

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Magistro darbas

**3D objektų eksportavimas į M3G formatą, naudojamą 3D grafikai
mobiliuosiuose telefonuose**

Atliko: 2 kurso, 10 grupės studentė

Julija Pragarauskaitė (parašas)

Darbo vadovė:

dr. Margarita Kazakevičiūtė (parašas)

Darbo recenzentas:

doc. Rimvydas Krasauskas
(parašas)

Vilnius
2008

Turinys

Santrumpų sąrašas	3
Anotacija.....	4
Summary.....	5
1. Įvadas.....	6
2. Darbo tikslai	7
3. J2ME mobiliuosiuose telefonuose.....	7
3.1 Konfigūracijos CDC ir CLDC.....	8
3.2 Profiliai	9
3.3 MIDP	9
3.4 Pasirenkamieji paketai.....	10
4. M3G standartas.....	11
4.1 M3G suderinamumas.....	11
4.2 M3G realizacijos būdai.....	12
4.3 M3G 1.0 klasių modelis.....	12
4.4 M3G 1.0 failo formatas	13
4.5 Objektų M3G failo saugojimo tvarka.....	14
4.6 M3G standarto duomenų tipai	15
4.7 M3G 2.0 standarto apžvalga.....	17
5. Modeliavimo priemonių apžvalga.....	18
6. Egzistuojančių M3G eksportavimo priemonių apžvalga.....	19
7. Python programavimo kalba.....	20
8. Eksportavimo priemonės programavimas	22
8.1 Eksportavimo schema.....	22
8.2 3D modelių kūrimas	23
8.3.1 Python skripto naudojimas Blender.....	24
8.3.2 Python skripto programavimas.....	25
8.3.4 Python skripto vykdymas	28
8.4 3D scenos duomenų saugojimas XML formatu	29
8.5 Eksportavimo priemonės kūrimas Java programavimo kalba.....	32
8.5.1 Projekto struktūra	32
8.5.2 Aplikacijos vartotojo sąsaja.....	33
8.5.3 Aplikacijos valdymas	33
8.5.4 Eksportavimo metu atliekami veiksmai	34
8.5.5 Eksportavimo informacijos bei klaidų pranešimai	36
9. Eksportavimo rezultatų palyginimas	37
9.1 Eksportavimo priemonių pasirinkimas.....	37
9.2 Eksportavimo rezultatų palyginimo kriterijai.....	39
9.3 Eksportavimo rezultatų palyginimas	39
10. Magistro darbo išvados.....	48
Literatūros sąrašas	49
Priedai.....	51

Santrumpų sąrašas

- ✓ **3D** – trimatė erdvė.
- ✓ **3DS Max** – 3D Studio Max (programinis paketas 3D objektų kūrimui bei jų animavimui).
- ✓ **J2ME** - Java 2 Micro Edition.
- ✓ **J2SE** – Java 2 Standart Edition.
- ✓ **M3G** - Mobile 3D Graphics.
- ✓ **API** - Application Program Interface.
- ✓ **MIDP** - Mobile Information Device Profile.
- ✓ **JSR** - Java Specification Request.
- ✓ **CLDC** - Connected, Limited Device Configuration.
- ✓ **CDC** - Connected Device Configuration.
- ✓ **IDE** - Integrated Development Environment.
- ✓ **PBP** - Personal Basis Profile.
- ✓ **PDA** – Personal Digital Assistent.
- ✓ **PP** - Personal Profile.
- ✓ **OpenGL ES** - OpenGL for Embedded System.
- ✓ **JAD** – Java Application Description file.
- ✓ **JAR** – Java Archyve.
- ✓ **XML** – Extensible Markup Language.
- ✓ **FP** – Foundation Profile.

Anotacija

Pastaruoju metu trimatės aplikacijos mobiliems telefonams dažnai kuriamos J2ME programavimo kalba bei M3G standarto Java klasėmis.

Šiame darbe, analizuojamos M3G standarto 3D grafikai galimybės mobiliuosiuose telefonuose, suderinamumas bei pats M3G failo formatas. Nagrinėjamos eksportavimo galimybės iš paplitusių 3D modeliavimo priemonių į M3G formatą. Pasirinkta 3D modeliavimo priemonė Blender, kuriai sukurta eksportavimo į M3G formatą schema bei ji realizuojama suprogramavus eksportavimo priemonę su vartotojo sąsaja vartojant Python ir Java programavimo kalbas.

Remiantis šia aplikacija, pasirinktos kitos eksportavimo į M3G formatą priemonės, palyginti eksportavimo rezultatai bei pateiktos išvados.

Summary

In the master thesis the M3G standard for 3D graphics in mobile phones, its compatibility and possibilities of exporting 3D models to M3G format are investigated. Most popular 3D modelling and exporting to M3G format tools are analyzed. Blender was selected as main master thesis modelling tool for possibility to reach 3D model data using Python scripts. An exporting scheme from Blender to M3G format was created and realized using Python and Java programming languages.

The exporter works as a plug-in for the Blender modelling tool. It can be accessed in the main Blender menu. The Python programming language in the exporter was used for extracting data from 3D scene and saving it in XML format. The Java programming language was used for reading 3D data from XML file, making hierarchical 3D elements tree, creating 3D elements tree, constructing data arrays and sectors, where 3D data arrays and sectors are kept and exporting data to M3G file. The exporter can load an exported M3G file to mobile application and show it in Java emulator – mobile phone.

On the basis of several 3D models, the created exporter was compared to other exporters using the quality of performance, reliability and other criteria.

1. Įvadas

Mobiliųjų telekomunikacijų rinka sparčiai auga, didėja mobiliųjų telefonų įvairovė bei jų funkcinės galimybės. Šiuolaikiniai telefonai ne tik turi integruotas fotokameras, radiją, MP3 grotuvą, diktofoną ir kitas įrenginius, bet netgi savo operacinę sistemą, 3G ryšį, Java aplikacijas ir t.t. Prasidėjus naujai trečiosios kartos mobiliojo ryšio erai, didžiausia mobiliųjų telefonų gamintoja „Nokia“ net pasiūlė mobilųjį telefoną vadinti mobiliuoju daugiafunkciniu įrenginiu, kadangi skambinimo funkcija yra tik viena iš daugybės mobiliojo įrenginio galimybių.

Anksčiau mobiliojo telefono programinę įrangą sudarė tam tikras kiekis aplikacijų pritaikytų konkrečiai operacinei sistemai. Atsiradęs Java standartas, skirtas mobiliesiems telefonams, suteikė galimybę programuotojams kurti naujas aplikacijas. Dabar „protingųjų“ telefonų eroje vartotojui siūloma daugybė Java žaidimų bei aplikacijų. Java 2 Micro Edition (J2ME) standartas yra plačiausiai palaikomas visų mobiliųjų įrenginių gamintojų ir diegiamas beveik visuose naujai išleidžiamuose gaminiuose.

