VIIKKORAPORTTI 5

Tällä viikolla sain takaseinien poiston (backface culling) toimimaan ja piirtämisen toimimaan joten kuten. Bugeja vielä löytyy ja seuraava urakka onkin niiden liiskaaminen. Kunhan saan piirtofunktion toimimaan kunnolla, onnistuu konveksien kappaleiden piirtäminen täydellisesti, jos kappaleet eivät ole toistensa edessä. Tämän jälkeen jäljellä olisi vain BSP-puu, joka mahdollistaa oikean piirtämisen.

Törmäsin taas uusiin hankaliin ongelmiin, joiden ratkaisut eivät ole tällä hetkellä kauhean elegantteja. Muutenkin koodin ja algoritmien laatu on hieman kärsinyt, sillä aika alkaa käydä vähiin. Esimerkiksi pinnan normaalien laskemisessa olen jättänyt pintavektorit normalisoimatta, jolloin myös itse normaalin pituus vaihtelee. Se ei onneksi vaikuta ainakaan tällä hetkellä mihinkään, mutta olisi hyvä kuitenkin toteuttaa. Takaseinien poistoa en ole toteuttanut aivan standardin mukaan, sillä oma koordinaatistoni on tavallisesta poiketen vasenkätinen, eikä oikeakätinen. Tästä syystä joudun tarkistamaan monet laskut käsin ja merkkivirheitä tulee paljon, referenssimateriaalin seuraaminen on myös välillä hankalaa. Koordinaatiston kätisyyden muuttaminen tässä vaiheessa tietäisi melko paljon refaktorointia ja laskujen sekä algoritmien tarkistamista, joten en tule sitä toteuttamaan.

Funktioiden rivimäärät ovat hieman karanneet käsistä. Tämä ei ole hyvä kehityskulku, mutta jotenkin deadlinen lähestyessä tulee aina otettua ”teknistä velkaa” yhä enemmän ja enemmän. Koodin laatu ei ole enää ihan sitä, mitä itse haluaisin, mutta aikaa ei oikein riitä kunnollisen refaktoroinnin tekemiseen. Syitä voi myös löytyä omasta tavastani kirjoittaa koodia. Kirjoitan yleensä aluksi pitkän funktion, joka toteuttaa jonkun tietyn asian ja alan sen jälkeen refaktoroimaan ja pilkkomaan koodia osiin ja selkeyttämään sitä. Monesti sama koodinpätkä tulee kirjoitettua parikin kertaa. Tämä takaa hyvän laadun koodille, mutta on ehkä hieman hidasta. Ohjelman parempi suunnittelu auttaisi tähän. Toisaalta esimerkiksi juuri piirtofunktion on toteutettava niin monta asiaa, että koodin pilkkominen vielä pienempiin osiin ainakin omasta mielestäni vaikeuttaisi koodin lukemista. Parempi vaihtoehto on varata tietorakenteisiin tilaa laskujen tuloksille tai luoda niitä varten omia tietorakenteita. Olen esimerkiksi siirtänyt paljon laskettuja koordinaatteja verteksi- ja polygoni-tietorakenteisiin, mikä on mielestäni hyvä ratkaisu. Myöskin pienimmälle kolmion sisältävälle neliölle olen luonut oman tietorakenteen.

Kamerasta poispäin osoittavien seinien poistaminen on loppujen lopuksi melko helppoa. Se toteutetaan yleensä määrittämällä polygonin vertekseille kiertosuunta, joka määrää myös normaalin suunnan. Jos verteksit kiertävät polygonin keskipistettä myötäpäivään, osoittaa normaali kameraan päin, jos vastapäivään, osoittaa normaali poispäin kamerasta. Polygonit, jotka osoittavat poispäin kamerasta ovat todennäköisesti muiden polygonien takana, joten ne voidaan poistaa kuvasta. Kun polygoneille on määritelty normaali, voidaan normaalin ja kamerasta polygoniin osoittavan vektorin välinen kulma laskea. Jos tämä on positiivinen, osoittaa polygoni poispäin ja se poistetaan kuvasta.

Toinen ongelma, johon törmäsin, on polygonin piirtäminen. Ongelman ratkaisu vaatii algoritmia, joka selvittää, onko piste annetun polygonin sisällä. Oma algoritmini on seuraavanlainen: Ohjelma laskee pienimmän neliön, joka sisältää annetun kolmion. Tämän jälkeen se käy läpi kaikki pisteet neliössä ja tarkistaa, onko se polygonin sisällä. Tarkistamiseen käytän kolmiolle räätälöityä ”Crossing numbers”-algoritmia. Se tarkistaa jokaiselta pisteeltä, kuinka moneen viivaan pisteestä oikealle piirretty vektori törmää. Jos määrä on parillinen, piste on kolmion ulkopuolella, jos pariton, on se kolmion sisäpuolella. Tämä algoritmi vaatii viivojen leikkauspisteen laskemista. Käytän tähän yksinkertaista

jakolaskua, mikä ei ole lopulta kannattavaa sen hitauden vuoksi. Parantamisen varaa jää siis tähänkin asiaan.

BSP-puun toteuttamiseen tarvitaan myös viivojen ja tasojen leikkauspisteiden laskemista, joten joudun todennäköisesti kirjoittamaan ensi viikolla kunnollisen matematiikkakirjaston. Samalla voin hieman parannella myös piirtofunktion toiminnallisuutta. Itse BSP-puun toteuttaminen saattaa mennä tiukille, mutta yritän saada senkin toteutettua. Hieman hölmöltä tuntuisi koodata koko kurssi algoritmia varten, jota ei loppujen lopuksi ehtisi valmistelevan koodin monimutkaisuuden takia toteuttaa ☺ Projekti on ollut työläs, mutta erittäin mielenkiintoinen. Normaalin jakson aikana en varmasti olisi panostanut tähän näin paljon, mutta näin kesäpäivien ratoksi tämä on sopinut hyvin. Tämän viikon perjantaina toivon saavani piirtofunktion täysin toiminnalliseksi, tällä hetkellä se on vielä hieman buginen. Osallistun mielelläni kuvakollaasiin ja yritän viikonlopun aikana saada renderöityä muutaman tyylikkään kuvan. Kuvat löytyvät github repon renders-kansiosta, valitkaa sieltä parhaimmat.