

VRML-TIEDOSTOJEN LUONTI JA KÄYTTÖ VISUALISOINNISSA

LAHDEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Teknisen visualisoinnin suuntautumisvaihtoehto
Opinnäytetyö
9.5.2006
Timo Luntta

**Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma**

LUNTTA, TIMO: VRML-tiedostojen luonti ja käyttö visualisoinnissa

Teknisen visualisoinnin opinnäytetyö, 61 sivua, 6 liitesivua
Ohjaaja: Ismo Jakonen

Kevät 2006

TIIVISTELMÄ

Opinnäytetyössä perehdytään VRML-kielen mahdollisuuksiin visualisoinnissa. VRML on monipuolinen mallinnuskieli virtuaalimaailmojen visualisoimiseen. Se tukee monia ominaisuuksia, kuten valaistus, värit, animaatio, kamerat ja äänet.

Opinnäytetyö käsittelee VRML-kielen tukemia sekä sisältämiä visualisointiominaisuuksia käyttämällä niitä 3D-mallinnusohjelmalla luotujen mallien visualisoimiseen ja viemiseen VRML-tiedostoksi. Lisäksi työssä vertaillaan VRML-kieltä X3D-kieleen sekä niiden vientiliitännäisten eroja.

Teoriaosuudessa käsitellään VRML-mallinnuskielen perusteita, luomista ja käyttöä sekä X3D-kielen taustaa. Teoriaosuuden jälkeen tavoitteena oli käyttää VRML-kielen tukemia visualisointiominaisuuksia 3ds Max -ohjelmalla luotujen mallien visualisointiin sekä tallennettiin mallit vientiliitännäisellä VRML-tiedostoksi. Lisäksi mallit tallennettiin X3D-tiedostoiksi X3D-vientiliitännäisellä, jotta oli mahdollista vertailla vientiliitännäisten 3ds Max -ohjelmasta mukanaan viemiä ominaisuuksia.

Nykyään ainoa käyttökelpoinen VRML-kielen ominaisuus on animaatio, esimerkiksi hierarkia-animaatiot ja pintaverkkoanimaatiot. Teksturointiominaisuudet ovat X3D-kielessä paljon kehittyneemmät kuin VRML-kielen. Animaation vientiominaisuudet ovat samanlaiset X3D-kielessä kuin VRML-kielessä.

Avainsanat: VRML, X3D, HTML, 3ds Max

**Lahti University of Applied Science
Faculty of Technology**

LUNTTA, TIMO: Creation and use of the VRML language in visualization

Bachelor's Thesis in Visualization Engineering, 61 pages, 6 appendices
Supervisor: Ismo Jakonen

Spring 2006

ABSTRACT

The objective of the thesis was to explore the possibilities of the VRML language in visualization. VRML is a versatile language to visualize virtual environments. It supports many properties, for example lighting, colours, animation, cameras and sounds.

The theory of the thesis explores the basics, creation and use of the VRML language and also examines the X3D language. The aim of the empirical part was to implement a model with 3D modeling software which then was exported to a VRML document by using an export plug-in. Exporting to X3D document was done similarly.

The thesis compares the VRML language with the X3D language and the differences between their exporting plug-ins. The thesis explores the visualization properties which the VRML language contains and supports. The properties were used to visualize models created with the 3ds Max program and to export the models to VRML document. Also the visualization properties which X3D language supports were explored with the same software and the models were exported to X3D document.

Today the only useful property of VRML is animation, for example animated hierarchies and animated meshes. The texturing properties of the X3D language are further developed than in the VRML language. Exporting properties are exactly the same in VRML and X3D.

Keywords: VRML, X3D, HTML, 3ds Max

SISÄLLYS

1	JOHDANTO	1
2	VIRTUAALITODELLISUUS	2
2.1	Määritelmä	2
2.2	Virtuaalitodellisuuden historia.....	2
2.3	Virtuaalitodellisuuden käyttökohteita	3
2.31	Virtuaaliprototyypit	3
2.32	Koulutus	5
2.33	Sotilaalliset sovellukset.....	5
3	VRML-KIELEN KEHITYS	6
3.1	Yleistä.....	6
3.2	VRML 1.0.....	6
3.3	VRML 2.0.....	7
3.4	VRML97	7
3.5	X3D.....	7
3.6	VRML:n leviämiseen vaikuttaneita asioita	8
4	VRML-KIELI.....	11
4.1	VRML-tiedosto.....	11
4.2	Syntaksi	11
4.3	Perusgeometria	14
4.31	Perussolmut	14
4.32	Box-solmu	16
4.33	Cone-solmu	17
4.34	Cylinder-solmu	17
4.35	Sphere-solmu	17
4.36	Kolmiulotteinen teksti.....	18
4.37	Perusgeometrian esimerkit	18
4.4	Objektin siirto, kierto ja skaalaus	20
4.5	Teksturointi	21
4.51	RGB-värit	21
4.52	Kuvatiedostot	22
4.6	Varjostus.....	23
4.7	Valaistus	23
4.8	Optimointi	24
5	VRML:N LUONTI JA KÄYTTÖ	26
5.1	VRML-mallin luonti 3D-grafiikkaohjelmalla.....	26
5.11	Geometria	26
5.12	Animaatio.....	27
5.13	Teksturointi	27
5.2	Vientiliitännäiset	28
5.21	Octaga Exporter	28
5.22	VRML97 Exporter	29
5.23	Web3D X3D Exporter	30
5.3	Selaimet	31
5.31	Yleistietoa	31
5.32	Liitännäiset ja katseluohjelmat	32
5.33	Navigointi ja vuorovaikutus	33

5.34	VRML-tiedoston upottaminen HTML:ään	34
5.35	Kehykset	34
5.36	Näkymä	35
5.37	VRML-selaimet ja -liitännäiset	35
6	CASE: VRML-mallin visualisointi	37
6.1	Yleistä	37
6.2	Teksturointi	37
6.3	Animaatio	42
6.4	Partikkelit ja NURBS:t	44
7	YHTEENVETO	46
	LÄHTEET	47
	LIITTEET	49

TERMISTÖ

3D (three dimensional): kolmiulotteinen.

Bump mapping: tekniikalla esineiden pinta saadaan kuvissa näyttämään epätasaiselta, esimerkiksi kolhuiselta. Objektin ääriviivoihin tekniikka ei vaikuta, eikä epätasaisuuksista aiheudu varjoja.

HTML (Hypertext Markup Language): avoimesti standardoitu kuvauskieli, jolla voidaan kuvata hyperlinkkejä sisältävää tekstiä.

IEC (the International Electrotechnical Commission): sähköalan kansainvälinen standardointijärjestö.

ISO (the International Organization for Standardization): kansainvälinen standardointiorganisaatio.

NURBS (Non-Uniform Rational B-Splines): Kontrollisäikeet (engl. Splines) on yksi tapa ilmaista laskennallista käyrää. Kaareva pinta jatkuu yhtenäisenä kontrollipisteiden ohitse, kulkematta niiden kautta.

Polygoni (polygon): monikulmio, suljetussa tasossa oleva suorien viivojen rajoittama muoto. 3D-grafiikassa kuvat muodostuvat polygoneista.

RGB colors: RGB-värimalli on väriavaruus, jossa eri värejä muodostetaan yhdistelemällä punaisen, vihreän ja sinisen värisiä valonlähteitä.

Renderöinti (rendering): 3D-mallin materiaalien laskeminen tietokoneella ääriviivoina tai pintoina 2D-kuviksi.

Tiling: 2D-kuvan monistamista siten, että se peittää suurenkin pinnan.

URL (Uniform Resource Locator): käytetään osoittamaan WWW-sivuja.

UTF (Universal Transform Format): UTF-8 on Unicoden vaihtelevanpituisen koodaustapa.

Verteksi (vertex): kärkipiste, joka on yksi kolmesta kärkipisteestä kolmiikulmiossa.

VRML (Virtual Reality Markup Language): WWW-tekniikan laajennus, jonka avulla voidaan esittää kolmiulotteisia tiloja ja kappaleita.

VRML (Virtual Reality Modeling Language): avoin ohjelmointikieli, jonka avulla kolmiulotteista virtuaalimaailmaa voidaan julkaista Internet-sivulla.

1 JOHDANTO

Internetin leviäminen loi mahdollisuuden kehittää uusia ja jo olemassa olevia tekniikoita hyödyntämään Internetin levinneisyyden laajuutta ja kykyä siirtää tietoa laajalle käyttäjäryhmälle. Eräs Internetin levityskanavakseen valinnut tekniikka oli virtuaalitodellisuus. Virtuaalitodellisuudessa ihmisen toimintaa eivät ole rajoittamassa reaali maailmasta tutut fysiikan lait. Virtuaaliympäristössä voidaan toteuttaa todellisuudesta kokonaan riippumattomia maailmoja tai hyvin realistisia simulaatioita, joiden toteuttaminen käytännössä olisi työlästä.

Vaikka virtuaalitodellisuuden tutkimus ja kehitys oli alkanut jo 1960-luvulla, niin virtuaalitodellisuus ja Internetin ensimmäinen merkittävä kohtaaminen tapahtui vasta 1990-luvun puolivälissä. Tällöin alettiin kehittää Internetin kautta välitettävää kolmiulotteista mallia kuvaavaa kieltä – VRML-kieltä.

VRML-kieli on ensimmäinen kolmiulotteisen tilan mallinnuskieli, jolla tehtyjä tekstipohjaisia tiedostoja pystyi lataamaan Internetistä. VRML on lyhenne sanoista Virtual Reality Modeling Language, joka tarkoittaa vapaasti suomennettuna keinotodellisuuden mallintamiskieltä. Kieli on tehty kuvaamaan monen käyttäjän interaktiivisia ympäristöjä, simulaatioita ja virtuaali maailmoja, jotka sijaitsevat Internetissä ja on linkitetty WWW-maailmaan.

Opinnäytetyön tarkoitus on selvittää VRML-kielen perusobjektien luomisen teoriaa, käytäntöä ja niiden julkaisemisen Internet-sivulla. Lisäksi työssä vertaillaan kielen visualisointiominaisuuksia kehittyneempään X3D-kieleen. VRML-kielen laajuuden vuoksi on muutamia asioita jätettävä pois, kuten apuobjektit ja uusien solmujen luominen.

Työssä tutkitaan VRML-kielen tukemia visualisointiominaisuuksia käyttämällä niitä 3ds Max -ohjelmalla luotujen mallien visualisointiin sekä niiden siirtymistä VRML-tiedostoon. Lisäksi tutkitaan myös X3D-kielen tukemia visualisointiominaisuuksia samalla ohjelmistolla ja niiden siirtymistä X3D-tiedostoon. VRML- ja X3D-tiedostojen viennissä käytettiin VRML97 Exporter, Octaga Exporter -ja Web 3D X3D Exporter -vientiliitännäisiä. Tallennettujen tiedostojen katseluun käytetään Octaga Professional -selainta sekä Java-kielillä toteutettua Xj3D-selainta.

2 VIRTUAALITODELLISUUS

2.1 Määritelmä

Virtuaalitodellisuudella tarkoitetaan yleensä tietokoneella luotua komiulotteista ympäristöä, jossa käyttäjä voi liikkua. Tavoitteena virtuaalitodellisuudessa oli luoda käyttäjälle illuusio virtuaalisesta maailmasta. Maailma, jossa käyttäjä voi olla vuorovaikutuksessa ja kokea usean aistin avulla olevansa sekä henkisesti että fyysisesti uppoutuneena maailmaan. Alussa tuotettiin visuaalista informaatiota erilaisilla näyttölaitteilla ja virtuaalitodellisuuden kehityksen mukana pyrittiin lisäämään tunto- ja kuuloaistien hyödyntäminen. (Hintikka 1994, 60.)

Virtuaalitodellisuus on havainnoimisen ja ihmisen reaktioiden vaikutusten kautta saatu vaikutelma todentuntuisesta olotilasta mutta ei todesta olotilasta (Reitmaa, Vanhala, Kauttu & Anttila 1995, 3). Todentuntuinen näköaistimus, jossa käyttäjä tuntee olevansa sisällä ja läsnä tietokoneen luomassa todentuntuisessa ympäristössä. Käyttäjä tuntee kykenevänsä toimimaan virtuaaliympäristössä vuorovaikutteisesti ja käsittelemään monimutkaista tietoa. (Reitmaa ym. 1995, 2.) Tuntomerkkeinä pidetään läsnäolon tunnetta (immersio), sekä toisaikaista vuorovaikutteisuutta (Reitmaa ym. 1995, 4). Lisäksi käyttäjän sijainti, liikkeet ja toimenpiteet vaikuttavat niin havaintoihin, kuin näennäisiin esineisiin ja koko näennäiseen maailmaan (Reitmaa ym. 1995, 20). Käsitteeseen liitetään myös autonomisuus: käyttäjä on vapaa tai ainakin tiettyyn rajaan asti vapaa toimimaan ilman erityistä laitteiston aiheuttamia rajoituksia (Reitmaa ym. 1995, 3).

Eräs tapa esittää tietokonepohjaisia virtuaaliympäristöjä on VRML-kieli, jota käytetään ympäristön esittämiseen. VRML-kieli on tiedostomuoto, jolla voidaan luoda kolmiulotteisia objekteja ja ympäristöjä. VRML-teknologia on suunniteltu käytettäväksi Internet-sivuilla, sisäisissä verkoissa, ja paikallisissa järjestelmissä. VRML-kieli on suunniteltu maailmanlaajuisesti tiedonsiirtomuodoksi 3D-grafiikalle ja multimedialle. VRML-ympäristöillä on paljon eri käyttökohteita, joissa vain ihmisen kyvyt ovat rajana. VRML-kieli mahdollistaa staattisen ja animoidun dynaamisen 3D- ja multimediaobjektien esittämisen, joissa on hyperlinkitys toisiin medioihin, kuten esimerkiksi tekstiin, ääni-, kuva ja video-tiedostoihin. (Web3d 2006a.)

2.2 Virtuaalitodellisuuden historia

Virtuaaliympäristöjen idea ja tutkimus alkoivat 1960-luvulla. Virtuaalitodellisuuden kehityksen katsotaan alkavan Ivan Sutherlandin vuonna 1965 julkaisemasta The Ultimate Display -konferenssipaperista. Konferenssipaperissa Sutherland ehdotti, että koneeseen liittämistä ja näyttötekniikoita kehitettäisiin realistisempaan suuntaan. Tällöin syntyisi mahdollisimman uskottava realismi, ikkuna virtuaalimaailmaan. Virtuaalitodellisuuden kehitys, ku-

ten moni muukin tietotekniikan alue on ollut 1960-luvulta alkaen läheisessä yhteydessä Yhdysvaltain sotateknologian kehitykseen. (Reitmaa ym. 1995, 7.)

Vuonna 1956 ennen Sutherlandin ajatusta, oli Morton Heilig'n huvipuis-tossa käytössä Sensorama-kone. Koneella pääsi ajamaan moottoripyö-räsimulaattorilla New Yorkin kaupunki maisemassa. Laitteen immersio to-teutettiin kolmiulotteisella kuvalla, moottoripyörän äänellä, rungon tärinöillä ja pakokaasun hajulla sekä ohjattiin ilmavirtaa kasvoille. (Reitmaa ym. 1995, 7.) Sutherlandin ensimmäinen näyttölaite on vuodelta 1968. Näyttö-laitteessa pään liikkeiden seuranta tapahtui vivustolla, jotta tietokone oli selvillä koko ajan katselupisteen paikasta. (Reitmaa ym. 1995, 7.)

Avaruustutkimus oli ensimmäinen todellinen virtuaaliympäristöjen potenti-aalin oivaltaja. Yhdysvaltojen avaruushallinto (NASA) rakensi oman neste-kidenäytön ja LEEP-optiikkaan perustuvan kypäränäytön vuonna 1984. Vuodesta 1984 lähtien NASA Ames Research Center on ollut virtuaaliymp-äristöjen vahvana tutkija. Tavoitteet ovat olleet virtausten visualisointi, ast-ronauttien työolojen esittäminen koulutustarkoituksessa sekä teleoperointi. (Reitmaa ym. 1995, 7.) Samoihin aikoihin alkoi näkyä lisää ilmiselviä so-vellusmahdollisuuksia, kuten esimerkiksi lentokoneella lentämisen simu-lointi ja taistelulentäjän informaationhallinnan parantaminen (Reitmaa ym. 1995, 9). 1990-luvun alkaessa virtuaalialan tutkimuksen tehtävän määrän-pää oli vakiintunut. Määränpäänä oli ihmisen aistien ja reaktioiden yhä tar-kempi kytkeminen synteettiseen ympäristöön. (Reitmaa ym. 1995, 10.)

2.3 Virtuaalitodellisuuden käyttökohteita

2.31 Virtuaaliprototyypit

Elektroniikkatuotteen konseptisuunnittelu vaatii nopeasti rakennettavia di-gitaalisia malleja tuotteen visualisointiin. Virtuaaliprototyyppi Internetissä on uusi tapa nopeuttaa tuotekehitysprosessia. Digitaalisena esitettävä tuote-malli on saatavilla kaikkialla ja lisäksi siihen voidaan rakentaa myös tuot-teen toiminnallisuutta ja yhteiskäyttöä tukevia ominaisuuksia. (Tuikka, Kert-tula, Salmela & Paasovaara 1999.)

Virtuaaliprototyyppi on tuotteen digitaalinen malli, jolla voidaan todentaa tuotteen ominaisuudet jo ennen varsinaisen fyysisen mallin valmistamista. Virtuaaliprototyyppi on mahdollisimman todenmukainen malli tuotteesta, jo-ka voidaan luoda CAD-mallista. Se on kolmiulotteinen, sitä voi katsella eri kulmista ja se on toiminnallinen. Virtuaaliprototyypin oleellisia piirteitä on helppo tutkia www-sivulla, olivat piirteet sitten mekaanisia, toiminnallisia tai muotoiluun liittyviä seikkoja, lisäksi on helppo tehdä muutoksia ja tutkia mil-tä ne näyttävät. (Tuikka ym. 1999.)

Virtuaaliprototyypin jakaminen on myös tärkeä ominaisuus. Suunnittelijalla saattaa olla monia eri tuotekonsepteja, joita halutaan tutkia samanaikaisesti. Jakamalla prototyypin suunnittelija voi ottaa asiakkaan prototyypin ohjaukseensa omalta koneeltaan ja esitellä toimintoja. Tietoverkon kautta toimiva virtuaaliprototyyppi voidaan ottaa käyttöön missä tahansa, missä on Internet-yhteys. Tulevaisuudessa täysdigitaalinen tuotekonsepti voidaan viedä nopeasti käyttöliittymän ja sulautetun ohjelmiston testaukseen, tuotantoon ja markkinoille. (Tuikka ym. 1999.)

Virtuaaliprototyyppien mahdollisia käyttökohteita on laajalti, muun muassa tuotemarkkinointi, tuotteen simulointi ja tuotekehityksen kommunikointi, tuotetiedon hallinnan käyttöliittymä, tuotesuunnittelusuhteen luominen ja ylläpito sekä tuotteen loppukäyttäjän liittäminen tuotekehitysprosessiin. Virtuaaliprototyyppiin liitettävät laitetekniikat mahdollistavat myös tuntoaistimukset haptisella eli tunto- ja voimavasteen toteuttavalla käyttöliittymällä, jolla voidaan todeta esimerkiksi pinnan kovuus, kappaleen dimensiot, tekstuuri, kitka ja ääriä. (Tuikka ym. 1999.)

Älykkäät virtuaaliprototyypit (engl. smart virtual prototypes) kehitettiin Java-pohjaisena, jotta tuotemalleja voitaisiin käyttää verkossa ympäristöstä riippumatta. Tavoitteena oli myös luoda sovellusalusta, johon voidaan rakentaa virtuaaliprototyyppiä tukevia kehittyneitä työkaluja, sovelluksia ja palveluita. Eräs merkittävä älykkäiden virtuaaliprototyyppien vahvuus on niiden tuki heterogeenisille prototyypeille. Ne ovat suoritettavissa olevia prototyyppejä, jotka koostuvat eritasoisista simulointimalleista loogisen tason malleista toteutustason malleihin asti. Älykkäiden virtuaaliprototyyppien Java-pohjaiset komponentit tarjoavat mekanismin ulkopuolisten simulaatioiden, ohjelmistojen ja jopa laitteiden kytkemisen tuotemalleihin. Heterogeeniset prototyypit tulevat tulevaisuudessa mahdollistamaan saumattoman siirtymisen konseptisuunnitteluvaiheen loogisista simulointimalleista valmiiseen tuotteeseen, jolloin tuotteen virtuaalimallin ja lopullisen tuotteen rajat hämärtyvät. (Tuikka ym. 1999.)

Teknologia mahdollistaa toiminnalliset ja todenmukaiset tuotemallit, jotka voidaan tarvittaessa jakaa kahteen osaan. Tuotteen interaktiivinen 3D-malli voidaan ladata VRML-mallin ja erityisen Java appletin muodossa käyttäjän web-selaimeen. Virtuaalimalliin liittyvät erilaiset simulointimallit ja sovellusagentit sen sijaan voidaan sijoittaa suoritettaviksi web-palvelimessa esimerkiksi tietoturvallisuuden takaamiseksi. Digitaalisiin tuotekomponentteihin voidaan liittää erilaisia älykkäitä toimintoja, ohjelmistoagentteja tai hypertextipohjaista informaatiota. (Tuikka ym. 1999.)