Per pastaruosius keletą metų sparčiausiai besivystančios mobiliųjų telefonų programinės įrangos sritys buvo žaidimai bei 3D grafika. „Nokios“ vadovaujama ekspertų grupė, 2004 m. pasiūlė standartą trimatei grafikai mobiliuosiuose telefonuose – JSR-184, dar vadinamą M3G 1.0. Šiame standarte yra realizuoti paprastieji objektai (Mesh), transformacijos, apšvietimai, kameros, animacijos. M3G standartas vartojamas žaidimų kūrimui, žemėlapių atvaizdavimui, animuotų žinučių kūrimui, ekrano užsklandoms ir t.t. M3G siūlo dviejų tipų realizavimą: netiesioginis būdas (retained mode), kai programuotojas pasinaudoja scenos grafo hierarchiniu medžiu, bei tiesioginis būdas (immediate mode), kai objektai yra realizuojami Java kodu.

Šiuo metu yra išleistas M3G 1.0 standarto tęsinys – naujas patobulintas standartas trimatei grafikai mobiliuosiuose telefonuose – JSR-297 (arba M3G 2.0), kuris pagerins 3D aplikacijų veikimą bei jų užimamą vietą mobiliuosiuose telefonuose. Šiuo metu viešai prieinamas M3G 2.0 standartas yra tik ankstyvoji versija (Early Draft Review version). Nors JSR ekspertų grupė planavo išleisti galutinę standarto versiją 2008 metų pirmajame ketvirtyje, tačiau tai jiems padaryti nepavyko ir ji vis dar yra tobulinama.

Nors M3G standartas plačiai taikomas mobiliuosiuose telefonuose, tačiau egzistuoja nedaug patikimų eksportavimo priemonių, kurios leistų eksportuoti 3D modelį iš populiarių 3D modeliavimo paketų kaip 3D Studio Max, Blender, Maya ir t.t. į M3G formatą, naudojamą 3D grafikai mobiliuosiuose telefonuose. Šių eksportavimo priemonių galimybės vis dar ribotos, o

eksportuojant net palyginti nesudėtingus trimačius objektus, eksportavimo priemonės dažnai „pakimba“ taip ir nepranešusios, kurioje vietoje yra klaida (žr. [EXP01]). Ši problema labai apsunkina sudėtingesnių 3D modelių eksportavimą į M3G formatą.

Magistro darbo tikslas – ištirti eksportavimo į M3G formatą galimybes ir realizuoti eksportavimo priemonę, kuri galėtų eksportuoti 3D modelį į M3G formatą kartu su kameromis, šviesomis bei palyginti eksportavimo rezultatus su paplitusiomis eksportavimo priemonėmis.

2. Darbo tikslai

Šio darbo tikslai:

- Išnagrinėti M3G standartą, skirtą 3D grafikai mobiliuosiuose telefonuose.
- Ištirti bei palyginti 3D scenos duomenų pasiekimo modeliavimo priemonėse bei eksportavimo į M3G standartą galimybes.
- Suprogramuoti Python skriptą, kuris pasiektų 3D modelio duomenis Blender modeliavimo priemonėje ir juos eksportuotų į XML formatą.
- Sukurti eksportavimo priemonę Java programavimo kalba, kuri atkurtų 3D scenos grafą iš XML failo ir eksportuotų duomenis į M3G formatą.
- Išbandyti eksportavimo priemonę su paprastais ir sudėtingesniais 3D modeliais bei užtikrinti eksportavimo priemonės funkcionalumą.
- Sukurti testavimui skirtą Java aplikaciją, kuri nuskaitytų eksportuotą M3G failą ir jį įkrautų Java emuliacijoje – mobiliajame telefone.
- Palyginti kitų egzistuojančių eksportavimo priemonių galimybes bei rezultatus su magistro darbe sukurtos eksportavimo priemonės rezultatais, bei pateikti išvadas.

3. J2ME mobiliuosiuose telefonuose

J2ME yra Java platforma įrenginiams, kurie turi mažai atminties ir procesoriaus galios palyginus su asmeniniais kompiuteriais, pavyzdžiui mobilieji telefonai, delniniai kompiuteriai ir t.t. J2ME buvo sukurta 1999 m. ir tapo standartu mobiliuosiuose telefonuose. Garsūs telefonų gamintojai „Nokia“, „Samsung“, „BenQ - Siemens“, „Sony - Ericsson“ ir kiti parduoda telefonus su integruota J2ME. J2ME platforma yra tam tikras J2SE poaibis.

J2ME apibūdina (žr. [JME]):

- Konfigūracijos
- Profiliai
- Pasirenkamieji paketai

3.1 Konfigūracijos CDC ir CLDC

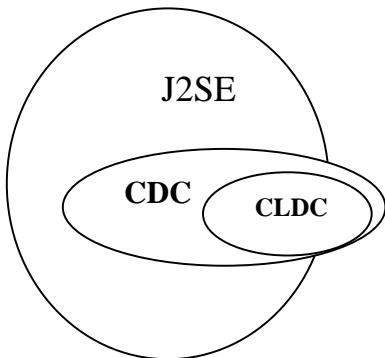
Konfigūracija apibrėžia Java kalbą, virtualią mašiną ir bibliotekos klases. Kad galėtų naudoti konfigūraciją, J2ME turintis įrenginys privalo tenkinti minimalius reikalavimus, kurie apibrėžti konfigūracijos specifikacijoje. Yra dvi J2ME konfigūracijos (žr. [Knu03]):

- *Connected Limited Device Configuration (CLDC)*
- *Connected Device Configuration (CDC)*

CLDC konfigūracija skirta įrenginiams, turintiems mažai atminties ir/arba lėtą procesorių. Šiuo metu vartojama CLDC 1.1. Šios konfigūracijos specifikacija:

- bent 192 kilobaitų bendrosios prieinamos atminties Java platformai
- turi 16- ar 32-bitų procesorių
- vartoja mažai procesoriaus galios
- turi tam tikrą prieigą prie tinklo

CDC konfigūracija susideda iš pilnos Java virtualios mašinos ir daug daugiau pagrindinių klasių nei CLDC, todėl ši konfigūracija reikalauja daugiau atminties ir greitesnio procesoriaus. CDC konfigūracija skirta vidutiniams įrenginiams, kurių atmintis ir procesorius yra per daug riboti, kad palaikytų J2SE ir yra galingesni nei tie, kuriuose naudojama CLDC konfigūracija. Ryšį tarp J2ME konfigūracijų ir J2SE atvaizduoja schema:



Pav. 1. Ryšys tarp J2SE bei CLDC ir CDC konfigūracijų.

