunk. Ezt a folyamatot tesszelációnak hívjuk. Az "elfogadhatóság" teljesen szubjektív tulajdonság, így az egyetlen objektív behatároló tényező a grafikus hardver teljesítménye, amelynek folyamatos fejlődése ezen a téren remekül illusztrálható.

Önmagukban a háromszögeket meghatározó térbeli pozíciók még nem elégségesek. Ismernünk kell a felület tetszőleges pontjának irányultságát, amelyet egy, a felületből "kifelé" álló, normalizált vektorral írunk le és (felületi) normálvektornak hívunk. Jelölése: \mathbf{n} . Ezeket legegyszerűbben a háromszögeket alkotó pontokkal együtt tárolhatjuk, majd ezeket menet közben interpolálva kaphatjuk meg a háromszög által meghatározott felület tetszőleges pontján vett normálvektort.

3.1.2. A megjelenítési egyenlet

A számítógépes grafika egyik alaptételének tekinthető megjelenítési egyenletet (*rendering equation*) David Immel et al. és James Kajiya írta le 1986-ban. Az egyenlet segítségével meghatározható a felület egy adott pontját elhagyó sugárzás a felület

által kibocsátott ill. visszavert sugárzás összegének geometriai optika alapú közelítésével:

$$L_0(\mathbf{x}, \mathbf{v}) = L_e(\mathbf{x}, \mathbf{v}) + \int_{\Omega} f_r(\mathbf{x}, \mathbf{l}, \mathbf{v}) L_i(\mathbf{x}, \mathbf{l}) (-\mathbf{l} \cdot \mathbf{n}) d\mathbf{l}$$

ahol

- \mathbf{x} a felület adott pontja
- \mathbf{v} az \mathbf{x} vektorból a nézeti pozícióba mutató vektor (nézeti vektor)
- \mathbf{n} a korábban bevezetett felületi normálvektor
- \mathbf{l} a beérkező fény negatív iránya
- Ω dd
- L_e a felület által kibocsátott fény az adott pontban
- f_r
- L_i

Megjegyzés: az eredeti egyenlet figyelembe veszi még az időt és a fény hullámhosszát is. Az egyenlet paraméterei az idő előrehaladtával ritkán változnak, és ebben az esetben is előre kiszámolhatóak, ezért az időt állandónak tekinthetjük, amely így kiesik az egyenletből. A gyakorlatban RGB színtérben dolgozunk, és ennek 3 összetevőjét külön-külön számoljuk, így a hullámhossz paraméter is elhagyható.

Számítógépes grafikában a kimenetünk színértékek kétdimenziós mátrixa, jellemzően a képernyő pixelei. Ahhoz tehát, hogy kimenetet állítsunk elő, ezen