2.32 Koulutus

Virtuaalimallien avulla voidaan helposti havainnollistaa usein hyvin käsitteellistä opetettavaa tietoa. Kolmiulotteinen havainnollistaminen tuo uusia menetelmiä ja apuvälineitä muun muassa uusien matematiikan alojen, kuten esimerkiksi kaaos- ja fraktaalitutkimusten apuvälineinä. Virtuaalikirjastojen avulla kaikenlainen informaatio on helposti saatavilla.

Kemiallisessa tutkimuksessa voidaan myös hyödyntää kolmiulotteisuutta. Voidaan havainnollistaa ja luoda uusia rakenteita ohjelman samalla laskiessa erilaisia sidosarvoja. Esimerkiksi Pohjois-Carolinan yliopistossa tutkijat ovat kokeilleet erilaisia menetelmiä molekyylien tutkinnassa ja uusien lääkkeiden kehittämistä virtuaalisesti, joka ei ole ollut mahdollista aiemmin. (Hintikka 1993, 85.) Virtuaalimallintamista voidaan käyttää hyödyksi myös fysiikassa. Fysiikalle on tyypillistä suuren tietomäärän käsittelyyn ja havainnollistamiseen tarkoitetut sovellukset, esimerkiksi energia-aaltojen simulointi, aerodynamiikan laskut ja nesteen virtaukset (Reitmaa ym. 1995, 112).

Mielenkiintoisen sovellusryhmän muodostavat myös erilaiset virtuaaliympäristö-simulaattorit. Niillä koulutetaan käyttäjiä vastuullisiin tehtäviin ja vaarallisiin tai yllättäviin tilanteisiin, jotka vaativat suurta tarkkuutta ja huolellisuutta sekä ympäristön ennalta tuntemista. Simulaattoreita käytetään silloin kun todellisen ympäristön käyttö koulutukseen ei ole mahdollista tai sillä voidaan kouluttaa tehokkaammin ja edullisemmin. (Reitmaa ym. 1995, 109.) Esimerkiksi koulutus avaruusasemien tai ydinvoimalan huoltotehtävistä, satamien konttinosurikoulutus, reitti- ja matkustajakoneiden lentokoulutus ja sotilassimulaatiot (Reitmaa ym. 1995, 113).

2.33 Sotilaalliset sovellukset

Sotateollisuus on aina ollut vahvasti mukana uusien teknologioiden kehittämistyössä tai niiden soveltamisessa omiin tarkoituksiinsa, eikä virtuaalito-dellisuuskään ole poikkeus tästä. Sovelluskohteita ovat esim. räjähdysten simulointi ja niiden energia-aaltojen tutkiminen, ohjusten lentoradat ja niihin liittyvä havainnollistaminen. Virtuaalimallien sovelluskohteita ovat muun muassa erilaiset ympäristökoulutussimulaattorit, joissa käyttäjät voivat tutustua rakennusten, kaupunkien ja maisemien 3D-malleihin, usein satelliittikuvien avulla luotuihin. (Hintikka 1993, 85.)

3 VRML-KIELEN KEHITYS

3.1 Yleistä

Vuonna 1989 Rikk Carey ja Paul Strauss alkoivat rakentaa projektiympäristöä, joka pystyisi jakamaan 3D-sovelluksia sekä käyttämään tätä ympäristöä rakentaakseen 3D-rajapinnan (engl. desktop interface). VRML-kuvauskieli sai alkunsa Mark Pescen ideasta luoda kolmiulotteinen käyttöliittymä Internetiin. Hän esitteli ideansa ystävälleen Anthony Parisille, ja he alkoivat kehittää ideasta käyttökelpoista liittymää. Pesce ja Parisi saivat myöskin aikanaan kaikkien hyväksymän World Wide Webin 3D-standardin. (Bell, Pesce & Parisi 1995.)

Vuonna 1994 ilmestyi Silicon Graphicsin Open Inventor (Bell ym. 1995). Open Inventorissa oli tuki geometrinen 3D-maailmojen, eri valaistusten, pintamateriaalien ja kolmiulotteisen käyttöliittymän luomiselle (Kovaniemi 2002, 17). Samana vuonna Mark Pesce'n ja Anthony Parisin kehitystyö oli valmis; oli syntynyt Mosaic-selaimessa toimiva 3D-käyttöliittymä, joka sai nimekseen Labyrinth. He hankkivat lisenssin Silicon Graphicsilta, jotta voisivat alkaa kehittämään VRML-kieltä Open Inventorin pohjalta (Smith, Boyd, Scott 1996, 102).

Kieltä muokattiin ja yksinkertaistettiin sekä siihen lisättiin URL-pohjainen hyperlinkitys Pesce ja Parisi osallistuivat ensimmäiseen kansainväliseen WWW-konferenssiin kesällä 1994 Sveitsin Genevessä. Siellä he innokkaan ydinjoukon kanssa osallistuivat BOF-keskusteluun (BOF: Birds of a Feather -samanhenkiset) ja tuloksena oli nykyisen VRML-kielen perusta. Pesce ja Parisi saivat konferenssista hyvää palautetta, jonka innostamina he loivat WWW-VRML -postituslistan, jossa alasta kiinnostuneita pyydettiin ilmoittautumaan kielen kehittämiseksi. (Bell ym. 1995.)

3.2 VRML 1.0

Toukokuun lopussa 1995 kehityksen tuloksena syntyi VRML 1.0 -kieli. Se syntyi lopulta muutaman kuukauden työn jälkeen perustuen Silicon Graphicsin Open Inventor -tiedostomuotoon. Kieli tukee muotojen luontia, valaistusta ja tekstuureja. (Nadeau 1997.)

VRML 1.0:n kehittämisestä kiinnostuneiden tutkijoiden ja kehittäjien tapaaamisen seurauksena perustettiin VAG-ryhmä (VRML Architecture Group), jonka tehtävänä oli VRML-kielen kehittäminen. Vuoden 1995 lopulla ryhdyttiin kehittämään VRML 1.0:n laajennusta, VRML 1.1:tä. Siihen suunniteltiin lisättäväksi uusia kieliominaisuuksia, jotka olisivat tehneet selaimen toteuttamisen vaikeaksi ja mahdottomaksi. VRML 1.1:tä luovuttiin ja voimavarat suunnattiin VRML 2.0:n kehittämiseen. (Nadeau 1997.)

Vuoden 1996 alussa julkaistiin VRML 1.0c, koska VRML-selaimien kehittäjät törmäsivät lukuisiin ongelmiin VRML 1.0 kielen kanssa. Päivityksessä kieleen ei lisätty uusia ominaisuuksia vaan sitä korjattiin ja selkeytettiin.

3.3 VRML 2.0

Vuoden 1996 alussa VRML-arkkitehtuurin kehittäjäryhmä kokoontui lisäämään kieleen tuen animaatiolle ja vuorovaikutukselle. Ryhmä sai aikaan Moving Worlds -nimisen ehdotuksen, joka myöhemmin hyväksyttiin VRML 2.0:n pohjaksi ja antoi alkusysäyksen VRML 2.0:n kehitykselle. Vuoden 1996 lopulla ilmestyi VRML 2.0 -spesifikaatio, jossa oli uusi kielisyntaksi. Lisäksi uuteen versioon oli lisätty tuki äänenkäytölle, laajempi perusprimitiivien valikoima, animaatio, sumu, vuorovaikutus, taustat sekä yhteensopivuus Javan ja JavaScriptin kanssa. (Nadeau 1997.)

3.4 VRML97

Vuonna 1997 VRML 2.0:lle haettiin ISO-standardia. Kieltä kuitenkin jouduttiin selkeyttämään, joten se kirjoitettiin lähes kokonaan uudelleen sekä kieltä muutettiin hieman. Korjausten jälkeen kieli sai ISO-standardin, ISO/IEC 14772:1997. Standardoitua ISO VRML:lää kutsutaan VRML97:ksi. Verratuna versioon 2.0 uudessa VRML:ssä oli joitain korjauksia dokumenttiin ja muutamia funktioiden muutoksia. Lähes kaikki VRML 2.0:a tukevat selaimet pystyvät näyttämään VRML97-kielillä koodattuja VRML-tiedostoja. (Nadeau 1997.) Lisätietoa VRML-kielen eri versioiden eroista löytyy työ lopusta liitteestä 1 (s. - 49 -).

3.5 X3D

Extensible 3D (X3D) on ohjelmistostandardi, joka määrittelee interaktiivisen verkko- ja yleislähetyspohjaisen 3D-sisällön integrointi multimediaan. X3D:ta tehtiin ensimmäinen ISO-standardi vuonna 2004, ISO/IEC 19775. X3D-standardi on suunniteltu käytettäväksi useissa laitealustoissa ja ohjelmistoissa, kuten tekniikan ja tieteen visualisointi, multimedia esitykset, viihde ja koulutus julkaisut, Internet-sivut ja jaetut virtuaalimaailmat. X3D:n tavoite on muodostua maailmanlaajuiseksi tiedonsiirtomuodoksi 3D-grafiikalle ja multimedialle. (web3d 2006c.)

X3D on VRML:n seuraaja, jossa on parannettu ohjelmointirajapinta, enemmän datankoodausformaatteja, parempi yhteensopivuus ja komponentteihin jaettu arkkitehtuuri (web3d 2006c). X3D:n rakenne tekee myös säännöllisen päivittämisen helpoksi. Lisäksi on entistä helpompaa lisätä uusia ominaisuuksia, muuttamatta grafiikkaa tai kaupallisia sovelluksia (Web3d 2006b).

X3D SAI (engl. Scene Authoring Interface) mahdollistaa ohjelmointikielten sisäisen ja ulkoisen yhdenmukaisen toimivuuden. Tämä ei toimi VRML:ssä, koska Javalla ja ECMAScriptillä on erilaiset ohjelmointimallit. X3D SAI ratkaisee kaiken tämän määrittelemällä joukon abstrakteja palveluita, jotka ovat ohjelmointikielestä riippumattomia. Näiden palvelujen avulla ympäristöt toimivat yhdenmukaisesti riippumatta ohjelmointikielestä. Kielisidokset on toteutettu Javalle ja ECMAScript:lle, mikä tekee X3D:n luomisesta helpompaa. (Web3d 2006b.)

Perinteinen VRML-koodaus ja XML-kieli lisättiin X3D-standardiin vuonna 2005, ja samalla luotiin uusi ISO-standardi, ISO/IEC 19776 (web3d 2006c). Perinteinen VRML-koodaus voi esittää useimmat koodaamattomat VRML 2.0 -maailmat pienin muutoksin. VRML-tekniikka on lisätty X3D-kieleen siitä mitään poistamatta. X3D:tä varten on tehty paljon työtä, että se on yhteensopiva VRML:n kanssa ja silti on voitu ratkaista toisto-ohjelmien yhteensopivuusongelmat. XML-kieli on nopeasti tulossa tiedon sisällyttämisen vaihtoehdoksi yhtiöissä ja valtion tietokannoissa. (Web3d 2006b.)

X3D-standardi on määritelty siten, että ympäristöt voivat toimia vuorovaihteisesti selainten kanssa, jolloin yhteensopivuusongelmia ei esiinny. Web 3D -yhteenliittymä kehittänyt yhteensopivuus ohjelman, jossa testataan sovelluksen yhteensopivuus. Testatun ja todetun yhteensopivuuden merkiksi ohjelmistoissa saa käyttää X3D-logoa, joka kertoo ohjelman olevan yhteensopiva X3D:n kanssa. Tämän vuoksi selaimet ja toistosovellukset, jotka saavat käyttää kyseistä merkkiä toimivat luotettavasti ja tiedostot toimivat kaikissa sovelluksissa samalla tavoin. (Web3d 2006b.)

Kehitteillä on X3D:n binaarinen formaatti, joka sisältää mallin suojauksen ja sen hyvin pieneksi pakkaavan algoritmin X3D-ympäristöissä. Pakkaus-suhde on merkittävästi suurempi kuin VRML:n gzip-pakkausalgoritmi. Ympäristön jäsentämisen ja lataamisen nopeutuminen 300–500 % pitäisi olla mahdollista. Selaimien useiden koodausten tukeminen helpottuu, kun ainoa todellinen eroavaisuus selainten välillä oli erilainen jäsenin. Kuitenkin koodaukset voidaan sisällyttää ympäristöön edellyttäen, että selain tukee käytettyjä koodauksia. Nykyisen X3D-selaimen kehittäjät suunnittelevat kaikkien koodausten tukemista. (Web3d 2006b.)

3.6 VRML:n leviämiseen vaikuttaneita asioita

Yksi suurimmista ongelmista VRML:ssä oli luoda VRML-ympäristöjä, jotka toimivat kaikissa yhteensopivissa selaimissa ja toisto-ohjelmissa. Lisäksi VRML-standardin puutteellinen määrittely aiheutti yhteensopivuusongelmia selaimissa ja toisto-ohjelmissa. Ohjelmistokehittäjillä on myös osuutensa VRML:n ongelmiin: koska osa heistä oli tyytymättömiä VRML 2.0 -spesifikaation julkistamisen hitauteen, niin he kehittivät omia ominaisuuksia selaimiinsa ja liitännäisiinsä. Nämä kehitetyt ominaisuudet lisäsivät yhteensopivuusongelmia entisestään, koska VRML-kieli ei tukenut niitä. Lisäksi oh-

jelmistot olivat levityksessä ennen kuin VRML97 julkaistiin, mikä heijasti yhteensopivuusongelmat käyttäjille asti. VRML97-spesifikaation julkaisemisen jälkeenkin kului aikaa, ennen kuin ensimmäiset VRML97:ää tukevat selaimet julkaistiin.

VRML-selaimen tai -liitännäisen lataaminen ja asentaminen tietokoneeseen oli monelle käyttäjälle kynnyks, koska osa käyttäjistä ei osannut tai viitsinyt hankkia tarvittavia sovelluksia. Ympäristössä liikkuminen selaimen avulla oli ja on edelleen hankalaa. Selaimien käyttöliittymät poikkesivat toisistaan paljon ja toimivat eri tavoin. Sujuva liikkuminen virtuaaliympäristössä edellyttäisi, että olisi mahdollista liikkua kaikkiin kuuteen liikesuuntaan – ylös, alas, eteen ja taakse sekä molemmille sivulle. Liikesuuntien määrä rajoittuu neljään, kun navigointiin käytetään näppäimistöä tai hiirtä.

1990-luvulla käytössä olleet tietokoneet ja tiedonsiirtonopeudet olivat melko vaatimattomia. Tietokoneiden tehokkuudelle käyttäjät eivät mahtaneet mitään, mutta tekniikka olisi mahdollistanut nopeammat tiedonsiirron. Kotikäyttäjillä oli yleensä käytössä lankamodeemi, jonka tiedonsiirtonopeus oli usein alle 33,6 kb/s. Hitaalla modeemiyhteydellä ja tehottomalla tietokoneella pystyi lataamaan hyvin pienen ympäristön ja liikkumaan siinä hitaasti ja vaivalloisesti. Lyhyet siirtymät ympäristössä onnistuivat kohtalaisen nopeasti, mutta pitkiä matkoja siirryttäessä aikaa kului todella paljon. Esimerkiksi kun tutkitaan kaupunkiympäristöä ja siirrytään kadun loppupäähän, niin siirtyminen saattaa kestää useita kymmeniä sekunteja. Ongelma on lähes samanlainen riippumatta siitä käytetäänkö hiirtä tai näppäimistöä navigointiin. Ongelmaa pystytään pienentämään lisäämällä ympäristöön kameeroita näkymiksi. Kameroiden lisääminen helpottaa ympäristön tutkimista, mutta se johtaa hyvin helposti kamerasta kameraan loikkimisen ja muu ympäröivä sisältö saattaa jäädä tutkimatta.

VRML-kielestä puuttui lähes kokonaan todellisen maailman visuaalinen realismi. Virtuaalimaailmojen luomisen nopeus ilman ohjelmointia oli etu käyttäjälle, mutta samalla sivuutettiin realismin jäljittely teksturoinnissa. Kielessä valot tukevat fysiikan lakien mukaista valon etenemistä, mutta niillä ei pystytä luomaan varjoja ympäristöön. Kolmiulotteisen tilan ja objektien hahmottaminen vaati aina varjot, että ihminen osaa suhteuttaa objektien koon, sijainnin ja mittasuhteet ympäristössä. Lisäksi kieli ei tukenut pinnan kiiltoja ja heijastuksia, joita esiintyy esimerkiksi kiillotetulla metallipinnalla.

Viihdeteollisuuden kiinnostus heräsi VRML:ää kohtaa 1990-luvun puolivälissä, kun CAA (Creative Artists Agency) ja ICM (International Creative Artists) halusivat toteuttaa yhteistyökumppaniensa kanssa verkossa toimivan kolmiulotteisen tietokonepohjaisen pelin, jonka projektinimi oli The Spot. Saman projektin yhteydessä oli myös vaikuttajia Hollywoodista ja tieto VRML:n mahdollisuuksista levisi elokuvateollisuuteen. Ensimmäisenä elokuvastudiona Paramount käytti VRML:ää Internet-sivujen sisällön luomiseen. Vuonna 1996 julkaistun Star Trek: First Contact -elokuvan Internet-sivuilla pääsi pelaamaan interaktiivista VRML-pohjaista peliä, joka esitettiin

pelihahmon näkökulmasta nähtynä (engl. first-person). Pelissä pystyi kulkemaan aluksen käytävillä ja väijymään Borg-vihollisia. Sivustolla oli liki 6 miljoonaa kävijää päivittäin.

Peli- ja elokuvateollisuus eivät nähneet suuria mahdollisuuksia VRML:ssä, koska niillä oli jo käytössä toimivat sovellukset ja tekniikat. Peliteollisuus ponnisti ylös id Softwaren kehittämällä Wolfenstein 3D:llä, joka julkaistiin vuonna 1992. Tämä FPS-peli (engl. first person shooter) oli todella suosittu, ja nykyäänkin valtaosa julkaistavista peleistä on FPS-pelejä. Elokuvateollisuus mullisti elokuvahistoriaa julkaisemalla lukuisilla erikoistehosteilla siivitetyn The Lawnmower Man -elokuvan vuonna 1992. Elokuvassa pääosaa näyttävä henkilö siirsi itsensä ikuisiksi ajoiksi kyber-avaruuteen ja liikkui hyvin efektiivisessä virtuaalitallassa. Elokuvassa käytettyjä efektejä ei olisi pystynyt toteuttamaan VRML-kielillä. Lisäksi osalle elokuvan nähneistä ihmisistä jäi varmasti lähtemätön mielikuva, miltä virtuaalitalat näyttävät ja miten niissä liikutaan.

VRML-kieli syntyi noin kymmenen vuotta liian aikaisin, kun asiaa tarkastellaan käyttäjän näkökulmasta. Tekniikka on kehittynyt valtavasti, ja laitteiden hinnat ovat murto-osia 1990-luvun hinnoista. Käytettävän laitteiston hinta vaikuttaa merkittävästi siihen, miten tekniikka jalkautuu käyttäjille. Nyt laitteet ovat edullisia, ja valtaosalla käyttäjistä on käytössä tehokkaat tietokoneet ja nopeat Internet-yhteydet, joilla ei ole suuria vaikeuksia renderöidä suuriakin VRML-ympäristöjä. Viihteen ja kaupallisuuden näkökulmasta tarkasteltuna VRML-kieli syntyi vähintään kymmenen vuotta liian myöhään, koska 1990-luvun alussa Sun Microsystems, Inc. kehitti Java-kielen ja Silicon Graphics kehitti OpenGL-ohjelmointirajapinnan. Varsinaisen ohjelmointituen pois jättäminen kielen ensimmäisestä versiosta oli selkeä merkki siitä, että kieli ei ollut tarkoitettu myös teolliseen käyttöön.

Virtuaalinen mallinnus on yleistynyt merkittävästi, ja samalla on lisääntynyt suurien tutkimuskeskusten ja korkeakoulujen käyttämien virtuaalitutkimusympäristöjen määrä. Virtuaalisen tilan simulaatioiden ja mallien esittämiseen VRML-kieltä ei enää käytetä, koska muilla ohjelmoitavilla vaihtoehdoilla saavutetaan todenmukaisempi realismi ja yhteensopivuus muihin ohjelmistoihin. VRML-kieltä voidaan vielä käyttää tiedonsiirtomuotona 1990-luvulla julkaistujen ohjelmien välillä. Esimerkiksi tehdasmallin simuloinnin toteutus käyttämällä 3ds Max -ohjelmaa mallin luontiin ja VRML-tiedostoksi tallentamiseen ja Delmia QUEST -simulointiohjelmaa simulaation toteuttamiseen.

4 VRML-KIELI

4.1 VRML-tiedosto

VRML-tiedosto on tekstikuvaus virtuaaliympäristöstä. Tiedosto sisältää tekstiä, joka voidaan luoda tekstieditorilla tai 3D-mallinnusohjelmalla ja tallentaa se VRML-tiedostoksi. (Ames, Nadeau & Moreland 1997, 11.) Tiedostopääte on *.wrl, joka ilmaisee että kyseessä on VRML-ympäristö. Tiedostossa tarvitaan aina tunniste, ja lisäksi tiedostossa voi olla kommentteja, kenttiä ja niiden arvoja, määriteltyjä solmuja ja käytettyjä solmuja (Ames ym. 1997, 12).

Tiedosto voi sisältää seuraavat neljä pääkomponenttia:

VRML-tunniste (engl. header)
Prototyyppi (engl. prototype)
Muodot, interpoloijat, tunnistimet ja ohjelmointikieli
Reitit (engl. routes)
(Ames ym. 1997, 11.)