3.2 Profiliai

Profiliai praplečia konfigūracijos funkcionalumą papildydami ją naujomis klasėmis. Kad būtų galima naudoti profilį, įrenginys turi tenkinti minimalius konfigūracijos reikalavimus ir taip pat visus papildomus profilio reikalavimus, kurie nurodyti profilio specifikacijoje.

Skirtingi profiliai vartojami skirtingų pajėgumų įrenginiams bei konfigūracijoms ir neveikia su kitomis konfigūracijomis. MIDP (*Mobile Information Device Profile*) yra su CLDC konfigūracija suderintas profilis, skirtas paleisti aplikacijoms mobiliuosiuose telefonuose su mažais ekranais, be laido HTTP sąsaja ir turintiems mažai atminties. Kitas CLDC konfigūracija paremtas profilis yra PDAP (*Personal Digital Assistant Profile*), kuris praplečia funkcionalumą papildomomis klasėmis ir savybėmis galingesniems įrenginiams.

Su CLDC konfigūracija suderinti profiliai yra: FP (*Foundation Profile*), kuris praplečia CLDC konfigūraciją papildomomis J2SE klasėmis; PBP (*Personal Basis Profile*) praplečia FP paprastosiomis (iš AWT išvestomis) vartotojo sąsajos klasėmis ir PP (*Personal Profile*), kuris praplečia PBP įskiepio palaikymu bei sudėtingomis vartotojo sąsajos klasėmis ir t.t.

3.3 MIDP

MIDP yra vienas iš daugelio CLDC konfigūracija paremtų profilių. Kad MIDP galėtų teikti papildomą funkcionalumą CLDC konfigūracijai, reikia, kad įrenginys ne tik tenkintų konfigūracijos reikalavimus, bet ir MIDP reikalavimus, pvz. atmintis Java bibliotekoms.

Dabartinė MIDP versija yra 2.0. MIDP 2.0 suteikia galimybę aplikacijai turėti tris būsenas: aktyvią, sustabdytą ir sunaikintą, bei leidžia keisti vieną būseną kita. MIDP 2.0 taip pat pristato aplikacijų modelį, vadinamą MIDlets. MIDP 2.0 versijoje MIDlet yra bazinis vykdymo vienetas. MIDlets gali būti įpakuojami į Java Archive (JAR) failą, kuris bus paleidžiamas Java turinčiame mobiliajame telefone. Taip pat gali būti naudojamas kitas formatas Java Application Descriptor (JAD), kuris naudojamas valdyti aplikacijai ar papildyti JAR failą.

Kiekviena MIDlet aplikacija susideda iš tokių svarbiausių dalių (žr. [Lam04]):

```
public class ManoAplikacija extends MIDlet {
    // konstruktorius, jame inicializuojami parametrai
    public MyApplication() {...}

    // metodas kviečiamas, kai aplikacija yra paleidžiama ar grįžta iš pauzės
    protected void startApp() {...}

    // metodas kviečiamas, kai aplikacija sustabdoma (pauzė)
```

```

protected void pauseApp() {...}

// metodas kviečiamas, kai norima aplikaciją pabaigti
protected void destroyApp(boolean force) {...}
}

```

Klasėje, praplečiančioje MIDlet, turi būti realizuoti aplikacijos gyvavimo ciklo metodai: *startApp*, *pauseApp* ir *stopApp*. Kai aplikacija paleidžiama mobiliajame telefone, pirmiausia iškviečiamas jos konstruktorius kaip ir įprastai Java klasei. Grįžus iš konstruktoriaus, kviečiamas metodas *startApp*. Šis metodas taip pat kviečiamas visada, kai MIDlet grįžta iš pauzės būsenos. *PauseApp* metodas kviečiamas, kai aplikaciją reikia sustabdyti, pavyzdžiui, gavus trumpąjį pranešimą ar esant įeinančiam skambučiui. *PauseApp* metodo metu MIDlet turėtų atlaisvinti kuo daugiau išteklių, kiek tik galima.

DestroyApp metodas yra kviečiamas, kai MIDlet turėtų sustabdyti vykdymą ir atlaisvinti išteklius.

3.4 Pasirenkamieji paketai

Pasirenkamieji paketai yra aplikacijos sąsajų (API) rinkiniai, palaikantys papildomas bendras elgsenas, kurios nepriklauso vienai konkrečiai konfigūracijai ar profiliui, pvz., Bluetooth palaikymas apibrėžiamas kaip pasirenkamas paketas.

Pasirenkamieji paketai, taip pat kaip ir konfigūracijos ar profiliai, turi savo minimalius reikalavimus. Pasirenkamieji paketai taip pat priklauso nuo konkrečios konfigūracijos ir/arba vieno ar daugiau profilių.

Kuriama daug pasirenkamųjų paketų, pvz. Java APIs for Bluetooth, kuris suteikia Bluetooth palaikymą įrenginiams su CLDC konfigūracija, Wireless Messaging API, kuris leidžia siųsti ir gauti pranešimus naudojant Short Message Service ir t.t.

4. M3G standartas

Kadangi mobilieji telefonai yra ribotos atminties ir procesoriaus galios, todėl Java 3D jiems netinka, nes užima daug atminties ir yra nesuderinama su MIDP profiliu. Jei 3D modelių kūrimui naudojama OpenGL ES, tai ji reikalauja daug kodo net ir paprastiems dalykams, o tokie MIDlet užima daug vietos ir lėtai veikia. Atsiradus pakankamai priešasčių naujo standarto kūrimui, „Nokios“ vadovaujama JSR ekspertų grupė iš 26 narių („Sun Microsystems“, „Sony Ericsson“, „Symbian“ ir kt.) pasiūlė naują API mobiliesiems įrenginiams, kuris būtų interaktyvus ir galėtų veikti kaip J2ME pasirenkamasis paketas. Šis pasirenkamasis paketas - M3G gali teikti 3D grafiką:

- Įrenginiams, turintiems mažai procesoriaus galios ir atminties bei neturintiems jokios techninės atramos skaičiavimams slankiojo kablelio sistemoje.
- Galingesniems įrenginiams, pavyzdžiui PDA, su didele rezoliucija ir spalvotais ekranais, galingais procesoriais, turintiems daugiau atminties ir techninę atramą skaičiavimams slankiojo kablelio sistemoje, taip pat kaip ir 3D grafikos techninę įrangą.

Šis standartas yra lankstus, t.y. neribojantis aplikacijų tipo, bei leidžiantis gana greitai ir nesunkiai kurti 3D aplikacijas, kurios turėtų tam tikrą savo failo formatą M3G, iš kurio efektyviai pakraunamas 3D scenos grafas.

M3G standartas grindžiamas tomis pačiomis idėjomis kaip ir Java 3D: mažai, scenos grafas, turi naują failų formatą M3G, animacijas su raktiniais kadrais, jis taip pat liko suderinamas su OpenGL ES.

