Sävytinohjelmointi

Janne Timonen

Seminaaritutkielma
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 15. lokakuuta 2015

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen	Tietojenkäsittelytieteen laitos

Tekijä — Författare — Author
Janne Timonen

Työn nimi — Arbetets titel — Title

Sävytinohjelmointi

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Seminaaritutkielma	15. lokakuuta 2015	3

Tiivistelmä — Referat — Abstract

Tiivistelmä.

Avainsanat — Nyckelord — Keywords
avainsana 1, avainsana 2, avainsana 3

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Sisältö

1 Johdanto

Sävyttimet ovat ohjelmia, joiden tehtävänä grafiikkaliukuhihnalla on *sävyttää*, eli tuottaa tietyillä tavoilla dataa kuvaksi. Tämä voi tarkoittaa esimerkiksi jonkin objektin piirtämistä sijainnin mukaan, per-pikseli -värinmääritystä, pinnanmuotojen simulointia tai muita erikoistehostemaisia keinoja. Sävytin ottaa syötteenään yhden elementin, esimerkiksi monikulmion *kärkipisteen*, *primitiivin*, eli monikulmion kuten kolmion, tai *fragmentin*, kuten pikselin, ja tuottaa siitä muunnoksena tulokseksi nollasta useaan dataelementtiä [Gre14].

2 Sävyttimien historiaa

Ennen ohjelmoitavia sävyttimiä grafiikkaa tuotettiin käyttämällä hyväksi näytönohjaimien (GPU) kiinteää liukuhihnaa (Fixed-Function Pipeline) (tätä ennen laskenta tehtiin prosessorissa (CPU), mikä oli hidasta). Kehittäjä saattoi siis antaa raskaat laskutyöt näytönohjaimelle hoidettavaksi kiinteällä liukuhihnalle, mutta itse sen suorittamiin funktioihin ei voinut puuttua kuin parametrien avulla. Kiinteä liukuhihna näytönohjaimessa nopeutti laskentaa, ja toi mukanaan mahdollisuuksia luoda standardioperaatioiden rajoissa efektejä (e.g. Goraud-sävytys).

Myöhemmin tulivat ensimmäiset *ohjelmoitavaa renderointiliukuhihnaa* tukevat näytönohjainpiirit, joissa kiinteän liukuhihnan pystyi korvaamaan omilla vapaasti ohjelmoitavilla sävyttimillä. Korvaamalla kiinteän liukuhihnan laskenta nykyajan grafiikkapiirien tukemilla ohjelmoitavilla sävyttimillä saavutetaan vapaus muokata vapaasti laskenta- ja muokkausoperaatioita, mikä antaa mahdollisuuden piirtää kuvaa enemminkin luovuuden rajoissa, kuin ennaltamääriteltyjen ehtojen. Ensimmäiset sävytinmallit tukivat ainoastaan alemman tason konekielillä ohjelmointia. Sen lisäksi, että kehittäjien täytyi luoda sävyttimen konekielellä, täytyi sävytin luoda lisäksi usein erikseen sekä OpenGL- että Direct3D-rajapinnoille johtuen näiden kahden suosituimman rajapinnan konekielien poikkeavuuksista. Myöhemmin verteksi-, eli kärkipiste-, ja pikselisävyttimet alkoivat yleistyä. Sävyttimien käyttö grafiikkaliukuhihnalla mahdollistaa rinnakkaistamisen erittäin hyvin. [AMH02]

Eräs edelläkävijöitä sävyttimien saralla oli tietokoneanimaatioelokuvistaan tunnettu Pixar-yhtiö kehittämällään *RenderMan*-kielellä, jota käytettiin muun muassa Toy Story -elokuvan tuottamiseen.

3 Ohjelmoitavat sävyttimet

Ohjelmoitavat sävyttimet antavat algoritmeillaan mahdollisuuden muokata vapaasti ja reaaliaikaisesti kuvaa muodostaviin elementteihin liittyviä attribuutteja. Yksi sävytinvaihe ottaa syötteenään vastaan edellisen tulosteen,

joten jokainen vaihe voi jatkaa seuraavan datan työstämistä, kun on saanut edellisen työn valmiiksi. Ohjelmoitavien sävyttimien osalta liukuhihna rakentuu pääpiirteissään kärkipistesävyttimestä, joka antaa laskutuloksensa (vaihtoehtoiselle) geometriasävyttimelle, joka voi luoda jopa uusia geometriaprimitiivejä, ja antaa puolestaan tuloksensa pikselisävyttimelle.