4.2 Syntaksi

Tunniste

Tunniste kertoo Internet-selaimelle, että kyseessä on VRML-tiedosto jonka kielen versio on 2.0 ja että tiedosto käyttää UTF-8 merkistöä (Ames ym. 1997, 13).

Esimerkki 1, VRML-tunniste.

```
#VRML V2.0 utf8  
(Ames ym. 1997, 13).
```

Kommentointi

VRML-tiedoston kommentointi ei vaikuta ympäristöön millään tavoin. Kommentointi helpottaa tiedoston lukemista kirjoitusta tai muokkausta myöhemmin. Kommentointi aloitetaan # -merkillä. (Ames ym. 1997, 13.) Kommentointi päättyy rivin lopussa ilman erillistä kommentointia päättävää merkkiä (Ames ym. 1997, 14).

Esimerkki 2, kommentointi.

```
# Tämä on kommentti  
(Ames ym. 1997, 14).
```

Solmu

Solmuilla (engl. nodes) määritellään muodot ja niiden ominaisuudet. Yksittäisillä solmuilla määritellään muoto, väri, valo, animaatio ja muut ominaisuudet. Solmu sisältää yleensä tiedon solmun tyypistä, { }-sulkeet sekä mahdollisesti kentän ja sen arvon. (Ames ym. 1997, 14.)

Esimerkki 3, sylinteri-solmu.

```
Cylinder {}  
(Ames ym. 1997, 14.)
```

Kenttä ja parametrit

Kenttä (engl. field) määrittelee solmun attribuutit. Kenttiä ovat esimerkiksi sylinterin korkeus, pohjan säde ja väri. Edellä olevassa esimerkissä (katso esimerkki 3 s. 31) ei ole määritelty kenttiä eikä niiden parametreja, joten solmu piirretään perusparametreilla. Alla olevassa esimerkissä pakotetaan sylinterin korkeudeksi 3.0 yksikköä ja pohjan säteeksi 3.0 yksikköä (Ames ym. 1997, 14.)

Esimerkki 4, solmun kentät (height ja radius) ja parametri (3,0 ja 3,0).

```
Cylinder {  
  height 3.0  
  radius 3.0  
} (Ames ym. 1997, 14).
```

Solmun nimeäminen ja käyttäminen

VRML-kielessä voidaan määritellä solmulle nimi. Nimi voi olla kirjaimia tai numeroiden ja kirjaimien yhdistelmiä. Nimettyä solmua voi käyttää toistuvasti tiedostossa. Etuna solmun määrittelyssä on se, että vain ensimmäisellä käyttökerralla tarvitsee kirjoittaa koko koodi. Tämän jälkeen kutsutaan nimettyä solmua koodissa ja nimetyn solmun määritelmä toistuu ympäristössä. (Ames ym. 1997, 16.)

Nimettyä solmua kutsutaan alkuperäiseksi solmuksi ja solmua uudelleen käytettäessä syntyviä solmuja kutsutaan instansseiksi. Vain alkuperäisen solmun kenttien parametreja voidaan muuttaa, mutta samat muutokset vai-

kuttavat myös instansseihin. (Ames ym. 1997, 16.) DEF-komennolla nime-
tään ruskea niminen solmu. Ruskea-solmu sisältää sylinteri-solmun. (Ames
ym. 1997, 17.)

Esimerkki 5, solmun nimeäminen.

DEF ruskea Cylinder { }
(Ames ym. 1997, 17.)

USE-komennolla kutsutaan nimetty ruskea-solmua. Aina kun koodissa kut-
sutaan ruskea-solmua, niin siihen kohtaan kopioidaan näennäisesti sylinteri
DEF-komennon alta. (Ames ym. 1997, 17.)

Esimerkki 6, nimetyn solmun käyttäminen.

USE ruskea
(Ames ym. 1997, 17.)

Muodot

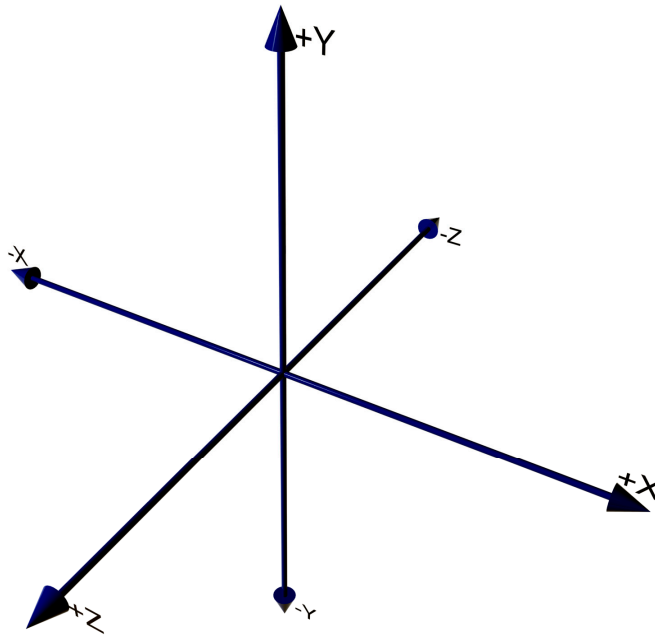
VRML-kielessä 3D-rakenteen määritelmä perustuu materiaaliin ulkonäköön,
väriin ja tekstuuriin. Kielessä on määritelty useita perusgeometrisiä muotoja
kuten kuutio ja pallo sekä monimutkaisia geometrioita, kuten muodon pur-
sotus. (Ames ym. 1997, 17.)

Muotojen ryhmittely

Muotoja voi ryhmittää (engl. grouping) keskenään, jolloin niistä monimutkai-
sempia muotoja. Ryhmittämällä kartio ja pallo voidaan rakentaa esimerkiksi
jäätelötuutti. Solmua, joka ryhmittelee muodot, kutsutaan vanhemmaksi
(engl. parent). Muotoja, jotka muodostavat ryhmän, kutsutaan lapsiksi (engl.
children). Ryhmän sisällä voi olla ryhmiä, jotka ovat päätasoryhmän lapsia.
(Ames ym. 1997, 18.)

Objektiavaruus

Solmuilla ja kenttien parametreilla määritellään objekteja kolmiulotteiseen ti-
laan. Parametreilla määritellään objektin tarkka koko ja paikka kolmiulot-
teisessa koordinaatistossa. Kielessä käytettävä koordinaatistojärjestelmä
on oikeakätinen koordinaatisto, jossa positiivinen z-akseli on kohti käyttä-
jää. (Ames ym. 1997, 18.)



Kuva 1 Oikeakätinen koordinaatisto.

Tapahtumat

Tapahtumilla (engl. events) voidaan virtuaaliympäristössä syyttää ja sammuttaa valo, käynnistää animaatio tai aktivoida ääni (Ames ym. 1997, 21.) Tapahtuman luomiseksi tarvitaan kaksi solmua, jotka ovat kytketty toisiinsa. Solmu A lähettää signaalin määritellyä reittiä (engl. route) pitkin solmu B:lle, joka vastaan ottaa signaalin. Vastaan otettuaan signaalin solmu B käynnistää määritellyn tapahtuman. Kytkemällä useita solmuja yhteen voidaan luoda monimutkaisen dynaaminen tapahtumaverkko. (Ames ym. 1997, 22.)

4.3 Perusgeometria

4.31 Perussolmut

VRML-kielessä on useita perusgeometrisiä solmuja, kuten laatikko, kartio, sylinteri ja pallo. Näitä solmuja voidaan käyttää muotosolmun kanssa. Objektin ulkonäkö määritellään ulkonäkö- ja materiaalisolmuilla. Jokaisella geometriasolmulla on yksi tai useampi kenttä, jolla voidaan muuttaa objektin parametreja. Objektit rakennetaan siten, että sen keskipiste on aina koordinaatiston origossa. (Ames ym. 1997, 25.)

Shape-solmu

Kaikki muodot VRML-kielessä on rakennettu niin, että ne käyttävät Shape-solmua (Ames ym. 1997, 26). Shape-solmun Appearance-kentässä määritellään objektin ulkonäkö, kuten väri ja pinnan materiaali. Geometry-kentässä määritellään objektin muoto. Usein kentän arvoina ovat perusgeometrian solmut, kuten laatikko- tai pallosolmu. Molempien kenttien oletusarvo on tyhjä (engl. null). (Ames ym. 1997, 26.)

Shape-solmun syntaksi:

```
Shape {  
  appearance NULL  
  geometry NULL  
} (Ames ym. 1997, 26).
```

Appearance-solmu

Appearance-solmu määrittää objektin ulkonäön ja sitä voidaan käyttää Shape-solmun Appearance-kentän arvona (Ames ym. 1997, 27). Material-kentän arvona on yleensä Material-solmu. Kun kentälle ei ole määriteltä arvoa, niin objektin materiaalina on hehkuva valkoinen. (Ames ym. 1997, 27.) Texture-kentällä määritellään objektin pintaan liitettävä kuva, ja arvona on yleensä ImageTexture, PixelTexture tai MovieTexture-solmu (Ames ym. 1997, 307). TextureTransform-kentällä luodaan uusi tekstuuri koordinaatisto, joka on suhteessa alkuperäiseen tekstuuri koordinaatistoon (Ames ym. 1997, 354).

Appearance-solmu syntaksi:

```
Appearance {  
  material NULL  
  texture NULL  
  textureTransform NULL  
} (Ames ym. 1997, 27).
```

Material-solmu

Material-solmu määrittelee objektin materiaalin parametrit. Solmu luo oletusarvoilla varjostettuja valkoisia muotoja, kun { } -sulkeiden väliin ei ole määriteltä kenttiä. (Ames ym. 1997, 27.)

Material-solmun syntaksi:

```
Material {  
  ambientIntensity 0.2  
  diffuseColor 0.8 0.8 0.8  
  emissiveColor 0.0 0.0 0.0  
  shininess 0.2  
  specularColor 0.0 0.0 0.0  
  transparency 0.0  
} (Ames ym. 1997, 27).
```

Group-solmu

VRML-kielessä solmuja voidaan ryhmitellä käyttämällä Group-solmua (Ames ym. 1997, 30). Children-kentässä määritellään lista lapsi-solmuista, jotka lisätään ryhmään. Yleisimmät lapsi-solmut ovat Shape-solmu ja toiset Group-solmut. (Ames ym. 1997, 30.) BboxCenter-kentässä määritellään rajainlaatikon koordinaatiston keskipiste, joka on sama kuin ryhmän kaikkien objektien keskipiste. BboxSize-kentässä määritellään rajainlaatikon koko niin, että se kattaa kaikki ryhmän objektit. VRML-selain yleensä määrittelee automaattisesti rajainlaatikon koon ja keskipisteen. (Ames ym. 1997, 189.)

Group-solmun syntaksi:

```
Group {  
  children [ ]  
  bboxCenter 0.0 0.0 0.0  
  bboxSize -1.0 -1.0 -1.0  
} (Ames ym. 1997, 30)
```

4.32 Box-solmu

Box-solmu luo laatikon muotoisen objektin (katso Laatikko s. - 18 -). Box-solmua voidaan käyttää Shape-solmun Geometry-kentän arvona. (Ames ym. 1997, 27.) Size-kentällä määritellään laatikon koko kolmiulotteisessa koordinaatistossa (x, y, z). Ensimmäinen arvo määrittelee leveyden, toinen arvo korkeuden ja kolmas arvo syvyyden. Size-kentän oletusarvo on 2.0 yksikköä, jokaiselle koordinaattipisteelle. (Ames ym. 1997, 28.)

Box-solmun syntaksi:

```
Box {  
  size 2.0 2.0 2.0  
} (Ames ym. 1997, 28)
```

4.33 Cone-solmu

Cone-solmu luo kartion muotoisen objektin (katso Kartio s. - 19 -). Cone-solmua voidaan käyttää Shape-solmun geometry-kentän arvona. (Ames ym. 1997, 28.) BottomRadius-kentällä määritellään kartion pohjaympyrän säde ja height-kentällä määritellään kartion korkeus. Side- ja bottom-kentillä määritellään piirretäänkö näkyviin kartion vaippa ja pohja. Arvoina kentillä ovat TOSI (TRUE) tai EPÄTOSI (FALSE). (Ames ym. 1997, 28).

Cone-solmun syntaksi:

```
Cone {  
  bottomRadius 1.0  
  height 2.0  
  side TRUE  
  bottom TRUE  
} (Ames ym. 1997, 28).
```

4.34 Cylinder-solmu

Cylinder-solmu luo sylinterin muotoisen objektin (katso Sylinteri s. - 19 -). Cylinder-solmua voidaan käyttää Shape-solmun Geometry-kentän arvona. (Ames ym. 1997, 29.) Radius-kentällä määritellään sylinterin pohjaympyrän säde ja height-kentällä määritellään kartion korkeus. Side-, top- ja bottom-kentillä määritellään, piirretäänkö näkyviin sylinterin vaippa, kansi ja pohja. Arvoina kentillä ovat KYLLÄ (TRUE) tai EI (FALSE). (Ames ym. 1997, 28.)

Cylinder-solmun syntaksi:

```
Cylinder {  
  radius 1.0  
  height 2.0  
  side TRUE  
  top TRUE  
  bottom TRUE  
} (Ames ym. 1997, 29).
```

4.35 Sphere-solmu

Sphere-solmu luo pallon muotoisen objektin (katso Pallo s. - 20 -). Sphere-solmua voidaan käyttää Shape-solmun geometry-kentän arvona. Radius-kentällä määritellään pallon säde. (Ames ym. 1997, 29.)

Sphere-solmun syntaksi:

```
Sphere {  
  radius 1.0  
} (Ames ym. 1997, 30)
```

4.36 Kolmiulotteinen teksti

Kolmiulotteinen teksti luodaan Text-solmulla (katso Kolmiulotteinen teksti s. - 20 -), jonka kentillä voi muokata tekstin ulkoasua. Text-solmua voidaan käyttää Shape-solmun Geometry-kentän arvona. (Ames ym. 1997, 39.) String-kentässä määritellään piirrettävä teksti, ja teksti pitää kirjoittaa lainausmerkkien sisään. Length-kentässä määritellään tekstin pituus ja maxLength-kentässä tekstin maksimi pituus. (Ames ym. 1997, 42.) FontStyle-kentässä käytetään yleensä FontStyle-solmua, jolla määritellään käytettävä merkistö, fontti, fontin koko, välistys, tekstin orientaatio ja kulkusuunta (Ames ym. 1997, 40).

Text-solmun syntaksi:

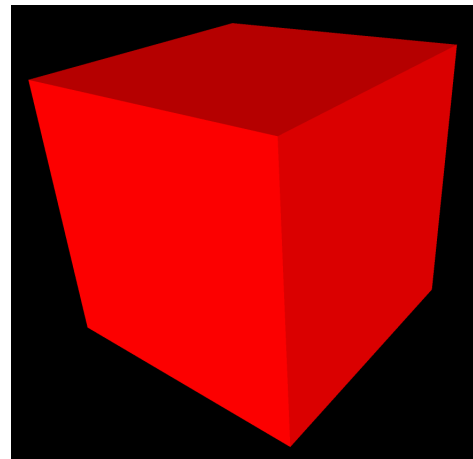
```
Text {  
  string []  
  length []  
  maxLength 0.0  
  fontStyle NULL  
} (Ames ym. 1997, 42)
```

4.37 Perusgeometrian esimerkit

Laatikko

Esimerkki VRML-kielen Box-solmusta.

```
#VRML V2.0 utf8  
Shape {  
  appearance Appearance {  
    material Material {  
      diffuseColor 1.0 0.0 0.0  
      ambientIntensity 0.2  
      shininess 0.2  
    }  
  }  
  geometry Box {}  
}
```

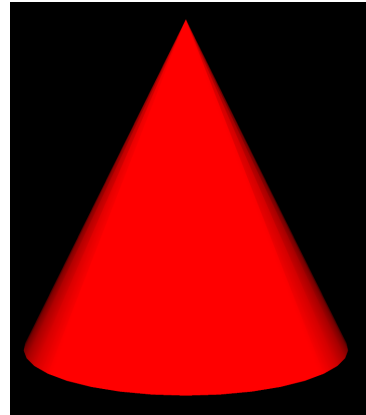


Kuva 2 VRML-kielillä luotu laatikko-objekti.

Kartio

Esimerkki VRML-kielen Cone-solmusta.

```
#VRML V2.0 utf8  
Shape {  
  appearance Appearance {  
    material Material {  
      diffuseColor 1.0 0.0 0.0  
      ambientIntensity 0.2  
      shininess 0.2  
    }  
  }  
  geometry Cone {}  
}
```

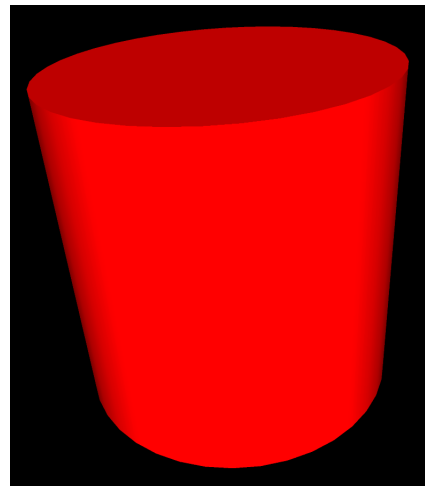


Kuva 3 VRML-kielillä luotu kartio-objekti.

Sylinteri

Esimerkki VRML-kielen Cylinder-solmusta.

```
#VRML V2.0 utf8  
Shape {  
  appearance Appearance {  
    material Material {  
      diffuseColor 1.0 0.0 0.0  
      ambientIntensity 0.2  
      shininess 0.2  
    }  
  }  
  geometry Cylinder {}  
}
```

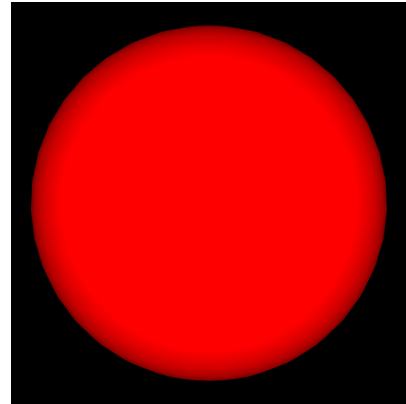


Kuva 4 VRML-kielillä luotu sylinteriobjekti.

Pallo

Esimerkki VRML-kielen Sphere-solmusta.

```
#VRML V2.0 utf8  
Shape {  
  appearance Appearance {  
    material Material {  
      diffuseColor 1.0 0.0 0.0  
      ambientIntensity 0.2  
      shininess 0.2  
    }  
  }  
  geometry Sphere {}  
}
```



Kuva 5 VRML-kielillä luotu pallo-objekti.

Kolmiulotteinen teksti

Esimerkki VRML-kielen Text-solmusta.

```
#VRML V2.0 utf8  
Shape {  
  appearance Appearance {  
    material Material {  
      diffuseColor 1.0 0.0 0.0  
      ambientIntensity 0.2  
      shininess 0.2  
    }  
  }  
  geometry Text {  
    string "3D-teksti"  
  }  
}
```



Kuva 6 VRML-kielillä luotu tekstiobjekti.

4.4 Objektin siirto, kierto ja skaalaus

VRML-kielessä voi luoda rajattoman määrän koordinaatistoja. Koordinaatistot asemoidaan tai muutetaan aina jonkun toisen koordinaatiston suhteen. Uusi koordinaatisto on lapsi-koordinaatisto (engl. child coordinate system), joka on sisäkkäin vanhempi-koordinaatiston (engl. parent coordinate system) kanssa. (Ames ym. 1997, 63.) Vanhempi-koordinaatisto voi vuorostaan olla jonkun toisen koordinaatiston lapsi. Lapsi–vanhempi koordinaatis-

to suhteesta syntyy koordinaatisto puu. Puun ensimmäinen koordinaatisto on juuri-koordinaatisto (engl. root coordinate system), ja se löytyy jokaisesta VRML-tiedostossa. (Ames ym. 1997, 64.)

Transform-solmun syntaksi:

```
Transform {  
  children [ ]  
  translation 0.0 0.0 0.0  
  rotation 0.0 0.0 0.0 0.0  
  scale 1.0 1.0 1.0  
  scaleOrientation 0.0 0.0 0.0 0.0  
  center 0.0 0.0 0.0  
  bboxCenter 0.0 0.0 0.0  
  bboxSize -1.0 -1.0 -1.0  
} (Ames ym. 1997, 67).
```

Children-kentän arvoina on yleensä Shape-, Group- tai Transform-solmu. Translation-kentässä määritellään luotavan koordinaatiston origon etäisyys x-, y- ja z-akselien arvoina vanhemman koordinaatiston origosta. (Ames ym. 1997, 67.) Rotation-kentällä määritellään pyörähdys x-, y- ja z-akselien ympäri ja sen kiertokulma radiaaneina (Ames ym. 1997, 82). Scale-kentällä määritellään koordinaatiston skaalauskerroimet jokaisen akselin suhteen. Skaalaus vaikuttaa kaikkiin solmuihin, jotka ovat children-kentässä. ScaleOrientation-kentässä määritellään skaalauksen koordinaatisto ja kiertokulma radiaaneina. Skaalauksen jälkeen koordinaatisto palautetaan ennalleen arvoista, jotka on annettu scaleOrientation-kentässä. (Ames ym. 1997, 97.) Center-kentällä määritellään kierron keskipiste, mikäli se ei ole origo. (Ames ym. 1997, 82.)