4.1 M3G suderinamumas

M3G veikia J2ME platformoje su CLDC konfigūracija ir MIDP profiliu. Kadangi M3G skaičiuoja slankiojo kablelio sistemoje, jis turi būti vartojamas tik su konfigūracijomis pradedant nuo CLDC 1.1 versijos, nes ši versija jau palaiko standartinius Java viengubo tikslumo slankaus kablelio (*float*, 32 bitai) ir dvigubo tikslumo slankaus kablelio (*double*, 64 bitai) tipus. Pav. 2 vaizduoja M3G standarto suderinamumą.

JSR-184	Pasirenkamasis paketas
MIDP	Profilis
CLDC 1.1	Konfigūracija

Pav. 2. M3G suderinamumas.

4.2 M3G realizacijos būdai

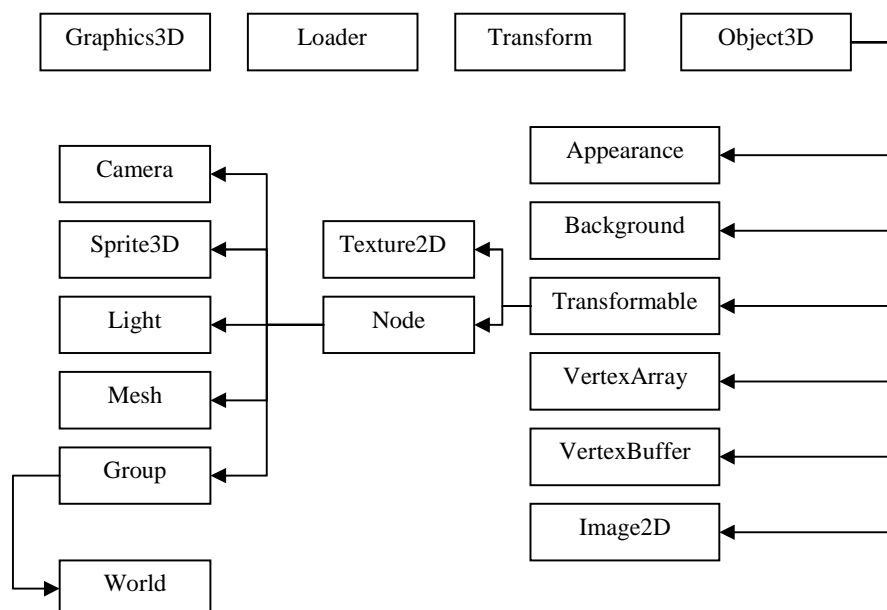
M3G standartas gali būti realizuojamas dviem būdais:

- Tiesioginis būdas (*immediate mode*) (žr. [IMB01]). Modeliavimu ir galimybėmis labai panašus į OpenGL ES. Pirmiausia nustatomos kameros, šviesos, tada programuojami geometriniai objektai; naudojami trikampiai kaip primityvai. Toks būdas suteikia programuotojui pilną 3D scenos kontrolę.
- Netiesioginis būdas (*retained mode*) (žr. [IMB02]). Šis būdas naudoja scenos grafa, kuriame saugoma scenos objektų informacija. Taip pat iš 3D pasaulio imamos šviesos ir kameros. Netiesioginis M3G realizavimas sutaupo daugiau laiko nei tiesioginis.

Šiame darbe realizuota eksportavimo priemonė bus skirta netiesioginiu būdu kuriamam 3D modeliui, kuriame naudojami scenos grafai yra saugomi M3G formatu. Scenos grafas yra efektyvios struktūros medis, skirtas saugoti 3D scenos duomenis. Tai hierarchinis būdas objektams išdėstyti 3D scenoje. Ši duomenų saugojimo struktūra yra medžio formos ir suteikia programuotojui aukšto lygio duomenų organizavimo galimybę.

4.3 M3G 1.0 klasių modelis

M3G standartas susideda iš 30 klasių, kurios suteikia programuotojui didesnę 3D funkcijų pasirinkimą bei sumažina MIDlet kodo kiekį. Apsiribosime tik tomis M3G klasėmis, kurios yra vartojamos kuriamoje aplikacijoje (žr. [JSR03] ir [MAPI]):



Sch. 1. Aplikacijoje panaudotų M3G klasių modelis

Kaip matyti iš pirmosios schemos, *Object3D* yra daugelio M3G klasių tėvinė klasė. Kaip parašyta M3G specifikacijoje „Object3D yra abstrakti pagrindinė klasė, skirta visiems objektams, kurie yra dalis 3D pasaulio“. Svarbus *Object3D* klasės atributas yra ID. Tai sveikas skaičius, kurį galima priskirti objektams scenoje ir vėliau juos surasti scenos grafe. Ši klasė taip pat vartojama kurti animacijoms scenoje.

Graphics3D apibrėžia scenos objektų vaizdavimą. Ši klasė yra vienobjektė, t.y. tik vienas objektas gali būti sukurtas klasei. Kad atvaizduotų 3D sceną, būtina susieti *Graphics3D* objektą su kuriu objektu bei nepamiršti jo atlaisvinti metodu *releaseTarget*. 3D modelio objektų vaizdavimas vyksta iškvietus *render* metodą (žr. [GR3D]):

```
void paint(Graphics g) {  
    myGraphics3D.bindTarget(g);  
    myGraphics3D.render(world);  
    myGraphics3D.releaseTarget();  
}
```

Transform klasėje visi metodai vartoja 4x4 matricas, kurios apibrėžia transformacijas. Ši klasė naudojama norint pasukti, pastumti, pakeisti mastelį ar kitaip modifikuoti objektą scenoje.

Loader klasė vartojama scenos grafo mazgų arba viso M3G failo įkėlimui į aplikaciją.

Appearance klasė apibrėžia atributus *Mesh* ar *Sprite3D* objektų atvaizdavimui. *Background* klasė skirta fono scenoje nustatymui. *Transformable* yra abstrakti klasė *Node* ir *Texture2D* klasėms, kuri apibrėžia metodus mazgų ir tekstūrų transformacijoms. *Sprite3D* klasė skirta atvaizduoti dvimačius paveikslus, esančius 3D erdvėje. *Sprite3D* objektai yra atvaizduojami visada nukreipti į ekraną, t.y. lygiuojami kartu su ekranu. *Mesh* klasė skirta atvaizduoti 3D objektams, kurie apibrėžti kaip daugiakampis paviršius. *Group* klasė saugo sugrupuotus scenos grafo mazgus. *World* klasė yra *Group* klasės plėtinys, ji skirta saugoti scenos grafo mazgams.

4.4 M3G 1.0 failo formatas

M3G failo formatas buvo sukurtas, kad galėtų kompaktiškai saugoti scenos grafo informaciją dvejetainiu formatu (žr. [JSR01]).