Sävyttimien käyttö mahdollistaa myös hyvin rinnakkaisuuden käytön, kun muunnoksia tehdään suurille datamäärille kerrallaan, esimerkiksi kaikille ruudun pikseleille. Moderneille grafiikkapiireillä onkin useita sävytinliukuhihnoja rinnakkaisuusmahdollisuuksien hyödyntämiseksi.

3.1 Kärkipistesävyttimet

Kärkipistesävytin, tai *verteksisävytin*, ajetaan kerran jokaista monikulmion, tai usein kolmion, kärkipistettä kohden. Kärkipistesävytin ottaa syötteenään kärkipisteen *attribuuttitiedon*, joka sisältää muun muassa kyseisen kärkipisteen sijainnin x-y-z -koordinaatistossa malli- tai maailma-avaruudessa, sekä pinnan normaalivektorin. Tulosteena kärkipistesävytin antaa kärkipisteen, joka on käynyt läpi valaistus- ja muunnosvaiheet, ja joka ilmaistaan nyt normalisoidussa kuvausavaruudessa. Vähintään kärkipistesävyttimen tulee siis antaa tuloksena kärkipiste uniformina tietona. [Puh08]

Yleisesti siis kärkipistesävytin voi muokata monikulmion kärkipisteen, normaalin, tekstuurikoordinaattien ja paikan arvoja.

3.2 Geometriasävyttimet

Geometriasävyttimet on kärkipiste- ja pikselisävyttimiin verrattuna uudempi sävytin sen tultua esitellyksi DirectX 10:n myötä. Geometriasävytin sijaitsee renderöintiliukuhihnalla kärkipistesävyttimen jälkeen, ja ennen pikselisävytintä. Sen käyttö ei ole pakollista, ja vielä usein esimerkiksi pelien sävytyksessä on otettava huomioon jonkinlainen varakeino mikäli geometriasävyttimien käyttö ei ole mahdollista. Geometriasävytin käsittelee kokonaisia primitiivejä, eli muokkaa, valikoi ja jopa luo uusia primitiivejä, kuten pisteitä, suoria ja janoja sekä kolmioita.

3.3 Pikseli-/fragmenttisävyttimet

Pikselisävytin, tai fragmenttisävytin (riippuen sävytinkielen terminologiasta; myöhemmin tekstissä puhutaan pikselisävyttimestä), on graafinen funktio, joka laskee muunnoksia per-pikseli -periaatteella, eli muunnokset voidaan tehdä jokaiselle yksittäiselle pikselille, tai muulle fragmentille, erikseen []. Pikselisävytin ajetaan useita kertoja jokaista syötteenä saatua monikulmiota kohden, sillä sävytin käsittelee jokaista monikulmion pikseliä erikseen. Pikselin väriarvo sekä Z-syvyys lasketaan syötteenä saatun vektorimuotoisen datan, kuten normaalivektorin, värin, tekstuurikoordinaatit, interpoloidut va-

lonlähteiden suunnat ja katsojan suunnan perusteella, perusteella. Erityisesti pikseliin kohdistuva valaistuksen laskenta voidaan johtaa edellisistä [Puh08].

Monet näyttävät 3d-peleissä käytettävät tehostekeinot, kuten pinnan kuhmutus tai Fresnel-heijastus, luodaan juuri pikselisävyttimien tasolla.

4 Korkean tason sävytinkielet

Ohjelmoitavien sävyttimien alkuaikoina oli sävyttimien luomiseen mahdollista käyttää vain alemman tason konekieliin pohjautuvia sävytinkieliä. Korkean tason kielillä on useita hyötyjä konekieliin nähden, ja ne pätevät myös korkean tason sävytinohjelmoinnissa: helpompi luettavuus, kirjoitettavuus, muokattavuus, virheiden etsintä ja löytäminen sekä yleisesti kehitysvauhdin nopeus [She08]. Ohjelmoitavien sävyttimien tultua kasvoi myös tarve korkean tason sävytinkielille, joista mainittavimpina muodostuivat C-pohjaiset Nvidian Cg, Microsoft HLSL ja OpenGL:n GLSL -kielet, joista ensimmäinen on jo deprekoitunut.

4.1 Koodiesimerkkejä

5 ??Sävyttimien pääsy muistiin??

5.1 Sävytinrekisterit

5.2 Tekstuurikartat

Lähteet

[AMH02] Akenine-Möller, Tomas ja Haines, Eric: *Real-Time Rendering.* A K Peters/CRC Press, 2. painos, 2002.

[Gre14] Gregory, Jason: *Game Engine Architecture.* A K Peters/CRC Press, 2. painos, 2014.

[Puh08] Puhakka, Antti: *3D-grafiikka.* Talentum, 1. painos, 2008.

[She08] Sherrod, Allen: *Game Graphics Programming.* Charles River Media, 1. painos, 2008.