4.5 Teksturointi

4.51 RGB-värit

Objektien värit määritellään tarkasti punaisen (R), vihreän (G) ja sinisen (B) valon arvoina, jotka sekoitetaan keskenään virtuaalisesti määrittelyn mukaan. Tätä värinmittausta kutsutaan RGB-väreiksi (engl. RGB colors), koska siinä määritellään, miten punaisen, vihreän ja sinisen valon määrä yhdistetään keskenään. RGB-väri koostuu kolmesta liukulukuarvosta, joiden arvot ovat väliltä 0.0 - 1.0. Arvolla nolla (0.0) valo ei ole päällä ja vastaavasti arvolla yksi (1.0) valo on päällä. Lisäksi arvoja voi olla myös väliltä 0.0 - 1.0, jolloin valo on osittain päällä. RGB-värien eri yhdistelmillä kyetään luomaan perusvärit sekä suuri määrä muita värejä (katso Taulukko 1 s. - 22 -). (Ames ym. 1997, 162). Esimerkiksi keltainen väri määritellään siten, että punaiselle ja vihreälle valolle annetaan arvoiksi 1.0 ja siniselle arvoksi 0.0.

Taulukko 1 RGB-värejä.

R	G	B	Väri
0	0	0	musta
0,5	0,5	0,5	harmaa
1	1	1	valkoinen
1	0	0	punainen
0	1	0	vihreä
0	0	1	sininen
1	1	0	keltainen

(Ames ym. 1997, 163.)

4.52 Kuvatiedostot

Etuna 2D-kuvan käyttämisenä tekstuurina (engl. picture mapping) on, että ympäristöön saadaan lisättyä yksityiskohtia, ilman että niitä tarvitsee luoda malleihin (Ames ym. 1997, 297.) Lähes kaikkia VRML-objekteja voidaan teksturoida kuvilla pois lukien pisteet ja viivat. Objektien pinnalle laitettavien kuvien koko on yleensä melko pieni. Kuva voi olla digitaalikameralla otettu kuva tai grafiikkaohjelmalla tehty 2D-kuva sekä video-tiedosto. (Ames ym. 1997, 298.) Kuvatiedostot voivat olla värillisiä tai harmaasävykuvia (Ames ym. 1997, 301).

Kuvien käyttäminen tekstuurina kasvattaa tiedoston kokoa huomattavasti. Objektien teksturointiin kannattaa käyttää kuvia, joiden koko on 128 pikseliä leveä ja 128 pikseliä korkea tai vielä pienempiä kuvia, jos kuvan informaatio ei siitä kärsi. Objektin kuvan sijainti määritellään ImageTexture-solmun URL-kenttään ja kuvatiedosto voi sijaita missä tahansa, kuten Internetissä tai tietokoneen kovalevyllä.

Teksturointiin käytetään yleensä jpg-, gif- tai png-kuvatiedostoja ja videoina mpeg-videotiedostoja. (Ames ym. 1997, 298.) Toimivin tiedostomuoto objektin teksturointiin on jpg-tiedostojen käyttäminen (Ames ym. 1997, 300). PNG-kuvatiedosto voi sisältää läpinäkyvyyskanavan, jossa määritellään, mistä kohtaa kuva on läpinäkyvä ja miltä kohtaa se on läpinäkymätön. Kun objekti teksturoidaan kuvalla, jossa on läpinäkyvyyskanava, niin sen pintaa syntyy samanlainen reikä tms. muoto kuin on kuvan läpinäkyvässä osassa. (Ames ym. 1997, 302). Videotiedosto toistuu objektin pinnalla samalla tavoin kuin se näkyisi videonkatseluohjelmassa (Ames ym. 1997, 298).

4.6 Varjostus

Objektit varjostetaan, jotta niistä välittyisi katsojalle kolmiulotteisuuden tuntu. Jos objekteja ei varjostettaisi, niin ne näyttäisivät piirroksilta kaksiulotteisessa tasossa. (Ames ym. 1997, 163.) VRML-selain lisää automaattisesti valon lähteen VRML-ympäristöön, jos sitä ei ole määritelty ympäristöön. Selain valaisee ja varjostaa objektit automaattisesti katselijan katselupisteestä. (Ames ym. 1997, 164.) Objektin pinnan kirkkaus riippuu siitä, kuinka paljon sen pintaan osuu valoa (Ames ym. 1997, 373).

Selain määrittelee objektin pinnoille pinnan normaalit ja laskee kulman pinnan normaalin ja kuvitteellisen pisteen välille, joka osoittaa valoa kohti. Pinnan normaali on vektori, joka osoittaa kohtisuoraan pinnasta poispäin. Kulman arvon perusteella selain saa tiedon, onko pinta valoa kohti vai ei. Pienellä kulman arvolla pinta on lähes kohtisuoraan valon kanssa ja suurella kulman arvolla (kulma $\geq 90^\circ$) pinta on kääntynyt poispäin valosta. Jos pinnan normaali osoittaa valoa kohti, niin se varjostuu kirkkaammin ja jos se osoittaa poispäin valosta, niin se varjostuu tummemmin. (Ames ym. 1997, 373.)

Kielessä pystyy myös estämään selaimen automaattisen pinnan normaalien luomisen ja määrittelemään omat pinnan normaalit. Itse luotavien pinnan normaalien etuna on se, että niillä voidaan määritellä, kuinka objektin pinta varjostuu. Lisäksi hyvin kulmikkaita objekteja voidaan varjostaa siten, että terävät kulmat näennäisesti pyöristyy ja objektin pinta näyttää sileältä, kuin pintaverkossa olisi enemmän polygoneja. (Ames ym. 1997, 373.)

4.7 Valaistus

VRML-ympäristön valaisemisella on sama tarkoitus kuin valolla todellisessa maailmassa. Ympäristön valaistus on tärkeä osa sen näyttävyyttä ja vaikuttaa ympäristön laatuun sekä tunnelmaan. Valoilla voidaan haluttuja seikoja joko korostaa tai häivyttää. Hyvin suunnitellulla ja toteutetulla valaistuksella voidaan kompensoida ympäristön ja sen mallien mahdollisia puutteita. (Ames ym. 1997, 413).

VRML-kielessä on tuettu kolmea valotyyppiä, jotka ovat pistevalo (engl. point light), suunnattu valo (engl. directional light) ja kohdevalo (engl. spot light). Lisäksi kielessä on otsavallo (engl. head light), jonka VRML-selain lisää automaattisesti ympäristöön. Valoilla ei ole fyysistä muotoa, jonka voisi nähdä ympäristössä. Valoista näkee vain niiden määrittelyn tuloksen, eli miten ja mitä ympäristössä valaistetaan. (Ames ym. 1997, 407.) Valot eivät luo varjoja objekteille ympäristössä, koska niiden laskeminen tietokoneella interaktiivisessa ympäristössä on todella raskasta (Ames ym. 1997, 413).

Valot voidaan sammuttaa tai laittaa päälle, ja niiden intensiteettiä voidaan muuttaa kirkkaammaksi tai tummemmaksi sekä valon vaikutusta hajavalotasoon ympäristössä ja valon väri voidaan muuttaa RGB-väreinä. (Ames ym. 1997, 408.) Valolle voidaan määrittää vaimenemisetäisyys valon lähteestä, jolloin valon määrä alkaa vähitellen vaimeta. Valon vaimenemista tukevat piste- ja kohdevalo. Ympäristössä voi olla useita valoja, jotka ovat erityyppisiä ja jokaisen valon väri voi olla erilainen. Ympäristöön sijoitettavien valojen määrä rajoittaa ainoastaan käytettävän tietokoneen 3D-grafiikka kortin ominaisuudet. (Ames ym. 1997, 412.)

Pistevalo valaisee ympäristön yhdestä pisteestä, josta valon säteet lähtevät kaikkiin suuntiin. Suunnattu valo valaisee ympäristön auringon tavoin siten, että kaikki säteet ovat samansuuntaisia kuin valo olisi sijoitettu todella kauas. (Ames ym. 1997, 408.) Kohdevalo valaisee ympäristön tietystä pisteestä ja valon säteet etenevät kartion muotoisena valokeilana. Valokeilan sisällä olevat objektit valaistuvat ja ulkopuolella olevat jäävät valaistumatta ympäristössä. (Ames ym. 1997, 409.)

4.8 Optimointi

Optimoinnilla pyritään muokkaamaan VRML-tiedoston toimivuutta ja käytettävyyttä siten, että ympäristö olisi mahdollisimman monen käyttäjän saatavilla. VRML-ympäristöä rakentaessa pitää välttää *.wrl-tiedoston suurta kokoa ja monimutkaisuutta. Suurien VRML-tiedostojen latausajat ovat pitkiä ja odottelu karsii pois ne käyttäjät, joilla on hitaat Internet-yhteydet. Monimutkaiset ympäristöt yleensä luovat lisävaatimuksia tietokoneille, joten tehoittomien tietokoneiden käyttäjät voivat tuntea tällaisten ympäristöjen käyttämisen hitaaksi ja hankalaksi. Käyttäjystävällinen ympäristö ei välttämättä ole yksinkertainen, jos sitä optimoidaan. Optimoimalla oikein suurta VRML-tiedostoa niin, se mahdollistaa sen sujuvan käyttämisen tietokone- ja verkoympäristössä. (Ashdown & Forestino 1998.)

Internet-sivulla julkaistavien objektien luomisessa kannattaa välttää turhien yksityiskohtien mallintamista. Mitä useampia geometrisiä yksityiskohtia objektissa on, sitä enemmän siinä on polygoneja. Lisääntyvä polygonien määrä vaikuttaa selaimen renderöintinopeuteen negatiivisesti. Sopivan yksityiskohtien lukumäärän määrittäminen objektille on hankalaa, joten yleensä tarvitsee tehdä hieman kokeilua polygonien määrää pienentämällä ja kasvattamalla. Alle tuhannen polygonin käyttäminen mallissa takaa se, että ympäristöä voi tutkia myös hitailla Internet-yhteyksillä, mutta mallien visuaalinen ulkonäkö ei ole laadukas. (Ashdown & Forestino 1998.)

Perusgeometriasolmuja kannattaa käyttää aina kun se on mahdollista. Esimerkiksi laatikko- ja pallo-objektit voidaan myös luoda IndexedFaceSet-solmulla ja määrittellä pinnat tarkasti. Mutta tällöin tarvitaan enemmän koordinaatteja sekä menetetään mahdollisuus objektin pintojen määrän muuttamiseen dynaamisesti sen ollessa tietyllä etäisyydellä katsojaan nähden. Tar-

vittavien solmujen määrää voidaan myös vähentää huolellisella rakennehierarkia suunnittelulla. Usein toistuvien solmujen määrittelyyn tiedostossa kannattaa käyttää DEF- ja USE-syntaksi määritelmää (katso Solmun nimeäminen ja käyttäminen s. - 12 -). Lisäksi PROTO- tai EXTERNPROTO-solmuja voidaan käyttää samantyyppisten objektien määrittämiseen. (Ashdown & Forestino 1998.)

Teksturointi on kätevä ja tehokas tapa ilmaista objekteja monimutkaisessa ympäristössä ilman täsmällistä geometristä mallintamista. Esimerkiksi rakennus voidaan luoda lisäämällä laatikon malliseen objektiin ikkunoita ja ovia. On kuitenkin pidettävä huolta kuvatiedoston koosta pienentämällä resoluutiota, värisyvyysä tai pakkaamalla kuvaa. Polygonien määrää voidaan edelleen vähentää käyttämällä Billboard-solmua. Billboard-objekti on aina katsojaan päin, joten vain yksi objektin sivu pitää laskea reaaliaikaisesti. Kun jätetään katsojalle näkymättömät pinnat laskematta, niin se säästää ympäristön mallintamisaikaa ja käyttäjän lataaman ympäristön laskemisaikaa. Billboard-solmu soveltuu tasomaisten objektien, kuten puiden esittämiseen. (Ashdown & Forestino 1998.)

LOD-solmulla (level of detail) voidaan määritellä saman objektin eriasteiset yksityiskohdat ympäristön. LOD-solmun avulla VRML-selain kykenee esittämään ympäristössä olevia objekteja tehokkaasti, koska käyttäjästä kaukana olevia objekteja ei tarvitse piirtää yhtä tarkasti kuin lähellä olevia objekteja. Lisäksi samaan aikaan esitettävien yksityiskohtien määrää voidaan hallita käyttämällä VisibilitySensor-solmua tai määrittelemällä näkyvyyden raja-arvo NavigationInfo-solmulla. Aina kun on mahdollista, kannattaa lopettaa piilossa olevien pintojen renderöiminen käyttämällä ElevationGrid-, Extrusion- ja IndexedFaceSet- solmujen Solid-kentän arvona TRUE. (Ashdown & Forestino 1998.)

Monet VRML-selaimet tukevat VRML-tiedostojen viiveellistä lataamista Inline-solmua käyttämällä. Tämä tarkoittaa sitä, että käyttäjä voi aloittaa ympäristön tutkimisen ja perusnäkökuvan kanssa toimimisen samalla, kun loppuja ympäristön osia ladataan. Kun ladattavaksi jääneet osat ovat latautuneet, niin ne lisätään ympäristöön automaattisesti. Kun VRML-ympäristön luomiseen käytetään useita pieniä tiedostoja lyhentää se ympäristön esittämisen aloittamiseen kuluvaa aikaa. (Ashdown & Forestino 1998.)

5 VRML:N LUONTI JA KÄYTTÖ

5.1 VRML-mallin luonti 3D-grafiikkaohjelmalla

5.11 Geometria

3ds Max -ohjelmalla voidaan helposti luoda 3D-malleja ja ympäristöjä, jotka voidaan tallentaa VRML-tiedostoksi VRML97-vientiliitännäisellä (engl. export plug-in). Ohjelmalla mallinnettaessa yksikköjärjestelmä tulee olla perusasetuksella ja ympäristön mittayksikkönä yksi metri. Ympäristöön kannattaa lisätä yksi valo ja kamera, koska yleensä ohjelman perusvalo ei riitä valaisemaan ympäristöä riittävästi ja koska kamerasta luodaan ympäristöön perusnäkyvä (engl. viewpoint). (Autodesk 2003b.) Ympäristön hyvän suorituskyvyn aikaansaamiseksi VRML-selaimella, täytyy pitää mallinnettavien objektien polygonien määrä hyvin pienenä. Valmis ympäristö, jossa on enintään 5000–10 000 polygonia yhteensä, toimii hyvin useimmissa tietokoneissa. (Autodesk 2003b.)

Perusprimitiivien käyttäminen mallissa on suotavaa aina kun se on mahdollista. Perusprimitiivejä ovat pallo-, laatikko-, kartio- ja sylinteri-objektit. Esimerkiksi ympäristöön on mallinnettu vain yksi pallo, ja se tallennetaan VRML-tiedostoksi niin, tiedoston koko on noin 0,4 kt. Jos kyseisen pallon pintaverkosta muutetaan yhden verteksipisteen paikkaa, niin se kasvattaa tallennettavan tiedoston kokoa. Muutetun pallon VRML-tiedoston koko on noin 7,4 kt, koska palloa ei voida viedä enää perusprimitiivinä. (Autodesk 2003b.)

Ohjelmassa on polygoneja laskeva apuohjelma (polygon counter), joka näyttää valitun objektin ja ympäristön pintojen määrän. LOD-apuobjekti (Level of detail) mahdollistaa ympäristön objektien pintojen määrän muuttamisen ja niiden määrän määrittämisen. Tiedoston optimointiin voidaan myös käyttää ohjelman omaa optimointitoimintoa (Optimize modifier) yksinkertaistamaan objekteja ennen niiden tallentamista VRML-tiedostoksi. Tiedostokokoa voidaan pienentää käyttämällä ohjelmassa olevaa älykästä kopiointiominaisuutta, josta käytetään instanssivaihtoehtoa monistettavalle objektille. (Autodesk 2003b.)

3ds Max -ohjelmalla voidaan piilottaa tai poistaa objektin pintoja, mutta piilotetut pinnat näkyvät aukkoina objektin pinnassa tiedostoa katseltaessa VRML-selaimella. VRML-vientiliitännäinen voi viedä piilotettuja objekteja tiedostoon, mutta se ei voi viedä piilotettuja pintoja. Pinnat, joita ei koskaan näe objektista, voidaan piilottaa tai poistaa. Esimerkkejä näistä ovat objektien takapinnat tai jotain toista pintaa vasten olevat sivut. (Autodesk 2003b.)

5.12 Animaatio

3ds Max -ohjelmalla pystytään animoimaan objektien muutoksia ja viemään ne VRML-tiedostoksi VRML97-vientiliitännäisellä. VRML-vientiliitännäinen tukee objektien liike-, pyörähdys- ja skaalausanimaatioita sekä koordinaatti-interpolointi-animaatioita. Animaatioita käytettäessä tallennettavan tiedoston koko kasvaa huomattavasti. Koordinaatti-interpolointi-animointia ovat esimerkiksi objektin taivutuksen (Bend modifier) ja objektin parametrimuutoksen sekä space warp -objektin parametrien animointi. (Autodesk 2003b.) Objektin pintaverkkoon kohdistuvissa animaatioissa ei saa poistaa tai lisätä objektin pintoja tai verteksipisteitä, koska VRML-kieli ei tue niiden lisäämistä tai poistamista. Ympäristössä voidaan myös käyttää hierarkia-animaatioita, joissa lapsiobjektit (engl. child object) perivät vanhemman (engl. parent object) objektin muutokset automaattisesti. Käänteiskinematikka (engl. inverse kinematics) animaatiot tallentuvat myös VRML-tiedostoksi, kuten ne ovat animoitu ohjelmalla. (Autodesk 2003b.) Animaation ohjaimina kannattaa käyttää TCB-ohjaimia, koska niillä voidaan ohjata animaatiota tarkasti ja luoda pienin mahdollinen tiedostokoko. Toiset ohjaimet, kuten Bezier-ohjaimet saavat aikaan suurempia tiedostoja eivätkä toimi yhtä hyvin kuin TCB-ohjaimet. (Autodesk 2003b.)

5.13 Teksturointi

Teksturoinnissa kannattaa käyttää pieniä kuvatiedostoja, koska kuvat kasvattavat tallennettavan tiedoston kokoa. Kuvien etuna on, että niillä saadaan aikaan yksityiskohtia objekteihin mallintamatta niitä ohjelmalla. Objektien materiaaleina voidaan käyttää vain standardi- ja multi / sub-object -materiaaleja. Multi / subobject -materiaalia käytettäessä tallentuu objektista niin monta kopiota, kuin alimateriaaleja on käytetty materiaalissa. (Autodesk 2003b.)

Seuraavat materiaalien ominaisuudet ovat tuettuja VRML-tiedostossa:

- diffuusi-, ambient- ja speulaariväri.
 - yksi 2D-kuva, jonka pitää olla diffuusikanavalla. Tuettuja kuvatiedostoja ovat jpg-, gif- ja png-tiedostot.
 - pinnankiilto, mutta ei sen voimakkuus
 - lankamalli
 - läpinäkymättömyys
- (Autodesk 2003b.)

5.2 Vientiliitännäiset

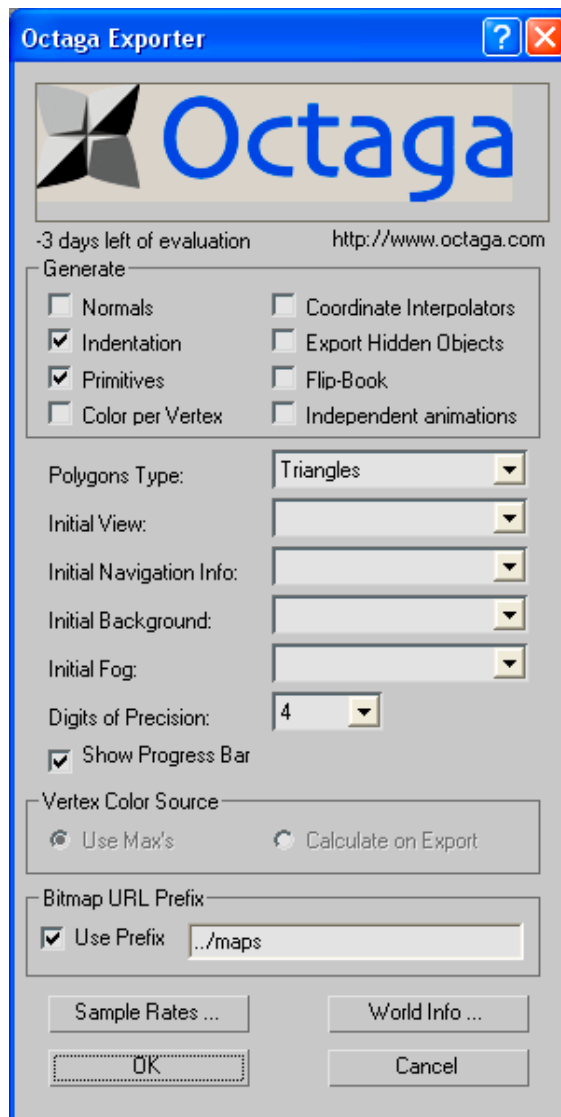
5.21 Octaga Exporter

Octaga Exporter on hyvin käyttökelpoinen työkalu 3ds Max:n käyttäjälle. Se helpottaa 3D-mallien reaaliaikaisen esityksen suunnittelussa ja säästää aikaa ja vaivaa viemällä visuaaliset efektit suoraan luotavaan tiedostoon. Octaga Exporter on saatavilla 3ds Max:n versioihin v5, v6 ja v7. Octaga Exporter on laajennettu versio 3ds Max -ohjelmassa olevasta VRML97-vientiliitännäisestä. Tämä liitännäinen mahdollistaa VRML-koodatun X3D:n ja MPEG4-solmujen käyttämisen Octaga-toisto-ohjelmien kanssa. (Octaga 2001.)