M3G failas susideda iš failo identifikatoriaus bei kitos informacijos, suskirstytos į sektorius:

	Failo identifikatorius
1 sektorius	Failo antraštė
2 sektorius	Nuorodos į išorinius objektus
3 sektorius	Scenos objektai
.....	...
n-tasis sektorius	Scenos objektai

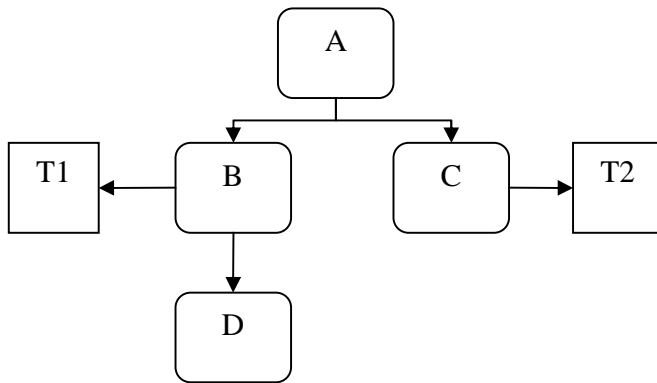
Failo identifikatorius yra unikali 12 baitų seka, pagal kurią atskiriama, kad tai M3G failo formatas. Failo antraštė nurodo, ar objektai yra spaudžiami ar ne, kiek yra objektų bei *checksum* reikšmę. *Checksum* reikšmė yra 32 bitų skaičius, pagal kurį galima patikrinti, ar sektorius buvo sėkmingai įkrautas. Sektorius yra vieno ar daugiau objektų duomenų saugykla, kuri saugoma baitų masyve. Į skirtingus sektorius objektai skirstomi tam, kad vėliau galėtų būti spaudžiami skirtingu suspaudimo laipsniu, pvz., *Mesh* objektai gali būti spaudžiami aukštu laipsniu, kad užimtų mažiau vietos, kiti objektai spaudžiami žemesniu laipsniu priklausomai nuo jų tipo, o antraštės informacija iš viso negali būti spaudžiama. Antrasis sektorius, kuriame saugomos nuorodos į išorinius failus gali būti kompresuojamas arba ne. Nuorodos į išorinius failus saugomos kaip UTF-8 koduotės *String* tipo kintamasis.

4.5 Objektų M3G failo saugojimo tvarka

Sektoriaus objektų duomenys pirmiausia yra dekompresuojami, tada interpretuojami kaip objektų seka. Kiekvienas failo objektas atvaizduoja vieną scenos grafo mazgą. Kad būtų galima dekompresuoti objektą, saugojamas jo tipas, baitų masyvo ilgis bei pats baitų masyvas.

Visi objektai M3G faile yra saugojami pagal tam tikrą tvarką: pirmiausia saugojami scenos grafo lapai, tuomet pagal nuorodas jų tėviniai mazgai, taigi scenos grafo hierarchijos viršūnė bus išsaugota vėliausiai. Faile negalimi ciklai (taip pat kaip ir scenos grafe).

Pavyzdžiui, scenos grafe yra objektai A, B, C, D ir tekstūros T1, T2:



Tuomet failo struktūra gali būti tokia:

		Failo identifikatorius
1 sektorius	Nekompresuojama	Failo antraštė
2 sektorius	Nekompresuojama	T1 T2
3 sektorius	Kompresuojama	D B C A arba D C B A

Kai kuriose M3G failo vietose gali būti klaidų, todėl šiame formate yra numatytas klaidų aptikimas, pvz. neteisingas objekto tipas, klaida išorinio failo nuorodoje, per daug arba per mažai objekto duomenų ir t.t.

4.6 M3G standarto duomenų tipai

M3G standartas turi dviejų rūšių duomenų tipus: pagrindinius bei sudėtinius. Pagrindiniai duomenų tipai skirti aprašyti bet kuriems 3D objekto duomenims.

Pagrindiniais duomenų tipais laikomi šie tipai:

- Byte - vienas baitas, galintis išsaugoti teigiamą skaičių, pvz. gali saugoti suspaudimo tipą (1 – jei duomenys spaudžiami, 0 – jei duomenys nespaudžiami), objektų skaičių sektoriuje, objekto tipą (specifikacijoje nurodyti tipai nuo 0 iki 255).
- UInt16 – dviejų baitų tipas, skirtas saugoti teigiamus sveikuosius skaičius, pvz. viršūnių numerius.
- Int16 – dviejų baitų tipas, skirtas saugoti sveikuosius skaičius su ženklu.
- UInt32 - keturių baitų tipas, skirtas saugoti teigiamus sveikuosius skaičius, pvz. viso failo svoris antraštėje, checksum reiškmė, kuri naudojama patikrinti, ar objektų sekcija buvo įkrauta sėkmingai.

- Int32 - keturių baitų tipas, skirtas saugoti sveikuosius skaičius su ženklu.
- Float32 – viengubo tikslumo slankaus kablelio tipas saugomas 4 baitų formatu, pvz. objekto išvaizdos švytėjimo informacija, kameros ir šviesos parametrai.
- String – Unicode sistemos simbolių seka, UTF-8 koduotės, pvz. autorius antraštėje, vieta, kurioje yra tekstūra.
- Boolean – duomenų tipas saugomas viename baite. Jo reikšmė gali būti 0 (netiesa), ar 1 (tiesa). Kitos reikšmės neleistinos ir turi būti traktuojamos kaip klaidos. Boolean duomenų tipas gali būti naudojamas antraštėje nurodant ar M3G failas turi nuorodą į išorines tekstūras.

Sudėtiniai duomenų tipai skirti išvengti pakartotino pagrindinių duomenų tipo naudojimo aprašant 3D objektus. Sudėtiniais duomenų tipais laikomi šie tipai:

- Vector3D – 3D vektoriui skirtas saugoti duomenų tipas. Jis aprašomas 3 pagrindinių tipų kompozicija: Float32 x, Float32 y, Float32 z. Vector3D duomenų tipu galima aprašyti transformacijas 3D erdvėje – postūmį, mastelio keitimą pagal X, Y, Z ašis.
- Matrix - 16 elementų dydžio matrica, kuria galima aprašyti bendrą objekto transformaciją erdvėje. Matricos elementai yra saugomi Float32 tipo masyve tokia tvarka:


```

0  1  2  3
4  5  6  7
8  9  10 11
12 13 14 15

```
- ColorRGB – spalvos be permatomumo informacijai skirtas saugoti duomenų tipas. Jis aprašomas 3 pagrindinių tipų kompozicija: Byte red, Byte green, Byte blue.
- ColorRGBA- spalvos su permatomumu informacijai skirtas saugoti duomenų tipas. Jis aprašomas 4 pagrindinių tipų kompozicija: Byte red, Byte green, Byte blue, Byte alpha. Šis duomenų tipas naudojamas fono spalvos informacijai išsaugoti.
- ObjectIndex – objekto eilės numeris M3G faile. Reikšmė 0 yra rezervuota pažymėti tuščią reikšmę; scenos objektų eilės numeris prasideda nuo 1. Šis sudėtinis duomenų tipas yra saugomas Uint32 tipo formatu.
- Type[] – kintamo dydžio masyvas, kuris saugo bet kokio tipo elementus. Count kintamasis saugo elementų masyve skaičių. Kiekvienas elementas gali būti pasiektas naudojant indeksą nuo 0 iki (count - 1). Šio duomenų tipo struktūra:

```
UInt32 count;
```


Type masyvas[0];

Type masyvas[1]; ir t.t.

