Projektin dokumentti

Miska Kananen (652102, TiK, 1.vk)

21. huhtikuuta 2019

Sisältö

1 Yleiskuvaus

Toteutin ohjelman, jolla voi renderöidä reaaliajassa kolmioista koostuvia maailmaan sijoitettuja kolmiulotteisia malleja. Mallit voivat olla mielivaltaisia, eivätkä ne ole rajoittuneita esimerkiksi pelkästään suorakulmaisiin seiniin. Mallin ympärillä voi liikkua vapaasti ja kameraa voi kääntää mielivaltaiseen suuntaan.

2 Käyttöohje

Kun ohjelma käynnistetään, se lataa mallinnettavan maailman `world`-nimisestä tekstitiedostosta ohjelman juurikansiosta. Jos lataaminen epäonnistuu, tulostuu tieto tästä standarditulosteeseen.

Kameraa voi liikuttaa eteen, taakse ja sivuille `W, A, S, D`-näppäimillä. `Q` ja `Z` liikuttavat kameraa ylös ja alas. Nuolinäppäimet kääntävät kameraa.

3 Ohjelman rakenne

Luokka ***Vertex*** kuvaa kolmion nurkkapistettä. Kullakin *Vertexillä* on:

- sijainti object spacessa (yksittäisen kappaleen omassa koordinaatistossa)
- väri

Luokka ***Triangle*** kuvaa kolmesta *Vertexistä* koostuvaa kolmiota. *Triangle* on avaruuden kappaleiden rakennuspalikka ja yksinkertaisin renderöitävissä oleva primitiivi. *Trianglella* on:

- kolme *Vertexiä*, eli kolmion nurkkapisteet object spacessa
- materiaali (kertoo miten kolmio piirretään: umpinainen, wireframe)

Luokka ***Model*** mallintaa yksittäistä, kolmioista koostuvaa kappaletta. Kolmiot on aseteltu kappaleen omaan object spaceen. *Modelilla* on:

- yhden tai useamman *Trianglen* muodostama mesh eli käytännössä lista kolmioista, jotka muodostavat yhdessä kappaleen
- metodit, joilla voi asettaa materiaalin ja värin kerralla koko meshille

Luokka ***CollisionBox*** kuvaa kappaleen muotoa törmäystunnistuksen kannalta. *CollisionBox* on koordinaattiakselien suuntainen suorakulmainen särmiö, jonka läpi liikkuminen ei ole sallittua. *CollisionBoxilla* on kaksi nurkkapistettä, jotka määrittävät särmiön muodon. Pisteet ovat kappaleen object spacessa, eli samassa koordinaattijärjestelmässä kuin kappaleen *Modelin* verteksit.

Luokka ***WorldObject*** kuvaa maailmassa olevaa objektia, kuten esimerkiksi seinää. *WorldObjectilla* on

- *Model*, joka määrittää kappaleen muodon ja ulkonäön
- mahdollinen *CollisionBox*, joka kuvaa kappaleen muotoa törmäystunnistuksen kannalta
- metodi, jolla voi kysyä, onko jokin piste kappaleen sisällä
- Transform, eli 4x4-matriisi (world matrix), joka kuvaa kappaleen *Modelin* ja *CollisionBoxin* verteksit object spacesta world spaceen. Matriisin avulla asetetaan kappale avaruuteen haluttuun paikkaan ja asentoon.

Luokka ***World*** kuvaa mallinnettavaa avaruutta ja pitää sisällään kaikki avaruudessa olevat kappaleet. *Worldilla* on

- lista *WorldObjecteista* sopivine Transformeineen, eli maailmassa olevat kappaleet oikeilla paikoillaan
- *Camera*, eli kamera, jonka näkökulmasta maailma renderöidään
- metodi, jolla voi kysyä, onko jokin piste maailman kappaleen sisällä

Luokka ***Camera*** määrittää, mistä paikasta maailmaa tarkastellaan ja mihin suuntaan katsotaan. Kamera on aina renderöitäessä oman koordinaatistonsa (view spacen) origossa osoittaen $+z$-suuntaan, muuta maailmaa liikutetaan ja käännetään tarpeen mukaan. *Cameralla* on:

- Transform, eli 4x4-matriisi (view matrix), joka kuvaa maailman *Modelien* verteksit world spacesta view spaceen siten, että kamera on origossa $+z$-suunnassa
- metodi view matrixin rakentamiseen siten, että kamera on maailmaan nähden halutussa paikassa ja osoittaa haluttuun suuntaan

Luokka ***Renderer*** ottaa renderöitäväksi **Worldin** ja kuvaa sen **Modeleiden** verteksit *Cameran* view spacesta clip spaceen, eli koordinaatistoon, jossa tarkastellaan verteksien näkyvyyttä. Tämän jälkeen *Renderer* selvittää, mitkä *Triangleista* ovat kameran näkökentässä, hylkää sen ulkopuolella olevat, leikkaa sopivasti osittain näkyvissä olevia ja lopulta projisoi koko näkymän clip spacesta image spaceen, eli kaksiulotteiselle tasolle. *Renderer* luo z-koordinaatin mukaan järjestetyn listan piirrettävistä *Triangleista*, jotka on projisoitu perspektiiviprojektiolla xy-tasolle siten, että niiden x- ja y-koordinaatit on normalisoitu välille $[-1.0, 1.0]$.

Luokka ***Screen*** kuvaa näkyvää 2D-pikseliruudukkoa, johon kuva piirretään. *Screen* ottaa *Rendereriltä* listan projisoiduista *Triangleista*, kuvaa niiden verteksit väliltä $[-1.0, 1.0]$ pikselikoordinaatteihin ja piirtää ne näkyviin ruudulle materiaalit ja värit huomioiden.

Luokat ***Vector4 ja Matrix4*** kuvaavat laskennassa käytettäviä neliulotteisia vektoreita ja matriiseja ja toteuttavat niille tarvittavat laskutoimitukset.

Luokka ***WorldLoader*** lukee maailman tiedostosta ja luo sen pohjalta *Worldin* ilmentymän.

Luokka ***Engine*** käynnistää ohjelman, luo käyttöliittymän, kuuntelee käyttäjän syötettä ja päivittää grafiikkaa ja logiikkaa tasaisin väliajoin.

Luokka **Constants** sisältää ohjelmassa käytettäviä vakioita.

Luokka **Helpers** sisältää usein tarvittavia apufunktioita.

4 Algoritmit

5 Tietorakenteet

6 Tiedostot

Renderöitävä maailma ladataan `world`-nimisestä tekstitiedostosta, jonka rakenne on seuraava.

Ensimmäisellä rivillä on 3 desimaalilukua: kameran x, y ja z-koordinaatti alussa.

Toisella rivillä on ei-negatiivinen kokonaisluku n: kappaleiden määrä.

Seuraa n kappaleen kuvausta. Kunkin kuvauksen rakenne on seuraava:

Kuvauksen ensimmäisellä rivillä on kolme desimaalilukua x, y ja z: kappaleen keskipisteen (model spacen origon) sijainti world spacessa.

Kuvauksen toisella rivillä on ensin kappaleen tyyppi, joka on joko `SOLID` tai `WIREFRAME`. Tyypin jälkeen samalla rivillä on neljä kokonaislukua r, g, b, a ($0 \leq r, g, b, a \leq 255$), jotka kuvaavat kappaleen värin ja läpinäkyvyyden.

Kuvauksen kolmannella rivillä on epänegatiivinen kokonaisluku m: kappaleen kolmioiden määrä.

Seuraa m riviä, joista kullakin on 9 desimaalilukua $x1$, $y1$, $z1$, $x2$, $y2$, $z2$, $x3$, $y3$ ja $z3$: kolmion verteksien koordinaatit.

Esimerkki kelvollisesta tiedostosta, jossa on yksi yhdestä kolmiosta koostuva kappale:

```
0 0.5 0
1
0 0 0
WIREFRAME 255 0 0 255
1
-1 -1 0 0 1 0 1 -1 0
```

7 Testaus

8 Puutteet ja viat

9 Parhaat ja heikoimmat kohdat

10 Poikkeamat suunnitelmasta

11 Kokonaisarvio

12 Viitteet