Vientiliitännäinen tukee seuraavia ominaisuuksia:

- Multi-texture: Käytetään reaaliaikaisen tekstuuri efektin luomiseksi. Yhdistelmäpinnoitteet, kuten Mix- ja RGB multiply -kartat, yhdistävät värejä ja pinnoitteita yhdeksi pintamateriaaliksi. Tämä on hyödyllinen ominaisuus, joka mahdollistaa monien pintamateriaalien sekoittamisen ja lisäämisen malliin. (Octaga 2001.)
- Bump ja heijastuminen: Bump- ja heijastuskartat toimivat materiaali editorissa kuten bittikartat, jotka otetaan käyttöön tietyn valikon kautta (Octaga 2001).
- Shell-materiaali: Shell-materiaalia käytetään helpottamaan vaikeimpien tekstuuri efektien kuten tekstuuriksi renderöintiä (engl. render to texture) käyttöä ja sen tuomista reaaliaikaiseen käyttöön (Octaga 2001).
- Partikkelit: Partikkeleita voidaan lisätä luomaan uusia efektejä reaaliaikaiseen malliin (Octaga 2001).
- Kolmioliuskat (engl. triangle strips): Octaga Exporter pystyy optimoimaan malleja kääntämällä mallin geometrian kolmioliuskoiksi, joka nopeuttaa reaaliaikaista renderöintiä (Octaga 2001).
- Läpinäkyvyys ja väri: Läpinäkyvyyttä ja väriä voidaan animoida ja ne voidaan tallentaa tiedoston mukana efekteiksi (Octaga 2001).
- Pinnan normaalit: Voidaan luoda animaatioita, joilla muutetaan objektin pintaverkko ja samalla säilytetään objektin pinnansileys (Octaga 2001).

Partikkelit ovat MPEG4-solmuja ja toimivat vain Octaga Professionalissa. Monitekstuuri-, kuhmutus- ja ympäristökartat toimivat kaikissa Octagan ohjelmissa. (Octaga 2001.)



Kuva 7 Octaga Exporter -vientiliitännäisen käyttöliittymä.

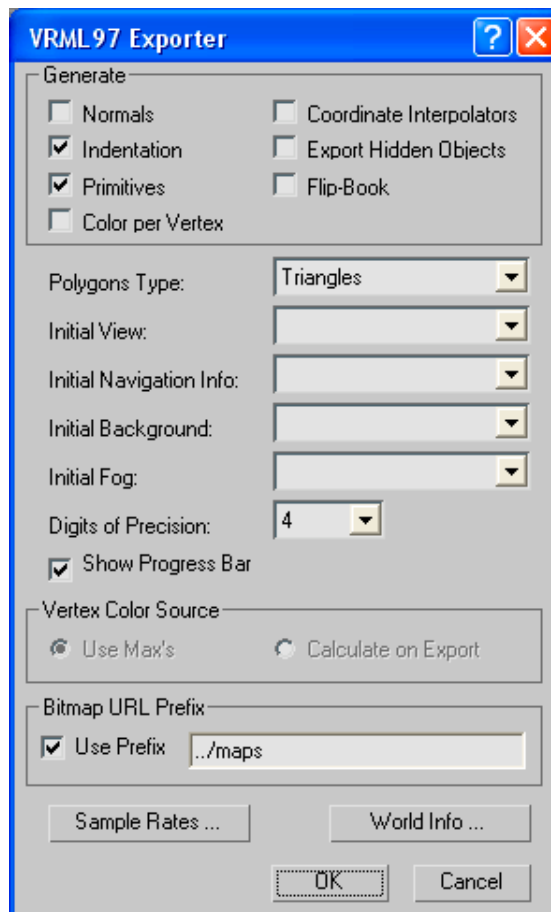
5.22 VRML97 Exporter

VRML97-vientiliitännäinen on sisäänrakennettu 3ds Max 6.0 -ohjelmaan. Liitännäisen avulla *.max -tiedosto voidaan viedä VRML-tiedostoksi ja tallentaa se kovalevylle.

Vientiliitännäinen tukee seuraavia ominaisuuksia:

- Valot: kaikkia VRML-kielen valot tuettu (suunnattu, piste- ja kohdevalo)
- Teksturointi: 2D-kuvat, diffuusiväri
- Kameran: perspektiivi ja ortogonaalinen kamera (engl. free ja target camera)
- Animaatio: Liike-, pyörähdys- ja skaalausanimaatio sekä koordinaatti-interpolointi animaatio

(Autodesk 2003a.)



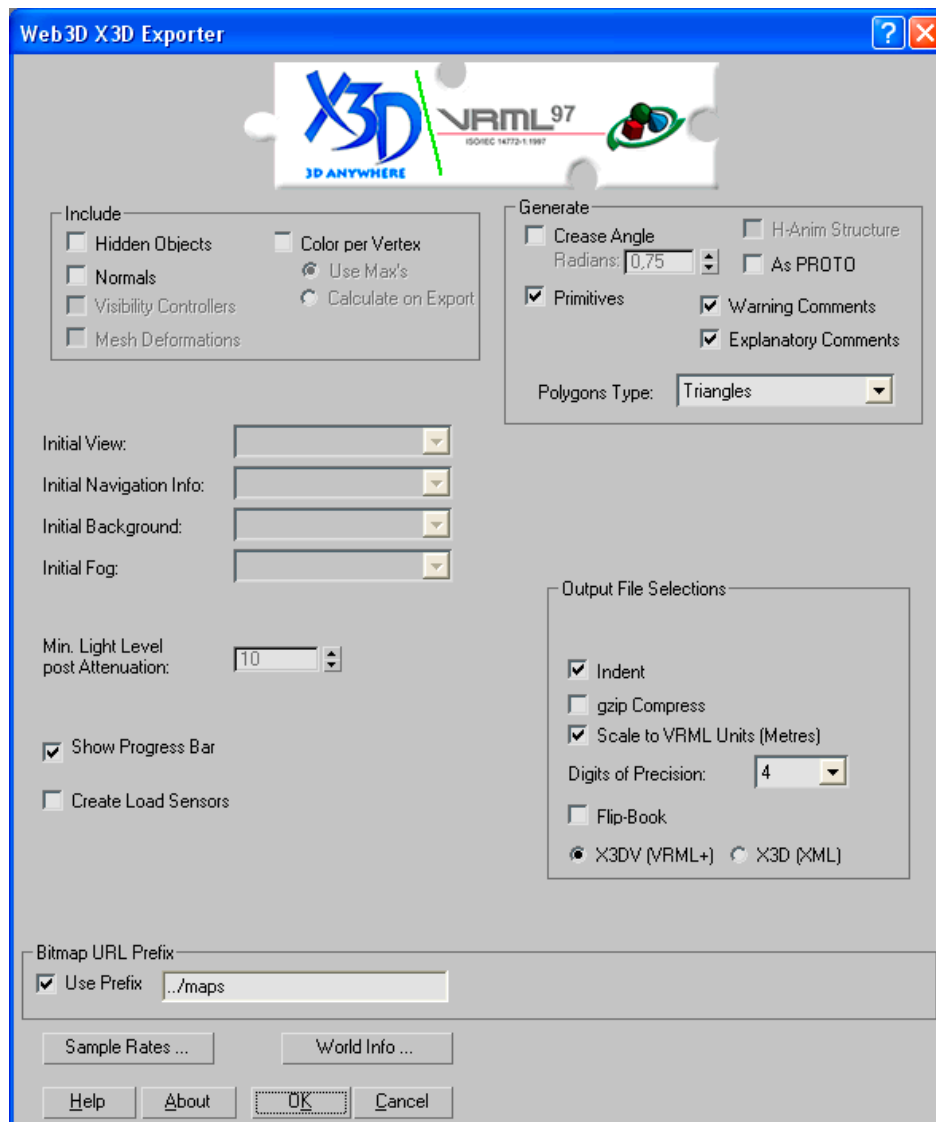
Kuva 8 VRML97 Exporter -vientiliitännäisen käyttöliittymä.

5.23 Web3D X3D Exporter

Web3D X3D -vientiliitännäinen pohjautuu Mark Callow:n alkuperäiseen VRML-vientiliitännäiseen, joka kuuluu 3ds Max -ohjelman kehityspakettiin – max SDK:hon (engl. Software Development Kit). X3D-liitännäisen avulla voidaan *.max tiedosto viedä VRML-koodattua X3D:tä X3D-tiedostoksi tai XML-koodatuksi X3D-tiedostoksi. (Unreal – Realm of concepts 2006.)

Vientiliitännäinen tukee seuraavia ominaisuuksia:

- Valot: kaikkia VRML-kielen valot tuettu (suunnattu, piste- ja kohdevalo)
 - Teksturointi: 2D-kuvat, diffuusiväri ja multi-texture -materiaali
 - Kameranat: perspektiivi ja ortogonaalinen kamera (engl. free ja target camera)
 - Geometria: NURBS-objektit
- (Unreal – Realm of concepts 2006.)



Kuva 9 Web3D X3D Exporter -vientiliitännäisen käyttöliittymä.

5.3 Selaimet

5.31 Yleistietoa

Hyperteksti pohjautuu 1950-luvulla esitettyyn ajatukseen, jossa kuvaruudulla esitettiin ”kuumia” visuaalisesti tekstistä erottuvia sanoja. Sanoissa oli linkki muuhun informaatioon, ja edellisen tekstiin pääsi takaisin linkin avulla. Avautuneessa tekstissä pystyi myös jatkamaan uusien linkkien seuraamista ja uuden informaation omaksumista.

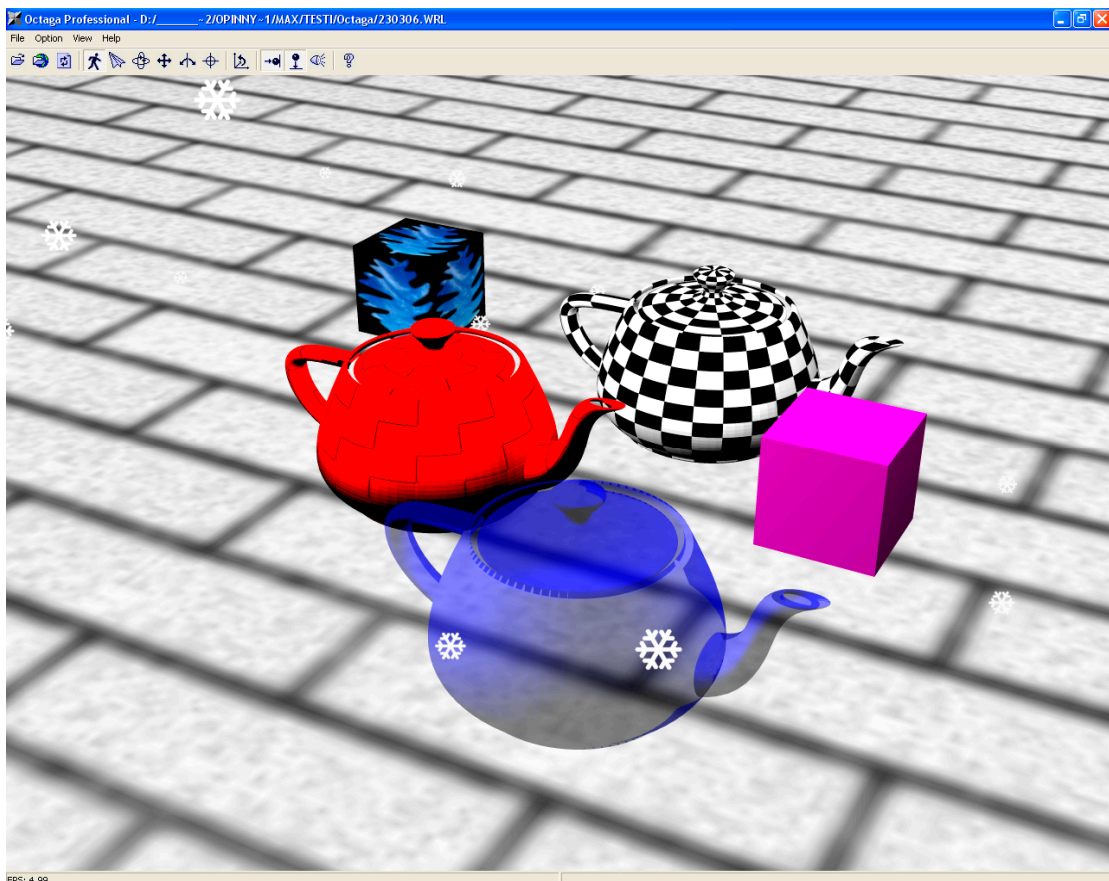
1980-luvun lopussa syntyi ehdotus hypertekstiverkon luomisesta. Ehdotuksen pohjalta ensimmäinen hypertekstiverkossa toimiva merkkipohjainen selain syntyi vuonna 1990. Hypertekstiverkko tunnetaan nykyisin nimellä WWW (World Wide Web). WWW:n suosi kasvoi merkittävästi kun ensimmäinen graafinen selain Mosaic julkaistiin 1990-luvun alussa. Mosaicia seurasivat muutaman vuoden kuluttua kehittyneemmät graafiset WWW-

selaimet kuten Netscape Navigator ja Microsoft Internet Explorer. WWW:n kasvu ja yleistymisen olivat räjähdysmäisiä muutama vuosi sen jälkeen kun uudet graafiset selaimet julkaistiin. Suurin syy WWW:n suosioon oli URL:n tuoma helppous löytää lähes mitä tahansa Internetissä sijaitsevaa tietoa ja HTML:n määrittelemä standardi tapa esittää graafista informaatiota.

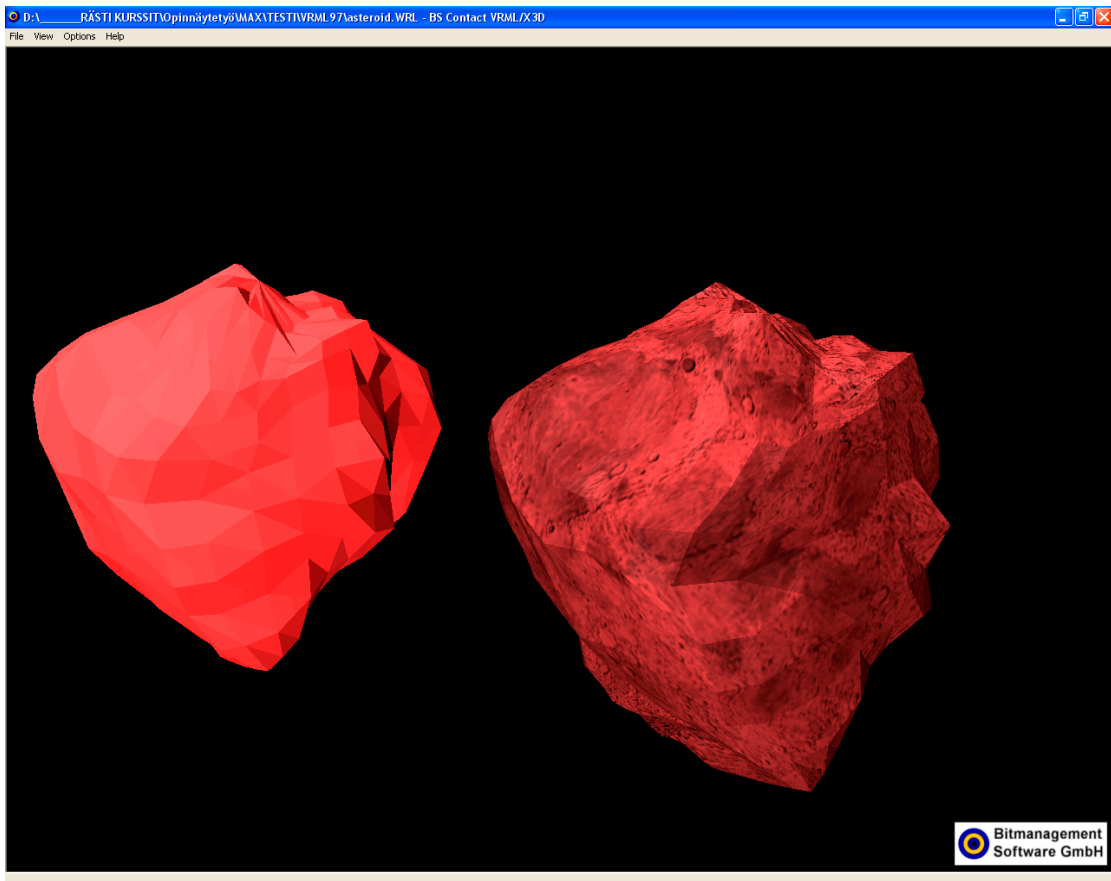
Internet-selain kuten Netscape Navigator, Microsoft Explorer ja Mozilla Firefox voivat normaalisti esittää vain tekstiä ja kuvia. Kun halutaan katsoa Internet-selaimella muutakin kuin kuvia ja tekstiä, kuten esimerkiksi VRML-kielellä luotua virtuaalimaailmaa niin tarvitaan erillinen VRML-selain. Internet-selain aukaisee automaattisesti koneeseen asennetun VRML-selaimen, kun wrl-tiedosto on ladattu. (Ashdown & Forestino 1998.)

5.32 Liitännäiset ja katseluohjelmat

Useimmat VRML-selaimet toimivat Internet-selaimen liitännäisinä, joissa 3D-ympäristö ja käyttöliittymä näytetään Internet-selaimen pääikkunassa. Tämä mahdollistaa VRML-ympäristön ”upottamisen” HTML-dokumenttiin. Muutamia VRML-selaimia esittävät VRML-maailman itsenäisenä erillään Internet-selaimesta. VRML-selainliitännäiset ja -katseluohjelmat pitää asentaa apusovelluksiksi toimiakseen yhdessä Internet-selaimen kanssa. (Ashdown & Forestino 1998.)



Kuva 10 Octaga Professional -katseluohjelman käyttöliittymä.



Kuva 11 BS Contact -katseluohjelman käyttöliittymä.

5.33 Navigointi ja vuorovaikutus

VRML-selaimissa on useita tapoja VRML-ympäristön tutkimiseen tai sen läpi kulkemiseen. Yleensä vuorovaikutusmenetelmät on suunniteltu eri tavoilla eri selaimille, mutta ne pohjautuvat tutkimiseen, lentämiseen, kävelemiseen sekä klikkaa ja etsi -teemoihin. (Ashdown & Forestino 1998.)

Tutkimismoodia käytettäessä voidaan kohdetta kääntää tai liikuttaa sitä suhteessa katselupisteeseen. Lentomoodi simuloi ympäristön läpi liikkumista, jolloin nopeutta ja suuntaa säädellään hiirellä tai näppäimistöllä. Kävely-moodi on samankaltainen lentomoodin kanssa, paitsi että näkymää seurataan käyttäjän katselupisteestä. Hakumoodi tukee osa selaimia, jolla voidaan klikata objektia hiirellä, ja katselupiste siirtyy automaattisesti sitä kohti. Käyttäjällä on yleensä mahdollisuus vaihtaa näkymää eri katselupisteiden välillä VRML-selaimissa, jotka ovat määritellyt VRML-tiedostossa. VRML-selaimista löytyy myös otsavalot, jolla voidaan valaista ympäristö katselupisteen edestä. Lisäksi tapahtumat, jotka sallii käyttäjän valita objektin klikkaamalla sitä hiirellä, tarkoituksenaan avata linkki tai aktivoidakseen sensorin. (Ashdown & Forestino 1998.)

5.34 VRML-tiedoston upottaminen HTML:ään

Internet-selaimiin ladattavissa olevat VRML:ää tukevat liitännäiset mahdollistavat sisällön luojalle HTML:n ja VRML:n elementtien yhdistämisen Internet-sivulla kehyksien ja EMBED-tagin avulla. EMBED-tagin on HTML-laajennus, joka sallii objektien, kuten videon ja äänen upottamisen HTML-sivuun ja sen esittämisen sopivalla liitännäissovelluksella. Esimerkiksi seuraava koodi rivi HTML-tiedostossa pakottaa VRML-maailman nimeltä model.wrl avautumaan ruutuun, jonka koko on 400x300 pikseliä.

```
<EMBED src="model.wrl" WIDTH=400 HEIGHT=300>  
(Ashdown & Forestino 1998.)
```

Internet-selaimella katsottavien virtuaaliympäristöjen ikkunan koko kannattaa aina määrittellä HTML-koodissa, jotta käyttäjä ei pysty muuttamaan ikkunan kokoa. Suureen ikkunaan avautuva virtuaaliympäristö vaati paljon laskentatehoa tietokoneelta, joka voi tehdä ympäristössä liikkumisen hitaaksi ja nykiväksi. Pienen ikkunakoon vuoksi virtuaaliympäristössä liikkuminen on huomattavasti jouhevampaa, varsinkin monimutkaisissa ympäristöissä. (Ashdown & Forestino 1998.) Joskus on hyödyllistä esittää VRML-maailma siten, että VRML-selaimen käyttöliittymä on poistettu käytöstä. HTML-koodissa lisätään EMBED-elementtiin seuraava:

```
VRML-DASHBOARD="FALSE"  
(Ashdown & Forestino 1998.)
```

Sama voidaan tehdä myös wrl-tiedostoon lisäämällä NavigationInfo-solmu:

```
NavigationInfo { type "NONE" }  
(Ashdown & Forestino 1998.)
```

5.35 Kehykset

Kehykset mahdollistavat sivun pääikkunan jakamisen useampaan pieneen ikkunaan tai kehykseen, jossa jokaisessa on oma HTML-tiedosto. Kehykset yleensä sisältävät HTML-tiedostoja, mutta niitä voidaan yhtälailla käyttää VRML-maailmojen esittämiseen. (Ashdown & Forestino 1998.)