- *Type*[count] – fiksuoto dydžio masyvas, kuris saugo nurodyto dydžio masyvo elementus.

Visi elementai yra surūšiuoti pagal indekso reikšmę. Šio duomenų tipo struktūra:

Type arrayValue[0];

Type arrayValue[1]; ir t.t.

4.7 M3G 2.0 standarto apžvalga

M3G 2.0 standarto ankstyvoji versija (Early Draft Review version) jau prieinama kiekvienam vartotojui. Ši versija buvo išleista 2007 m. liepos mėnesį ir iki šiol yra testuojama ir tobulinama.

Pasiūlytas M3G 2.0 standartas yra tęsinys ankstesniojo M3G 1.0 standarto. M3G 2.0 patobulins ankstesniąją versiją: pagreitins 3D grafikos vaizdavimą mobiliuosiuose įrenginiuose bei pagerins mobiliųjų aplikacijų veikimą paprastesniuose mobiliuosiuose telefonuose su ribotomis charakteristikomis. M3G 1.0 standarto patobulinimai skirti gausesnei, lygesnei, labiau realistiškai žaidimų grafikai, vartotojo sąsajai ir t.t., tačiau šie patobulinimai nenaudos daugiau mobiliojo įrenginio atminties ir resursų. M3G 2.0 standartas skirtas kurti aplikacijoms naujausiems mobiliesiems įrenginiams su moderniausiomis charakteristikomis.

Keletas pagrindinių naujojo M3G 2.0 standarto reikalavimų ir tikslų:

- Naujasis M3G 2.0 turi pilnai realizuoti ankstesniąją savo versiją M3G 1.0.
- Egzistuojančios mobiliosios aplikacijos ir jų vykdymo vienetai (MIDlets) turi veikti ir su naująja M3G 2.0 versija.
- Mobiliosios aplikacijos sukurtos naudojant OpenGL ES, turi veikti ir su naująja M3G 2.0 versija bei naujajame standarte turi būti realizuota kuo daugiau OpenGL ES komponentų.
- Naujajame standarte turi būti realizuotas šešėlių mechanizmas (ne tik naudojant tam tikrus funkcionalumą praplečiančius profilius).
- Aplikacijų veikimo skirtumai turi būti kiek galima sumažinti tarp skirtingų Java versijų.
- 3D objektai turi būti labiau kompresuojami - ir užimamos vietos, ir vykdymo laiko atžvilgiu.
- M3G 1.0 standarto paprastumas ir kompaktiškumas turi būti išsaugotas ir naujojo standarte.

- Naujasis M3G 2.0 standartas turi būti suderinamas su visais profiliais, kurie veikė su ankstesniu M3G 1.0 standartu.

M3G 2.0 standarto specialistų grupė tikisi, kad po testavimo ir tobulinimo periodo, kai bus išleista galutinė M3G 2.0 versija, mobiliųjų įrenginių gamintojai pakeis senąją M3G 1.0 versiją naująja versija ir įdiegs ją naujausiuose mobiliuosiuose įrenginiuose..

Nors M3G 2.0 standarto specialistų grupė planavo 2007 m. spalio mėnesį išleisti galutinę M3G 2.0 versiją, tačiau testavimo ir klaidų taisymo laikotarpis užtruko ilgiau nei planuota. Paskutiniame JSR ekspertų pranešime buvo žadėta galutinę versiją pateikti pirmajame 2008 m. ketvirtyje, tačiau ji dar vis nėra prieinama viešai.

5. Modeliavimo priemonių apžvalga

Šiuo metu populiariausia komercinė 3D modeliavimo priemonė yra 3D Studio Max (Autodesk, žr. [GRA01]). Tai galingas 3D scenų modeliavimo, animacijų bei objektų atvaizdavimo paketas, pirmaujantis pagal pirkimų skaičių. Taip pat gana populiari 3D modeliavimo priemonė yra Cinema 4D (Maxon). Ši priemonė neturi tiek galimybių kaip 3D Studio Max, tačiau ji labiausiai mėgstama dėl vartojimo paprastumo, išvengiant daugumos techninių detalių, pavyzdžiui vartotojas gali pats piešti tekstūras tiesiai ant modelių paviršiaus. Be populiariausio 3D Studio Max, Autodesk taip pat siūlo Maya 3D modeliavimo paketą, kuris populiariausias kuriant animacijas. Šis modeliavimo paketas garsėja sudėtingu priemonių valdymu bei naudojimu, tačiau tikriausiai yra pats galingiausias ir brangiausias 3D modeliavimo paketas.

Nors komerciniai Autodesk 3D modeliavimo paketai pripažinti labiausiai vartotojo poreikius atitinkančiomis 3D modeliavimo priemonėmis, tačiau nemaža dalis vartotojų renkasi nemokamas atviro kodo 3D modeliavimo priemones. Tokių modeliavimo priemonių kodas yra laisvai pasiekiamas ir gali būti redaguojamas bei tobulinamas paties vartotojo. Populiariausia nemokama 3D modeliavimo priemonė yra Blender (Blender foundation). Ši modeliavimo priemonė siūlo panašias funkcijas 3D modelių kūrimui kaip ir komercinės modeliavimo priemonės Maya, 3D Studio Max ar Cinema 4D. Blender modeliavimo priemonė yra sukurta pagal atvirojo kodo licenziją (GPL – General Public Licence) ir yra suderinama su visomis pagrindinėmis platformomis, kaip Windows, OS X, Linux, BSD, Sun.

Magistro darbe pagrindine modeliavimo priemone buvo pasirinktas Blender modeliavimo paketas, kuris yra nemokamas bei jo kodas gali būti redaguojamas. Blender paketas turi integruota Python interpretatorių, kuriame galima vykdyti skriptus, skirtus pasiekti 3D duomenims.

6. Egzistuojančių M3G eksportavimo priemonių apžvalga

Nors M3G standartas plačiai taikomas aplikacijose ir žaidimuose mobiliuosiuose telefonuose, tačiau egzistuoja nedaug patikimų eksportavimo priemonių, kurios leistų eksportuoti 3D modelį iš populiarių 3D modeliavimo paketų.