Kehyksiä sisältävän HTML-tiedoston luomiseen käytetään FRAMESET-elementtiä, joka määrittelee kehyksien ulkoasun. FRAME-elementillä määritellään tiedoston URL:n, jonka kehys sisältää. FRAME-elementti myös sisältää NAME-ominaisuuden, jolla nimetään jokainen kehys. Käyttämällä TARGET-ominaisuutta on mahdollista esittää tiedosto tietyn nimisenä, kun HTML-linkki on määriteltä seuraavasti:

```
<A href="doc.html" TARGET="frame_name"> ... </A>  
(Ashdown & Forestino 1998.)
```


VRML-linkit voidaan myös kohdistaa tiettyihin kehyksiin käyttämällä ANCHOR-solmun parameter-kenttää. VRML:ssä vastaava linkki olisi:

```
Anchor {  
    url "doc.html"  
    parameter "target=frame_name"  
    children {  
        tähän linkitetyt objektit  
    }  
}  
(Ashdown & Forestino 1998.)
```

5.36 Näkymä

Joissain VRML-selaimissa on mahdollista vaihtaa katselupiste käyttämällä <A> HTML-elementtiä. Esimerkiksi jos HTML-tiedosto sisältää seuraavan koodi rivin:

```
<A href="model.wrl#view1"> ... </A>  
(Ashdown & Forestino 1998.)
```

Tämä toiminto luodaan model.wrl -tiedostoon käyttämällä viewpoint-solmua, joka on nimetty DEF-komennolla VIEW1:ksi. VRML-selain esittää ympäristön siitä katselupisteestä, mikä määritelty kuhunkin linkkiin HTML-koodissa. Jos tätä tekniikkaa käytetään kehyksien kanssa, niin silloin VRML-selain vaihtaa katselupistettä toiseen lataamatta uudelleen koko wrl-tiedostoa. (Ashdown & Forestino 1998.)

5.37 VRML-selaimet ja -liitännäiset

VRML:n alkuaikoina oli vain kaksi merkittävää liitännäistä: Cosmo Player ja WorldView. Ne tukivat kaikkia yleisimpiä kuvatiedostoja, johon edes erillisinä sovelluksina toimivat VRML-selaimet eivät pystyneet (ks. Taulukko 2). Tällä hetkellä on saatavilla useita erilaisia VRML-selaimia ja liitännäisiä verrattuna 90-luvulla olevaan tarjontaan. Ne kaikki toistavat VRML-tiedostoja, ja suurin osa tukee X3D-tiedostoja sekä osalla selaimia pystyy myös katselemaan mallinnus ohjelmien luomia tiedostoja (ks. Taulukko 3).

Taulukko 2 Yleisimmät VRML-selaimet vuonna 1998.

Sovellus	Tyyppi	Käyttöjärjestelmä					Selain		Tuetut tiedostomuodot		
		Win	Linux	Mac	SGI	Sun Solaris	IE	Netscape	Kuva	Video	Ääni
Casus Presenter	S	X			X	X			gif jpg	-	mid wav
Community Place	S / L	X						X	bmp gif jpg	bmp gif	wav mod
Cosmo Player	L	X		X	X*		X	X	bmp gif jpg png	gif asf mpeg * qt*	midi wav aiff* au*
VRMLview	S	X	X		X				gif jpg	-	-
VRwave	S / L				X	X		X	gif jpg	-	-
WorldView	L	X		X			X	X	bmp gif jpg png ras ppm	asf	wav

Selitys: L=liitännäinen S=sovellus X=tuettu

(Ashdown & Forestino 1998.)

Taulukko 3 Yleisimmät VRML- ja X3D-selaimet vuonna 2006.

Sovellus	Tyyppi	Käyttöjärjestelmä			Selain			X3D-tuki
		Win	Linux	Mac	IE	Netscape	Firefox	
Cosmo Player	L	X			X	X	X	
Cortona	L	X		X	X	X	X	
Octaga	L / S	X	X		X	X	X	X
BS Contact	L	X			X	X	X	X
Flux	L	X			X			X
blaxxun Contact	L	X			X	X		
Venues	L	X			X			X
FreeWRL	L / S		X	X			X	X
OpenVRML	L / S		X	X			X	
Xj3D	J / T	X	X	X				X
Orbisnap	S	X	X	X				

Demotride	S	X						
Carina	S / L	X	X	X				X
VRMLview	S	X	X					
X3DToolKit	S / T	X	X	X				X
Selitys: L=liitännäinen S=sovellus X=tuettu T=toolkit J=Java								

(NIST 2006.)

6 CASE: VRML-mallin visualisointi

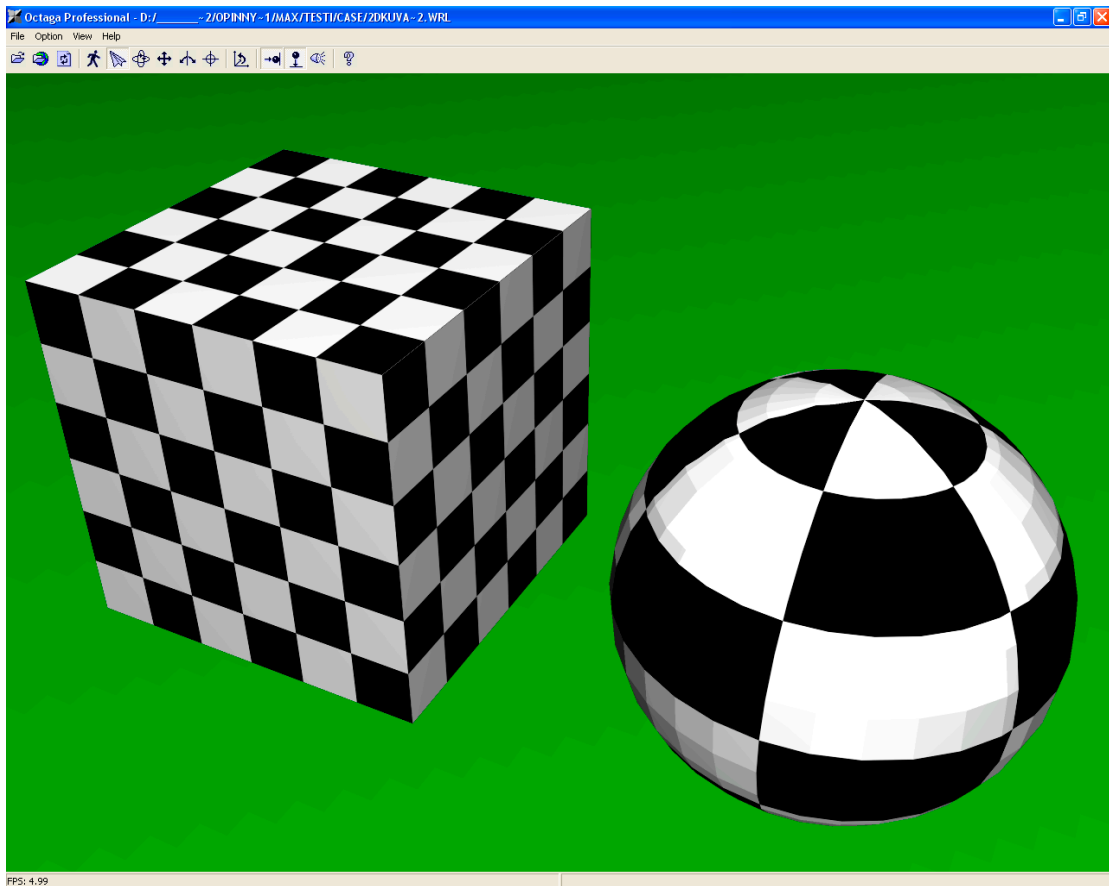
6.1 Yleistä

Tässä osiossa tutkitaan VRML-kielen tukemia visualisointiominaisuuksia käyttämällä niitä 3ds Max -ohjelmalla luotujen mallien visualisointiin sekä niiden siirtymistä VRML-tiedostoon. Lisäksi tutkitaan myös X3D-kielen tukemia visualisointiominaisuuksia samalla ohjelmistolla ja niiden siirtymistä X3D-tiedostoon. VRML- ja X3D-tiedostojen viennissä käytetään VRML97 Exporter, Octaga Exporter -ja Web 3D X3D Exporter -vientiliitännäisiä. Tallennettujen VRML- ja X3D-tiedostojen katseluun käytetään Octaga Professional -katseluohjelmaa sekä Java-kielellä toteutettua avoimeen lähdekoodiin perustuvaa Xj3D-selainta.

Tutkimuselosteessa käytetään hyvin paljon englanninkielisiä termejä, jotka liittyvät 3ds Max -ohjelman käyttöön ja materiaalien ominaisuuksiin. Teoriaosuudessa on käyty läpi seikkaperäisesti mitä 3ds Maxin ominaisuuksia kukin vientiliitännäinen pystyy viemään mukanaan tallennettuun tiedostoon. Tutkimus on jaettu kolmeen osa-alueeseen: objektien teksturointiin, animointiin ja geometriaan. Teksturoinnista tutkitaan heijastuskartan käyttöä, multi-texture -materiaaleja sekä bump mapping -ominaisuutta. Animaatioista tutkitaan hierarkia-animaatioita, valo- ja värianimaatioita. Lisäksi geometriaosuudessa tutkitaan NURB-pintamalleja ja partikkeleita.

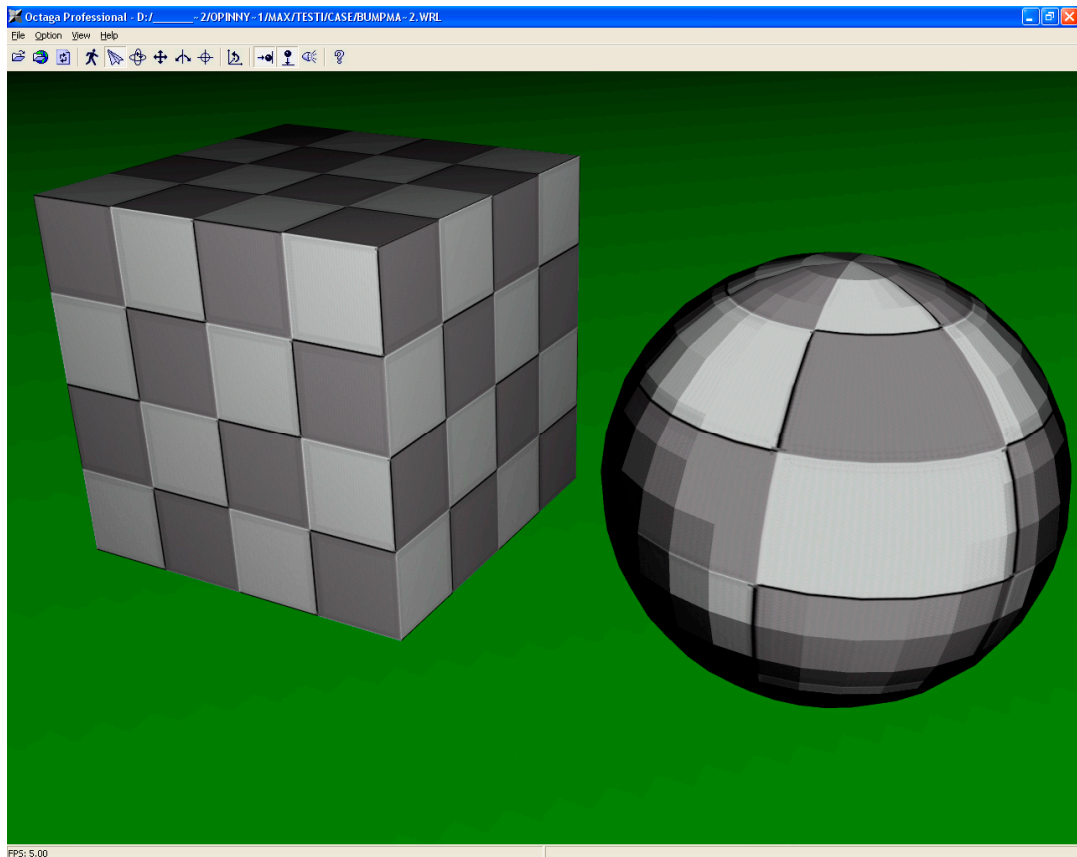
6.2 Teksturointi

Kaikki vientiliitännäiset tukivat 2D-kuvaa diffuse color -karttana, mutta kuvan tiling-arvon muuttamista tukivat vain VRML97- ja Octaga Exporter -vientiliitännäinen. Kun tiedostoja katsotaan selaimella ja vertaillaan niitä 3ds Max -ohjelmasta renderöityihin kuviin, niin ne näyttivät samanlaisilta keskenään. Web 3D -vientiliitännäinen ei tukenut tiling-arvon muuttamista, joten kuva piirtyy kappaleen pintaan niin kuin se näkyisi kuvankatseluohjelmassa.

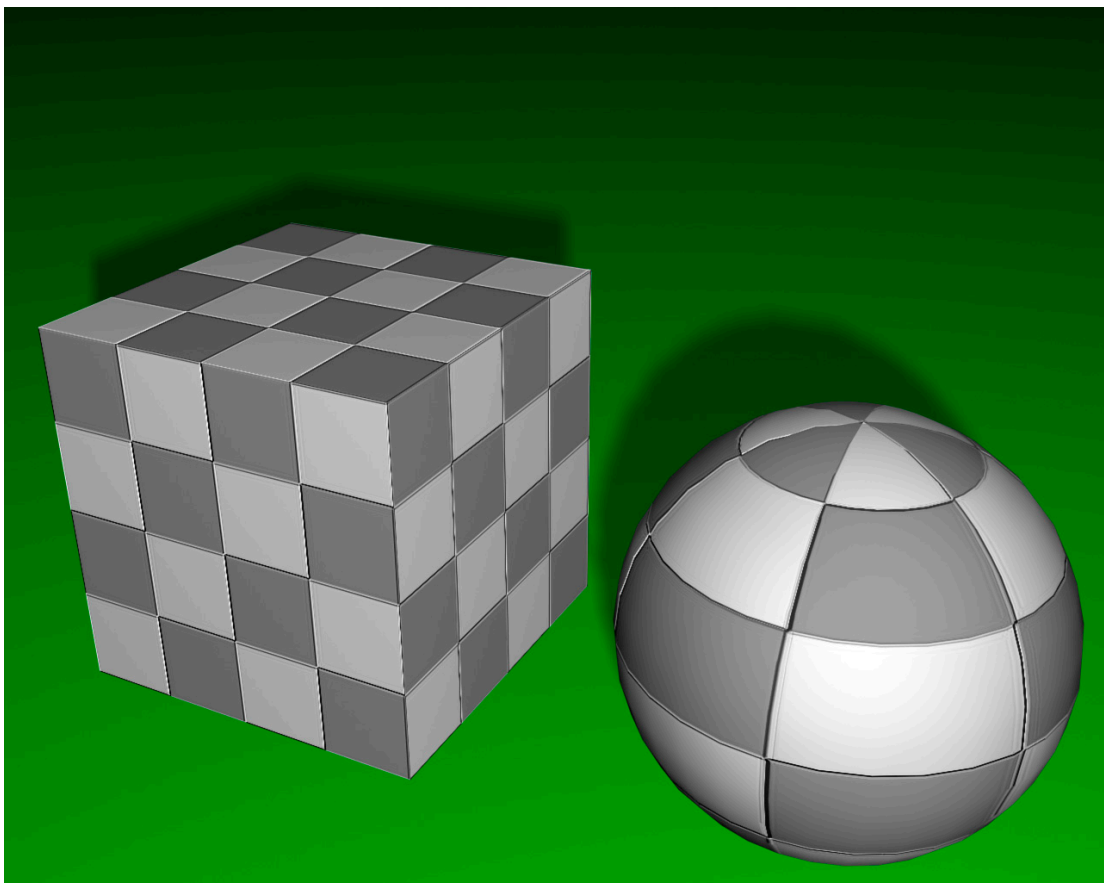


Kuva 12 Octaga Exporter:lla viety 2D-kuva.

Läpinäkyvyyttä tukivat kaikki vientiliitännäiset Opacity-karttaan liitetyn 2D-kuvan avulla. Lisäksi liitännäiset tukivat läpinäkyvyyden määrittelemistä muuttamalla parametria Opacity-liukusäätimestä. Octaga Exporter tuki bump mapping -ominaisuutta käyttämällä normaaliteksturoitua (engl. normal map) 2D-kuvaa liitettynä bump-karttaan. Bump-karttaan liitettävän kuvan tiling-arvo voidaan muuttaa samalla tavalla kuin diffuse color -kartan arvoja. Bump mapping -ominaisuus toimii kohtalaisesti taso-objekteissa ja objekteissa, joissa on tasopintoja, kuten laatikko ja pyramidi. Pyöreäpintaisissa pallo- ja teekannu-objekteissa bump mapping -ominaisuus toimi huonosti, koska näennäistä pintojen syvyyseroa ei pystynyt erottamaan väripintojen rajoista. Octaga Exporterin ja 3ds Maxin bump mapping -ominaisuudet tasopinnoilla ovat lähes samat, mutta pyöreiden pintojen toteutuksessa on merkittävä ero.

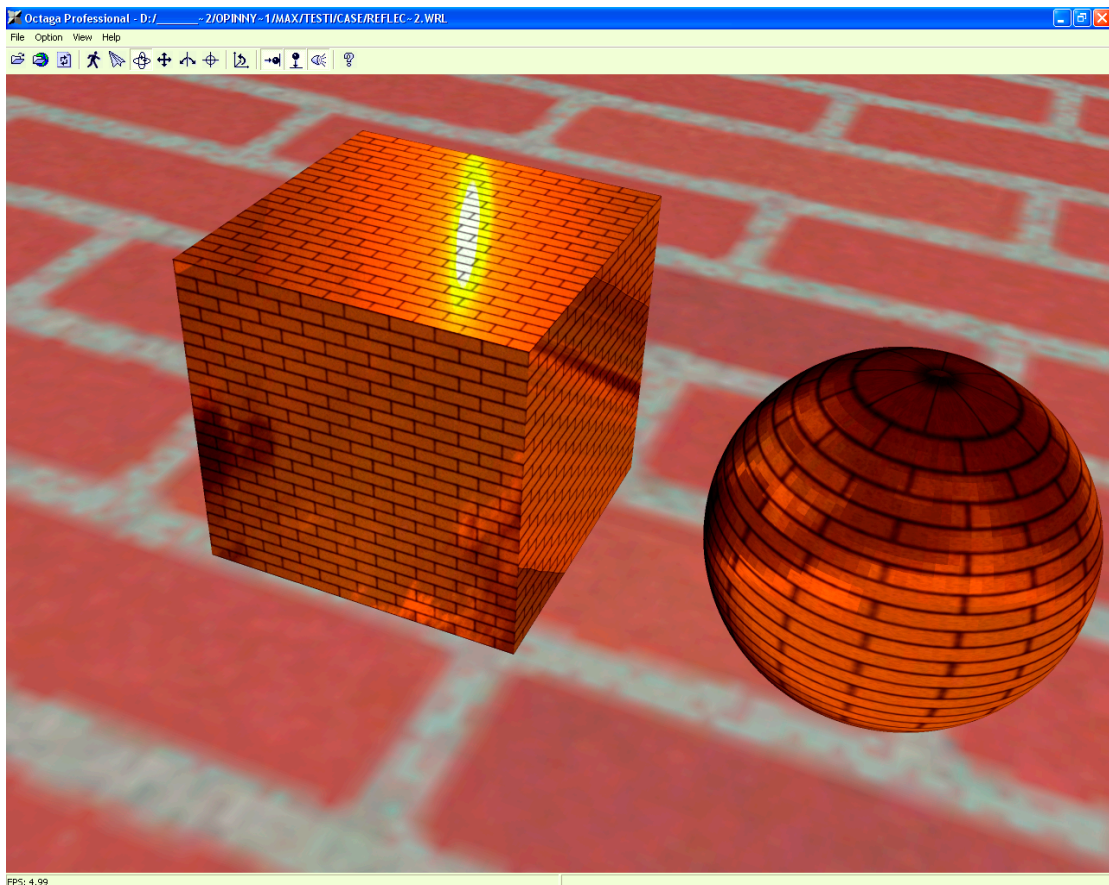


Kuva 13 Octaga Exporter:lla viety bump mapping -ominaisuus.