Šiuo metu populiariausia eksportavimo priemonė į M3G 1.0 standartą yra 3D Studio Max komercinis eksporteris. Šios eksportavimo priemonės valdymas gana patogus, turi galimybę eksportuoti tik tam tikras scenos dalis, prieš eksportuojant leidžia pakeisti tam tikrus duomenis, pvz. tekstūras, šviesos tipus ir t.t. Eksportavimo priemonė puikiai veikia eksportuojant paprastus, nesudėtingus 3D objektus, tačiau pagrindinis jos trūkumas – eksportuojant sudėtingesnius 3D objektus eksportavimo priemonė dažnai „pakimba“ taip ir nepranešusi, kurioje vietoje kilo klaida (žr. [EXP01]). Ši problema labai apsunkina sudėtingesnių 3D modelių eksportavimą į M3G formatą. Taip pat didelis šios priemonės trūkumas, kad naujausia 3D Max modeliavimo priemonė versija 9.0, o pastaroji eksportavimo priemonė nėra suderinta su šia versija, todėl eksportuojami modeliai dažnai skiriasi nuo pradinio.

JBenchmark3D kompanija siūlo eksportavimo priemonę Maya modeliavimo priemonei (Maya ir 3D Max priklauso tai pačiai kompanijai – Autodesk). Tai galinga eksportavimo priemonė palaikanti ne tik Windows operacinę sistemą, bet ir MAC bei Linux. Ši priemonė pati brangiausia iš visų egzistuojančių M3G eksportavimo priemonių. Ši priemonė yra dviejų tipų: nemokama ribota versija arba mokama pilna versija. Nemokama eksportavimo priemonės versija neleidžia eksportuoti net ir paprastų 3D modelių, todėl dauguma vartotojų dėl kainos skirtumo mieliau renkasi 3D Max modeliavimo priemonę su integruota eksportavimo priemone į M3G.

M3G eksportavimo priemonę, pasiūlytą Sony Erricson, turi komercinė MilkShape3D modeliavimo priemonė. MilkShape3D yra gana ribotas ir turi tik pagrindines funkcijas 3D modelių kūrimui. Dažniausiai ši priemonė naudojama scenų paprastiems žaidimams kūrimui. Pagrindinis vartotojų nusiskundimas šia priemone – problemos eksportuojant modelius su tekstūromis bei nepatogus eksportavimo priemonės valdymas.

Hi Corporation kompanija siūlo savo eksportavimo priemonę MascotCapsule (žr. [MCP01]), kuri gali eksportuoti failus iš .h3t formato į .m3g formatą. Didžiausia problema yra tai, kad modeliavimo priemonės neturi integruotų eksportavimo priemonių į .h3t formatą, todėl norint eksportuoti duomenis pirmiausia į .h3t formatą, reikia įdiegti tos pačios Hi Corporation kompanijos siūlomas eksportavimo priemones, skirtas Maya, 3D Max arba LightWave. Visos šios modeliavimo priemonės yra mokamos, todėl dažniausiai vartotojai perka 3D Max modeliavimo priemonę, kuri gali eksportuoti modelius tiesiai į .m3g be .h3t formato.

Vienintelė nemokama eksportavimo į M3G formatą priemonė gali eksportuoti iš Blender modeliavimo priemonės. Šią priemonę sukūrė Nelson-Games ir ji gali eksportuoti 3D modelius su tekstūromis, šviesomis, kameromis ir t.t. į M3G formatą. Tačiau priemonė nėra populiari, nes neturi patogios vartotojo sąsajos bei informatyvių klaidų pranešimų. Eksportavimo informaciją galima matyti tik komandinėje eilutėje.

7. Python programavimo kalba

Magistro darbe pagrindine modeliavimo priemone buvo pasirinktas Blender modeliavimo paketas, kuris yra realizuotas Python programavimo kalba. Blender yra nemokamas bei jo kodas gali būti redaguojamas. Blender modeliavimo paketas turi integruota Python interpretatorių, kuriame galima vykdyti skriptus, parašytus šia programavimo kalba. Šie skriptai gali naudoti Blender Python API, kad pasiektų programos 3D duomenis bei žymiai praplėstų Blender funkcionalumą.

Python yra dinaminė objektiškai orientuota programavimo kalba, kuri yra suderinama su kitomis programavimo kalbomis bei priemonėmis. Python buvo sukurtas pagal atviro kodo licenziją bei gali būti laisvai naudojamas netgi ir komerciniuose produktuose.

Python yra interpretuojamoji kalba, tai reiškia, kad Python kodas nėra konvertuojamas į kompiuterio skaitomą bei vykdomą kodą prieš paleidžiant programą, o pačios programos vykdymo metu. Toks programavimo kalbų tipas yra vadinamas skriptų kalba. Tačiau Python kalba, priešingai nei kitos skriptų kalbos Perl, TCL ar Ruby, nėra tokia ribota bei gali būti naudojama daugelyje aplikacijų bei integruota su kitomis programavimo kalbomis.

Python programavimo kalba pasižymi šiomis savybėmis:

- Turi aiškią bei lengvai skaitomą sintaksę;
- Turi objektinio programavimo galimybę;
- Dinaminio programavimo galimybė – nebūtina deklaruoti objekto tipo, nereikalingas duomenų tipų vertimo mechanizmas;

- Natūraliai išreiškiamas procedūros kodas;
- Turi bibliotekų modulius bei hierarchinius paketus;
- Turi klaidų valdymą;
- Kompaktiška programavimo kalba – Python kalba parašytas kodas yra vidutiniškai iki 5 kartų trumpesnis nei Java kalba parašytas kodas;
- Praplėtimai ir moduliai lengvai gali būti suprogramuoti C, C++ or Java programavimo kalbomis.

Viena iš Python siūlomų realizacijų – Jython (žr. [JYT01]), kuri yra suderinama su Java programavimo kalba. Jython yra aukšto lygio, dinaminė Python programavimo kalbos realizacija, skirta naudoti Python skriptų kalbą Java platformoje. Pridėjus Jython bibliotekas prie Java aplikacijos bibliotekų galima praplėsti Java aplikacijos funkcionalumą paprastais arba sudėtingesniais Jython skriptais.

Python skriptų kalba dažnai lyginama su Java programavimo kalba (žr. [PJC01]), tačiau Python neturi tiek daug galimybių kaip Java. Pagrindiniai Python trūkumai palyginus su Java programavimo kalba:

- Python programos veikia lėčiau nei Java programos;
- Python programavimo kalba skirta skriptų rašymui, bei neturi tokių galimybių kurti aplikacijas, skirtas žiniatinkliui kaip Java;
- Python neturi tokių vartotojo sąsajos kūrimo galimybių kaip Java;
- Java įskiepai (applets) gali būti paleisti naršyklėje, o Python tokios galimybės neturi
- Java turi puikų gijų (threads) valdymo mechanizmą;
- Java siūlo daug Java 2 platformos standartų, kurie suteikia aplikacijoms daugiau galimybių bei įgalina kurti aplikacijas mobiliams įrenginiams.