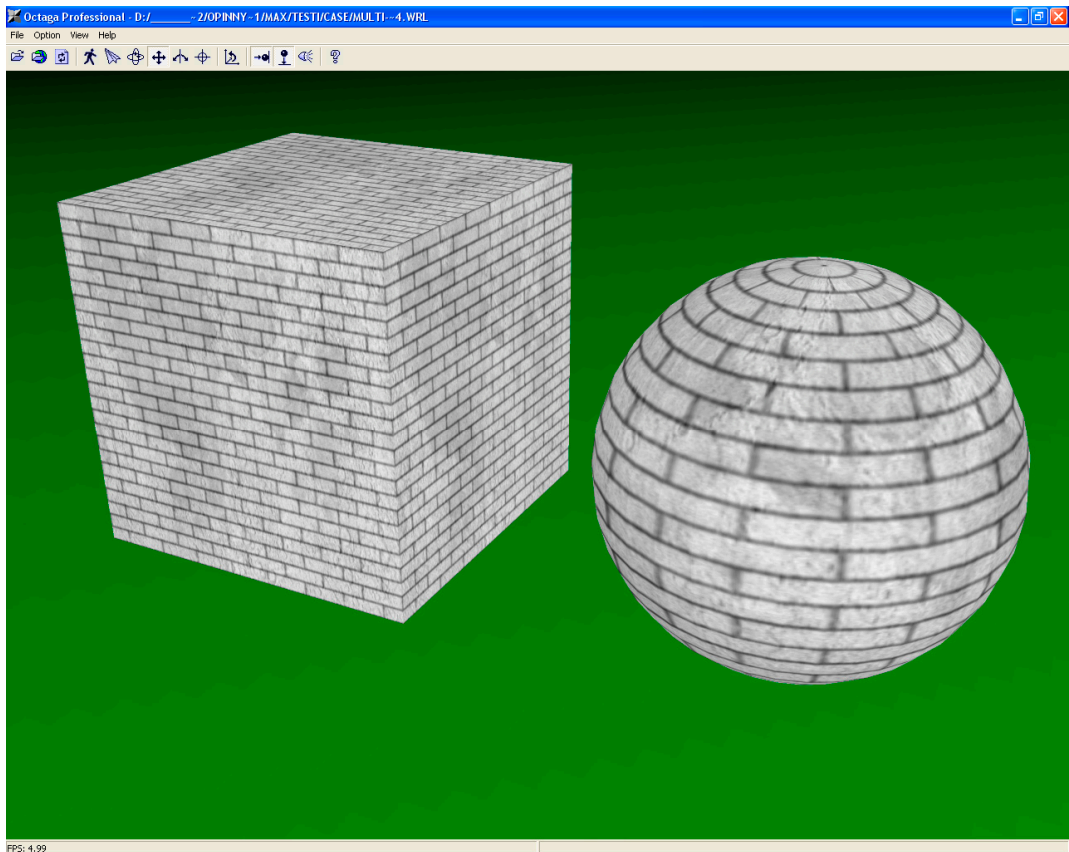
Kuva 14 3ds Max:sta renderöity bump mapping -ominaisuus.

Octaga Exporter oli ainoa vientiliitännäinen, joka tuki heijastuskartan (engl. reflection map) käyttöä objektin pinnalla. Objekteissa pitää käyttää UVW mapping -muokkainta, että heijastus toimii objektin pinnalla. Heijastuskartta toimii tasomaisissa pinnoissa kohtalaisesti, mutta pyöreissä pinnoissa se toimi, kuin pintaa olisi liitetty huonolaatuinen 2D-kuva ilman heijastusta.

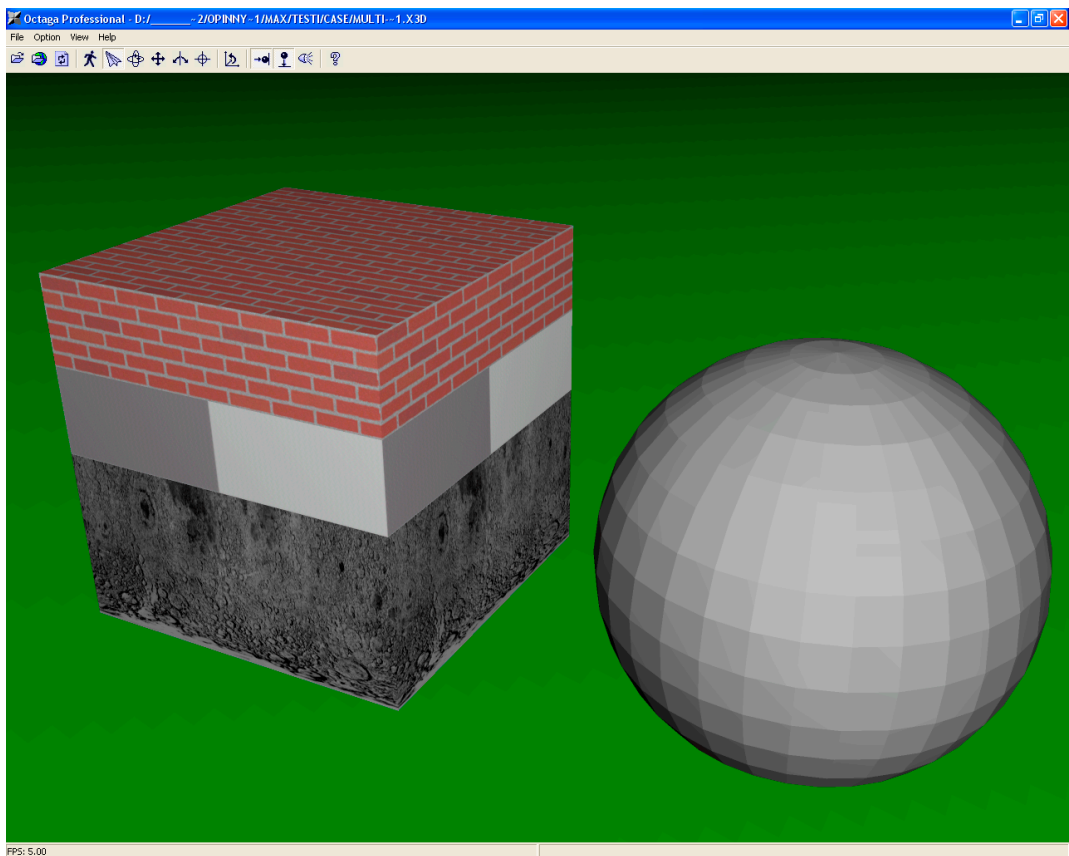


Kuva 15 Octaga Exporter:lla viety heijastuskartta, jossa aurinko ja pilviä.

Octaga Exporter tuki Multi-texturing -ominaisuutta mix-kartan avulla ja Web 3D suoraan multi / sub object -materiaalina. Mix-kartalla voidaan yhdistää kaksi väriä tai materiaalia yhdeksi yhdistelmäpinnoitteeksi objektin pinnalle. Vastaavasti multi / sub object -materiaalilla voidaan yhdistää satoja materiaaleja yhdeksi yhdistelmäpinnoitteeksi. Web 3D-vientiliitännäinen tukee ainakin kolmea alimateriaalia per materiaali. Alimateriaaleina ei voi käyttää 3ds Maxin 3D-materiaaleja, kuten esimerkiksi splatia ja speckleä, koska 2D-kuvat toimivat ainoastaan alimateriaaleina. Multi / sub object -materiaali voi olla käytössä ainoastaan yhdessä objektissa mallinnetussa ympäristössä. Jos ympäristössä halutaan käyttää useampaa Multi / sub object -materiaalia, niin täytyy luoda jokaiselle objektille oma materiaali. Lisäksi täytyy myös huomioida se, että 2D-kuvat eivät voi olla samannimisiä materiaaleissa. Jos 2D-kuvat ovat samannimisiä, niin materiaali näkyy vain ensimmäiseksi luodussa objektissa, ja loput objektit värjäytyvät materiaalin diffuse-värin mukaisesti. Lisäksi vientiliitännäinen ei tue multi-texturing -ominaisuutta pyöreäpintaisissa objekteissa, kuten pallo ja sylinteri.



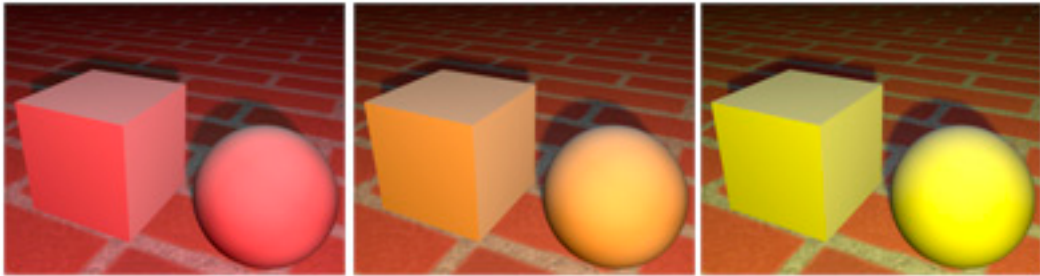
Kuva 16 Octaga Exporter:lla viety multi-texturing -ominaisuus.



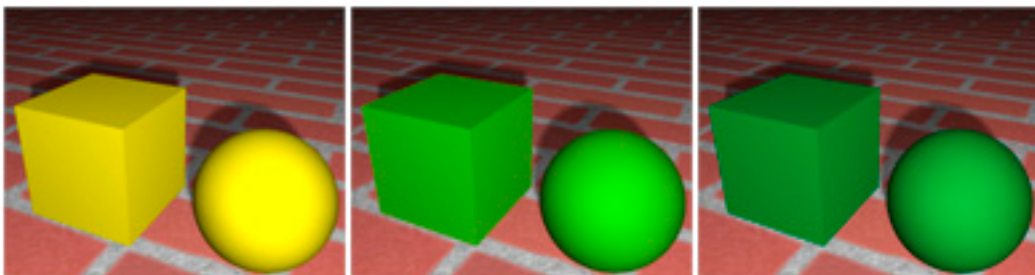
Kuva 17 Web 3D:llä viety multi-texturing -ominaisuus.

6.3 Animaatio

Octaga Exporter oli ainoa vientiliitännäinen, joka tuki objektin diffuusivärin animointia. Värit ja animaatio toistuvat katseluohjelmassa samalla tavoin kuin ne näkyvät mallinnusohjelmassa. Octaga Exporter:n lisäksi VRML97-vientiliitännäinen tuki valon värin animaatiota, ja tämäkin ominaisuus toistuu samalla tavoin katseluohjelmassa, kuin se näkyy mallinnusohjelmassa.

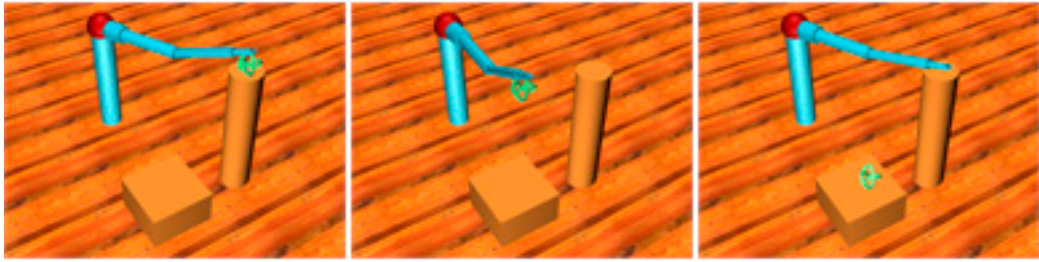


Kuva 18 Valon värin animointi.

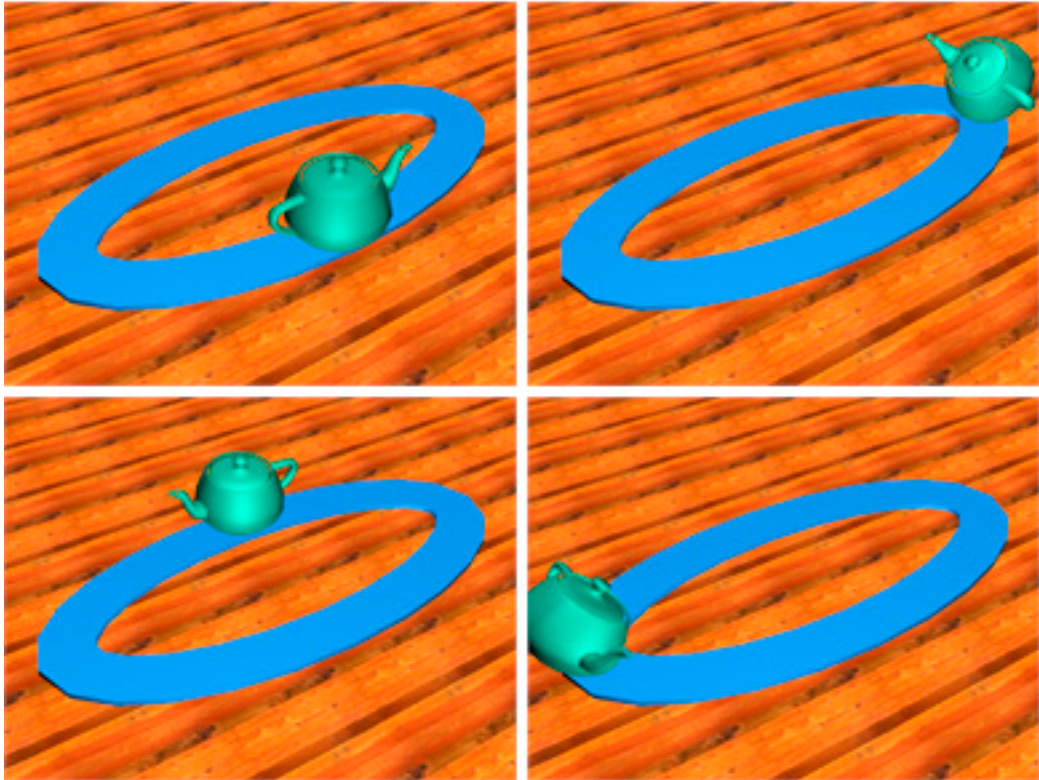


Kuva 19 Materiaalin diffuse-värin animointi.

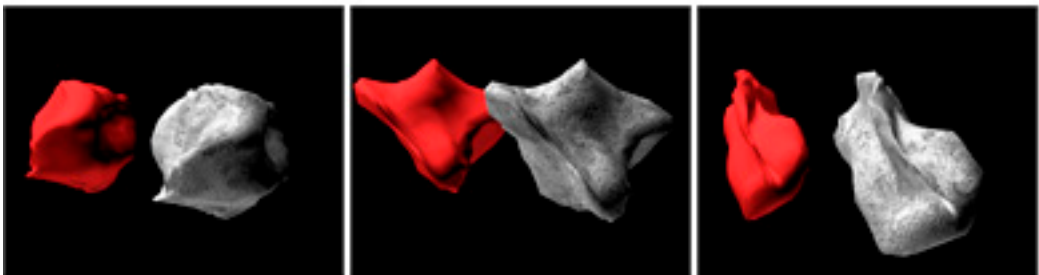
Octaga Exporter- ja VRML97-vientiliitännäinen tukivat käänteiskinematikan ja hierarkisten objektien animointia ja objektin pintaverkon muutosanimaatiota sekä objektien parametrianimaatioita. Animaatiot toistuvat katseluohjelmassa samalla tavoin, kuin ne näkyvät mallinnusohjelmassa. Octaga Exporter:illa viety käänteiskinematikan animaatio poikkeaa hieman mallinnusohjelmalla tuotetusta animaatiosta, koska "nosturin" varsi kulkee yhdessä kohdassa hieman eri kohdasta. Poikkeama johtuu todennäköisesti siitä, että animaatiopolun taivutussäde tai polun vertekspisteen kontrollointikahvan tyyppi muuttuu vietäessä animaatio toiseen tiedostomuotoon.



Kuva 20 Käänteiskinematikka-animaatio, jonka on toteutettu IK-nivelillä.



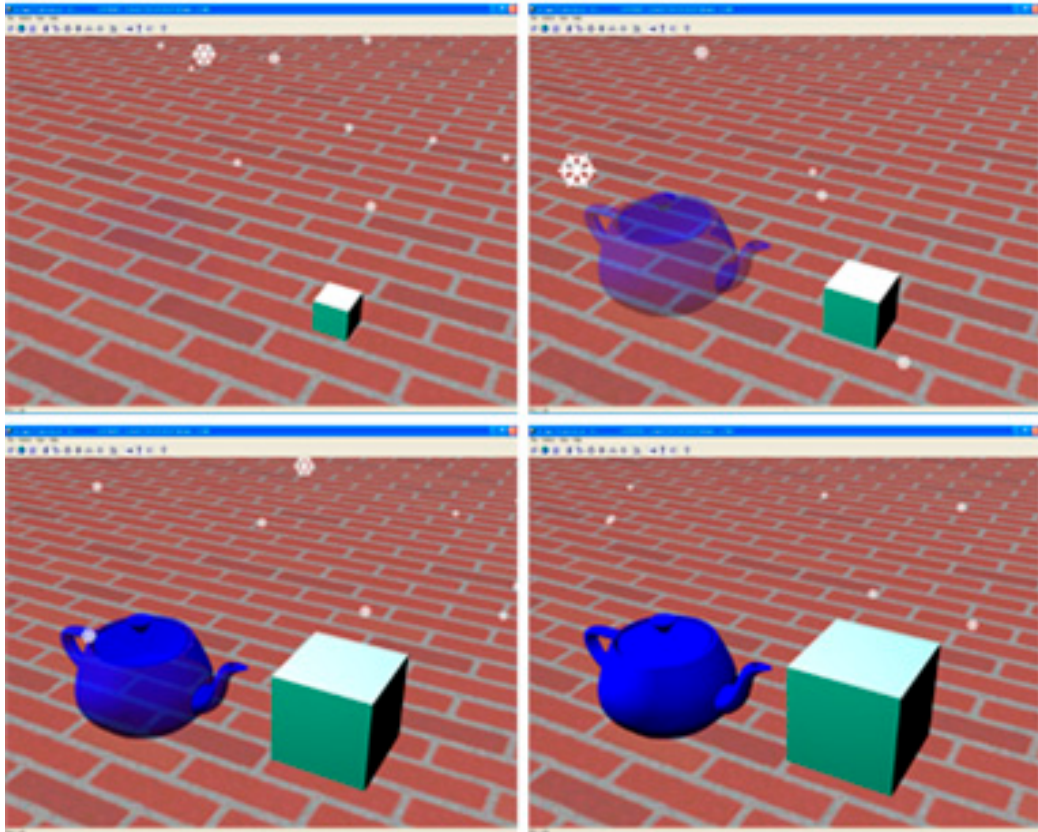
Kuva 21 Hierarkia-animaatio, joka on toteutettu animoidulla dummy-objektia.



Kuva 22 Pallon pintaverkkoa muokataan animoidulla noise-muokkaimella.

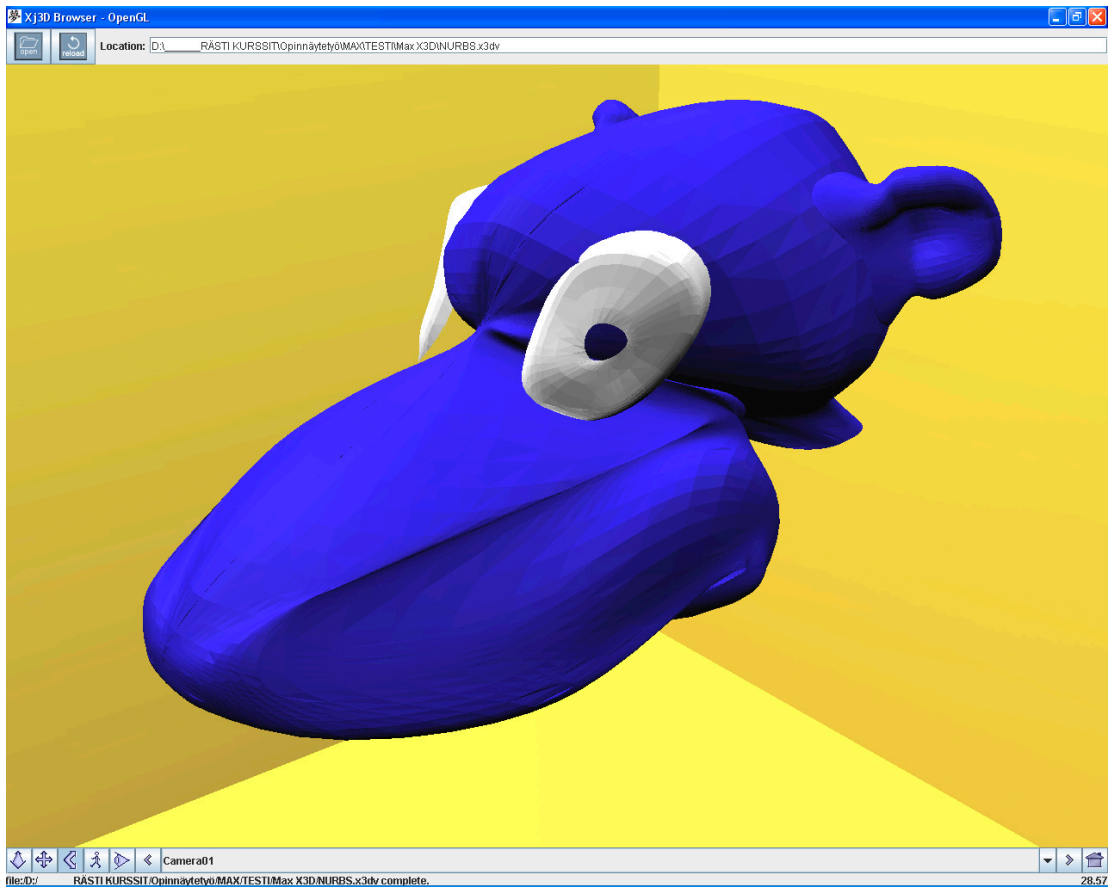
6.4 Partikkelit ja NURBS:t

Octaga Exporter oli ainoa vientiliitännäinen, joka tuki spray-partikkeleita. 3ds Max -ohjelmasta vientiliitännäinen vie tallennettavaan tiedostoon seuraavia ominaisuuksia, kuten partikkelien renderöinnin määrä, nopeus, muutos, koko ja elinaika. Muilla ominaisuuksilla ei ole vaikutusta partikkelien ulkonäköön tai vientiin. Partikkelien materiaalina käytetään ainoastaan 2D-kuvia ja ne täytyy lisätä diffuse color -karttaan.



Kuva 23 Octaga Exporter:lla viety spray-partikkelit.

Web 3d-vientiliitännäinen oli ainoa, joka tuki NURBS-pintamalleja. NURBS-pintamallin tekeminen on aloitettu CV curve -käyrillä, jonka jälkeen käyrät on yhdistetty U loft -pinnan rakennuskäskyllä. Mallin katselu onnistui vain Java-kielellä toteutetulla Xj3D-selaimella.



Kuva 24 CV curve -käyrillä luotu NURBS-pintamalli.

7 YHTEENVETO

Nykyään ainoa VRML-kielen käyttökelpoinen ominaisuus on animointi, kuten esimerkiksi hierarkia-animaatiot, objektin pintaverkon muutosanimaatio ja fysiikka-animaatiot. X3D-kielen teksturointiominaisuudet ovat paljon kehittyneemmät kuin VRML-kielen. X3D-vientiliitännäisellä voidaan viedä samat teksturointiominaisuudet 3ds Max:sta kuin VRML-vientiliitännäisellä sekä lisäksi bump mapping -ominaisuus, multi/sub object -materiaali sekä reflection map -ominaisuus. X3D-tiedostoon siirtyvät animaatio-ominaisuudet ovat täysin samat kuin VRML-kielessä. X3D-vientiliitännäisen geometriatuki on kehittyneempi kuin VRML:n, koska se tukee normaalin perusgeometrian lisäksi partikkeleita ja NURBS-geometriaa. VRML- ja X3D-vientiliitännäisten ehkä suurin vajavaisuus ovat valot, koska niillä ei pysty luomaan varjoja ympäristöön. X3D-vientiliitännäiset vaativat vielä lisää kehitystä, jotta ympäristön kolmiulotteisuus näyttäisi realistiselta.

X3D:n leviäminen on hyvin paljon riippuvainen siitä, miten eri sovellusten ja teknikoiden kehittäjät näkevät sen suomat mahdollisuudet. X3D-tuen lisääminen esimerkiksi Internet-selaimeen ei vaatisi kovin suuria ponnistuksia ohjelmistovalmistajalta. Lisäksi X3D:n leviämistä varjostaa VRML-kielen nopea kuoleminen, koska se on VRML-kielen seuraaja.

Lisätutkimusta X3D-kielen käytöstä ja mahdollisuuksista visualisoinnin kannalta kannattaisi jatkaa vasta, kun 3ds Max -ohjelmaan julkaistaan toimiva ja paljon enemmän ominaisuuksia tukeva vientiliitännäinen. Muitakin kaupallisia vaihtoehtoja on kuin Autodesk 3ds Max, kuten Alias|Wavefront Maya -grafiikkaohjelma, jonka Autodesk osti tämän vuoden alussa ja muutti nimen Autodesk Maya:ksi. Maya:lle julkaistiin avoimeen lähdekoodiin perustuva X3D-vientiliitännäinen maaliskuussa 2006. Lisätietoa saa Internet-osoitteista <<http://rawkee.sourceforge.net/>> ja <<http://www.web3d.org/>>. Lisäksi on kokonaan avoimeen lähdekoodiin perustuva vaihtoehto Blender -grafiikkaohjelma, jolle on julkaistu avoimeen lähdekoodiin perustuva X3D-vienti- ja tuontiliitännäinen. Lisätietoa liitännäisistä saa Internet-osoitteista <<http://www.web3d.org/>>.

Autodesk Maya´n ja X3D-kielen yhteensopivuus ja ominaisuudet olisi sopivin seuraava tutkimuskohde, koska ohjelmaa käytetään laajasti kaupallisessa toiminnassa sekä korkeakoululla on muutama lisenssi kyseiseen ohjelmistoon.

LÄHTEET

Painetut lähteet

Ames, A., Nadeau, D. & Moreland, J. 1997. VRML 2.0 Sourcebook. 2. painos. John Wiley & Sons, Inc., Yhdysvallat.

Hintikka, K. 1993. Tieto – neljäs tuotantotekijä. Painatuskeskus, Helsinki.

Hintikka, K. 1994. Virtuaalinen tila – julkinen olohuone. Painatuskeskus, Helsinki.

Hintikka, K., Kojo, I. & Metsämäki, M. 1998. Virtuaaliympäristöjen suunnitteluopas. Edita, Helsinki.

Reitmaa, I., Vanhala, J., Kauttu, A. & Anttila, M. 1995. Virtuaaliympäristöt - kuvan sisälle vievät tekniikat. TEKES, ISSN 0782-5420 ; 45. Teknologian kehittämiskeskus, Paino-center, Helsinki.

Smith, D., Boyd, R. & Scott, A. 1996. VRML & 3D. Hayden Books. Suomen ATK-kustannus Oy, Gummerus Kirjapaino Oy, Jyväskylä.

Sähköiset lähteet

Ashdown, N. & Forestino, S. A Guide to VRML 2.0 and an Evaluation of VRML Modelling Tools [verkkodokumentti]. 1998 [viitattu 12.1.2006].
Saatavissa: <http://www.agocg.ac.uk/train/vrml2rep/cover.htm>

Autodesk. 2003a. 3ds max User Reference: VRML Export [viitattu 20.3.2006].
Saatavissa ohjelman asennuskansioista ...\\3dsmax6\\help\\3dsmax.chm.

Autodesk. 2003b. 3ds max User Reference: VRML Tips [viitattu 20.3.2006].
Saatavissa ohjelman asennuskansioista ...\\3dsmax6\\help\\3dsmax.chm.

Bell, G., Parisi, A. & Pesce, M. VRML 1.0 Specifications [verkkodokumentti]. 9.9.1995. [viitattu 24.1.2006].
Saatavissa:
<http://www.web3d.org/x3d/specifications/vrml/VRML1.0/index.html>

Kovaniemi, T. 2002. Virtuaalimaailmojen toteuttamiseen käytettävät vaihtoehdot: tarkastelussa VRML, OpenGL ja Java 3D. Pro Gradu -tutkielma. Tampereen yliopisto, tietojenkäsittelytieteiden laitos, kesäkuu 2002.
Saatavissa:
http://www.cs.uta.fi/research/theses/masters/Kovaniemi_Toni.pdf

Nadeau, D. VRML feature summary [verkkodokumentti]. Päivitetty 30.8.1997. [viitattu 12.1.2006].
Saataavissa: http://www.sdsc.edu/DOCT/Publications/a6/intro_vrml.htm

NIST. 2006. VRML plugin and browser detector. [verkkodokumentti]. 1998. Päivitetty 20.2.2006. [viitattu 21.2.2006].
Saataavissa: <http://cic.nist.gov/vrml/vbdetect.html>

Octaga. 2001. Octaga Exporter for 3ds max v.5, v.6 and v.7. [verkkodokumentti]. Päivitetty 17.1.2006. [viitattu 29.1.2006].
Saataavissa: <http://www.octag.com>

Tuikka, T., Kerttula, M., Salmela, M. & Paasovaara, H. 1999. Virtuaaliprototyypit elektroniikkatuotteiden kehityksessä [verkkodokumentti]. [viitattu 17.3.2006].
Saataavissa: <http://www.proessori.fi/es99/Virtual.htm>

Unreal – Realm of concepts. 2006. 3D Studio Max X3D Exporter [verkkodokumentti]. [viitattu 15.3.2006].
Saataavissa: <http://www.unrealroc.com/MaxExporter.html>

Web3d. 2006a. The Virtual Reality Modeling Language Introduction. [verkkodokumentti]. [viitattu 30.3.2006].
Saataavissa: <http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/>

Web3d. 2006b. Why use X3D over VRML 2.0? [verkkodokumentti]. Päivitetty tammikuu 2005. [viitattu 30.1.2006].
Saataavissa: http://www.web3d.org/x3d/x3d_vs_vrml.html

Web3d. 2006c. X3D International Specification Standards. [verkkodokumentti]. [viitattu 14.3.2006].
Saataavissa: http://www.web3d.org/x3d/specifications/x3d_specification.html

KUVALÄHTEET

Opinnäytetyössä esitettävät kuvat on luonut T. Luntta.

Kuva 1 Oikeakätinen koordinaatisto.....	- 14 -
Kuva 2 VRML-kielellä luotu laatikko-objekti.....	- 18 -
Kuva 3 VRML-kielellä luotu kartio-objekti.....	- 19 -
Kuva 4 VRML-kielellä luotu sylinteriobjekti.....	- 19 -
Kuva 5 VRML-kielellä luotu pallo-objekti.....	- 20 -
Kuva 6 VRML-kielellä luotu tekstiobjekti.....	- 20 -
Kuva 7 Octaga Exporter -vientiliitännäisen käyttöliittymä.....	- 29 -
Kuva 8 VRML97 Exporter -vientiliitännäisen käyttöliittymä.....	- 30 -
Kuva 9 Web3D X3D Exporter -vientiliitännäisen käyttöliittymä.....	- 31 -

Kuva 10 Octaga Professional -katseluohjelman käyttöliittymä.	- 32 -
Kuva 11 BS Contact -katseluohjelman käyttöliittymä.	- 33 -
Kuva 12 Octaga Exporter:lla viety 2D-kuva.....	- 38 -
Kuva 13 Octaga Exporter:lla viety bump mapping -ominaisuus.	- 39 -
Kuva 14 3ds Max:sta renderöity bump mapping -ominaisuus.	- 39 -
Kuva 15 Octaga Exporter:lla viety heijastuskartta, jossa aurinko ja pilviä.	- 40 -
Kuva 16 Octaga Exporter:lla viety multi-texturing -ominaisuus.....	- 41 -
Kuva 17 Web 3D:llä viety multi-texturing -ominaisuus.....	- 41 -
Kuva 18 Valon värin animointi.	- 42 -
Kuva 19 Materiaalin diffuse-värin animointi.	- 42 -
Kuva 20 Käänteiskinematikka-animaatio, jonka on toteutettu IK-nivelillä.	- 43 -
Kuva 21 Hierarkia-animaatio, joka on toteutettu animoidulla dummy-objektia.	- 43 -
Kuva 22 Pallon pintaverkkoa muokataan animoidulla noise-muokkaimella.....	- 43 -
Kuva 23 Octaga Exporter:lla viety spray-partikkelit.	- 44 -
Kuva 24 CV curve -käyrillä luotu NURBS-pintamalli.....	- 45 -

LIITTEET

LIITE 1

Taulukko VRML 1.0:n ja VRML97 ominaisuuksista.

LIITE 2

CD-ROM -levy

Levyn sisältö:

- abstract doc-tiedostona
- opinnäytetyö pdf-tiedostona
- tiivistelmä doc-tiedostona

LIITE 1

Taulukko 4 VRML 1.0 ja VRML97 ominaisuuksia.

Yleistä ominaisuuksista		
Ominaisuus	VRML 1.0	VRML 2.0/97
Nimi	VRML = Virtual Reality Modeling Language	VRML = Virtual Reality Modeling Language
Luoja	Internet-yhteisö	Internet-yhteisö
Omistaja	Internet-yhteisö	ISO
Julkaisupäivä	1995	1997
Tärkein renderöintijärjestelmä	Interaktiivinen	Interaktiivinen
Tärkeimmät sovel-lusalueet	ACAD, tiede, virtuaalito-dellisuus	ACAD, viihde, tiede, si-mulaatio, virtuaalitodelli-suus
Tärkeimmät sisäl-tötyypit	Ympäristöt	Ympäristöt
Ominaisuuksien yhteenveto	VRML 1.0 voi sisältää monia geometrisia muo-toja, ja niiden varjostuksia, tekstuureja ja muodon-muutoksien määritelmiä. Muodot voivat olla ryhmi-telty hierarkkisesti, nimetty ja niistä on mainittu esi-merkit. Valonlähteet voivat olla sijoitettu ympäristöön. Voi sisältää kameroita.	VRML 2.0/97 sisältää sa-mat asiat kuin VRML 1.0. Lisäksi voidaan lisätä taustoja ja sumua, navi-gaatio-ohjaimet, animaati-oita ja vuorovaikutusoh-jaimet. Menettelytapasi-sältö voidaan luoda käyt-tämällä Javaa ja JavaSc-riptiä. Kieltä voidaan laa-jentaa macroilla.
Geometria		
Ominaisuus	VRML 1.0	VRML 2.0/97
Geometriatyypit	<i>Avoim:</i> viiva, piste, moni-kulmio <i>Määrite:</i> laatikko, kartio, lieriö, pallo, teksti	<i>Avoim</i> viiva, piste, moni-kulmio <i>Määrite:</i> laatikko, kartio, lieriö, pallo, teksti, purso-tus ja pyörähdyspinta
Geometriakieli	ei	Java, JavaScript
Viivan leveys	Aina yhden pikselin levyi-nen	Aina yhden pikselin levyi-nen

Varjostus		
Ominaisuus	VRML 1.0	VRML 2.0/97
Varjostus	Kyllä	Kyllä
Värimaailma	Liukuluku RGB	Liukuluku RGB
Valaisumalliparametrit	Ympäröivä väri, diffuusiväri, säteilevä väri, spekulaariväri, kiiltokerroin, läpikuultavuuskerroin	Ympäröivä intensiteetti, diffuusiväri, säteilevä väri, spekulaariväri, kiiltokerroin, läpikuultavuuskerroin
Varjostusmallit	Jatkuva, sileä	Jatkuva, sileä
Varjostuskieli	Ei	Ei
Tekstuurit		
Ominaisuus	VRML 1.0	VRML 2.0/97
Tekstuurin saataavuus	Kyllä	Kyllä
Tekstuurityypit	2D bittikartta	2D bittikartta, 2D video
Tekstuurimuisti	upotettu, selainmääritellyt tiedostoformaatit	upotettu, GIF, JPEG, MPEG, PNG
Tekstuurien käyttötyypit	Yksi tekstuuri kerrallaan, diffuusinen tai ympäröivä värikartta yhdistettynä vaihtoehtoiseen läpikuultavuuskarttaan	Yksi tekstuuri kerrallaan, diffuusinen tai ympäröivä värikartta yhdistettynä vaihtoehtoiseen läpikuultavuuskarttaan
Tekstuurien muutos-tyypit	Kääntäminen, pyöriminen, asteikko, keskikohta, asteikon suuntaaminen	Kääntäminen, pyöriminen, asteikko, keskikohta, asteikon suuntaaminen
Tekstuurien käytön muutokset	Useat muutokset missä tahansa järjestyksessä	Yksittäiset muutokset muokatussa järjestyksessä
Viivan rakenne	Teksturoidut viivat	Aina umpinainen
Muodonmuutokset		
Ominaisuus	VRML 1.0	VRML 2.0/97
Muutosten saataavuus	Kyllä	Kyllä
Muutos-tyypit	Sattumanvarainen 4x4, sattumanvarainen pyörähdys, sattumanvarainen skaalaus, sattumanvarainen kääntö, keskikohta, XYZ leikkaus, XYZ ka- vennus	Sattumanvarainen 4x4, sattumanvarainen pyörähdys, sattumanvarainen skaalaus, sattumanvarainen kääntö, keskikohta

Ryhmittely		
Ominaisuus	VRML 1.0	VRML 2.0/97
Ryhmän saatavuus	Kyllä	Kyllä
Ryhmän hierarkia	Kyllä	Kyllä
Ryhmän nimi	Valinnainen	Valinnainen
Ryhmätyypit	Anchor, group, inline, level-of-detail, separator, switch, transform separator	Anchor, billboard, group, inline, level-of-detail, switch, transform
Instanssien tyypit	attribuutti, ryhmä, muoto	attribuutti, ryhmä, muoto
Valaistus		
Ominaisuus	VRML 1.0	VRML 2.0/97
Valaistuksen saatavuus	Kyllä	Kyllä
Valojen määrä	Rajoittamaton	Rajoittamaton
Aktiivisten valojen määrä	Rajoittamaton, mutta suositellaan että se on vähemmän kuin 8	Rajoittamaton, mutta suositellaan että se on vähemmän kuin 8
Valon värin tilavuus	Liukuluku RGB	Liukuluku RGB
Valotyytit	Suunnattu, piste, spotti	Suunnattu, piste, spotti
Valoparametrit	<i>Suunnattu:</i> väri, suunta, intensiteetti, päällä/pois <i>Piste:</i> väri intensiteetti, päällä/pois, sijainti <i>Spotti:</i> väri, kartion kulma, kartion terävyys, suunta, intensiteetti, päällä/pois, sijainti	<i>Suunnattu:</i> väri, suunta, intensiteetti, päällä/pois <i>Piste:</i> vaimennus, väri, intensiteetti, päällä/pois, sijainti <i>Spotti:</i> Vaimennus, väri, kartion kulma, kartion terävyys, intensiteetti, päällä/pois, sijainti
valotyyppien laajuus	ryhmä	Vaimennus (piste and spotti), ryhmä (suunnattava)
Äänet		
Ominaisuus	VRML 1.0	VRML 2.0/97
Äänien saatavuus	Ei	Kyllä
Äänien määrä	Ei	Rajoittamaton
Aktiivisten äänien määrä	Ei	Rajoittamaton, mutta yleensä vähemmän kuin 8
Äänityypit	Ei	Piste

Ääniparametrit	Ei	Intensiteetti, päällä/pois, sijainti, alue
Äänen vaikutusalue tyypit	Ei	Vaimennus, alue
Äänien pakkaaminen	Ei	MIDI, MPEG, WAV
Äänien prosessointi	Ei	Sävelkorkeus, spatialinen
Taustat		
Ominaisuus	VRML 1.0	VRML 2.0/97
Taustojen saatavuus	Kyllä laajennuksen kautta	Kyllä
Taustojen määrä	1	Rajoittamaton
Aktiivisten taustojen määrä	1	1
Taustatyypit	Ei	Liukuvärjäys, panorama-kuvat
Taustojen pakkaaminen	Ei	GIF, JPEG, PNG
Sumu		
Ominaisuus	VRML 1.0	VRML 2.0/97
Sumun saatavuus	Ei	Kyllä
Sumujen määrä	Ei	Rajoittamaton
Aktiivisten sumujen määrä	Ei	1
Sumutyypit	Ei	Lineaarinen, eksponentiaalinen
Sumuparametrit	Ei	Väri, tiheys, päällä/pois
Sumun laajuustyyppit	Ei	Globaali
Kamerat		
Ominaisuus	VRML 1.0	VRML 2.0/97
Kameran saatavuus	Kyllä	Kyllä
Kameroiden määrä	Laajennuksen kautta rajoittamaton	Rajoittamaton
Aktiivisten kameroiden määrä	1	1
Kameratyypit	ortogonaalinen, perspektiivi	perspektiivi
Kameraparametrit	Katselunäkymä, orientatio, sijainti	Katselunäkymä, orientatio, sijainti
Navigaatio		
Ominaisuus	VRML 1.0	VRML 2.0/97

Navigaation saata- vuus	Kyllä, laajennuksen kautta	Kyllä
Navigaation rajoit- ukset	Ei mitään	törmäystarkastelu, valin- nainen painovoima
Navigaatiomoodit	Tutkiminen, lentäminen, käveleminen	Tutkiminen, lentäminen, käveleminen
Animaatio		
Ominaisuus	VRML 1.0	VRML 2.0/97
Animaation saata- vuus	Ei	Kyllä
Animaatiotyypit	Ei	Avainkehukset, menettely- tapa
Animaatiokielet	Ei	Animaation kiertokulku, Java, JavaScript
Aikaresoluutio	Ei	Vaihteleva
Ajan määrittely	Ei	Normalisoitu, reaaliaikai- nen
Vuorovaikutus		
Ominaisuus	VRML 1.0	VRML 2.0/97
Vuorovaikutusmah- dollisuus	Kyllä	Kyllä
Syöttötyypit	<i>Syöttölaite:</i> ei mitään <i>Käyttöliittymät:</i> painikkeet (ankkurit) <i>Sensorit:</i> ei mitään	<i>Syöttölaite:</i> ei mitään <i>Käyttöliittymä:</i> painikkeet, relative locators, valuators <i>Sensorit:</i> billboard, tör- mäystarkastelu, LOD, etäisyys tarkastelu, aika, näkyvyys
Vastaustyyppit	Ajastus (ankkurit)	Ajastus, menettelytapa
Vuorovaikutuskieli	Ei mitään	Animaatiokierros, Java, JavaScript
Laajennus		
Ominaisuus	VRML 1.0	VRML 2.0/97
Laajennusmahdolli- suus	Ei	Kyllä
Laajennustyyppit	Ei	Makro (PROTO)

(Nadeau 1997.)

LIITE 2