Python nebuvo pasirinkta pagrindine magistro darbo programavimo kalba dėl šių priežasčių:

- Java J2ME platforma suteikia galimybę kurti bei vykdyti aplikacijas mobiliuosiuose įrenginiuose;
- Python neturi tokių vartotojo sąsajos kūrimo galimybių kaip Java. Naudojant vartotojo sąsają galima patogiai manipuliuoti 3D scenos duomenimis bei juos keisti prieš eksportuojant į M3G;
- Java suteikia galimybę valdyti klaidas bei informatyviai apie jas pranešti vartotojui;

- Java suteikia galimybę eksportuoti M3G failą iš karto įkrauti į Java emuliatorių – mobilių telefoną.

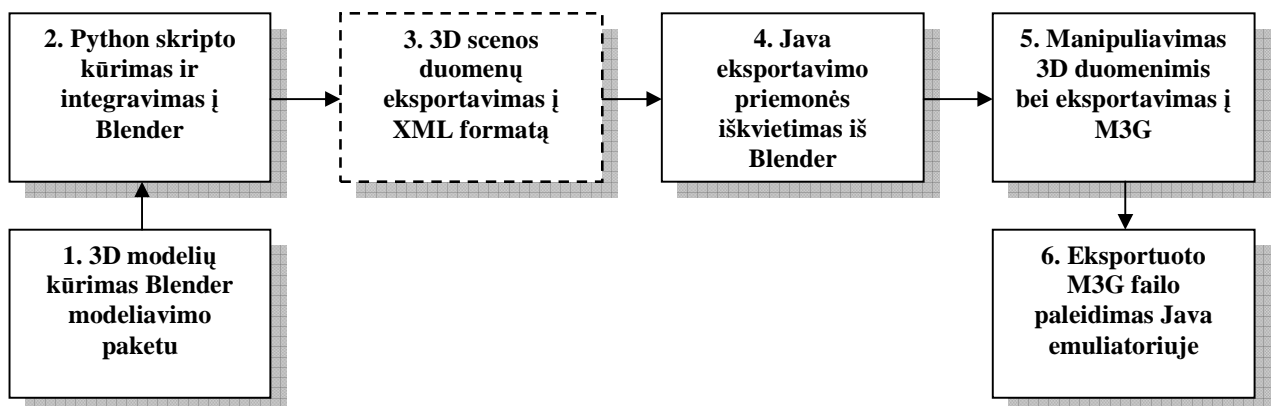
8. Eksportavimo priemonės programavimas

Pagrindine magistro darbo modeliavimo priemone buvo pasirinktas Blender modeliavimo paketas, kuris yra nemokamas, o Python programavimo kalba parašytais skriptais galima pasiekti 3D scenos duomenis.

Pagrindine magistro darbo programavimo kalba buvo pasirinkta Java – dėl patogios vartotojo sąsajos kūrimo galimybės, galimybės klaidų pranešimų pateikimo vartotojui bei eksportuoto modelio atvaizdavimo Java emuliacijoje – mobiliame telefone.

8.1 Eksportavimo schema

Eksportavimo priemonės kūrimas susideda iš šių dalių:



Sch.2. Eksportavimo į M3G standartą žingsniai

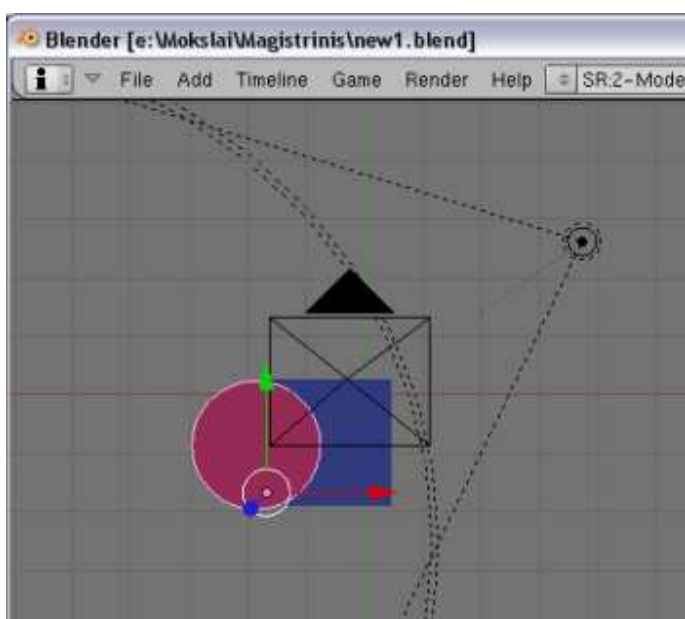
Eksportavimo į M3G standartą schemas platesnis aprašymas:

1. 3D modelių kūrimas Blender modeliavimo priemone. Šioje dalyje kuriami 3D modeliai, kurie vėliau bus eksportuojami į M3G standartą.
2. Python skripto programavimas, kuris skirtas pasiekti 3D modelių duomenis iš Blender modeliavimo priemonės. Python skriptas yra patalpintas į Blender `Blender\blender\scripts` direktoriją ir gali būti tiesiogiai pasiekiamas iš Blender pagrindinio meniu `File → Export → Blender to M3G` kaip papildomas įrankis.

3. Python skriptas eksportuoja 3D scenos duomenis iš Blender į XML formatą numatyta tvarka bei hierarchija. XML formatas buvo pasirinktas dėl patogaus duomenų išsaugojimo būdo bei duomenų nuskaitymo naudojant XML DOM iš Java eksportavimo priemonės. XML failas nėra matomas vartotojui, tai tik tarpinė duomenų laikmena – todėl scheme jis vaizduojamas punktyrine linija.
4. Baigęs darbą Python skriptas iškviečia išorinę Java eksportavimo priemonę. Java kalba suprogramuota eksportavimo priemonė nuskaityto Python skripto sugeneruotą XML failą.
5. Eksportavimo priemonė leidžia manipuluoti scenos duomenimis bei juos pakeisti prieš pradėdant eksportavimą. Pradėjus eksportavimą sukuriama duomenų sektoriai pagal M3G standarto specifikacijas bei duomenys eksportuojami į M3G failą.
6. Java eksportavimo priemonė turi galimybę nuskaityti eksportuotą M3G failą ir jį įkrauti į Java emuliatorių – mobilųjį telefoną. M3G failo testavimui sukurta aplikacija leidžia eksportuotą 3D objektą sukurti bei apžiūrėti jį iš visų pusių.

8.2 3D modelių kūrimas

3D modelio kūrimui buvo pasirinktas Blender modeliavimo paketas. Buvo sukurta keletas paprastų bei sudėtingesnių 3D modelių su apšvietimu ir kameromis. Šių 3D modelių duomenys bus eksportuojami iš Blender modeliavimo priemonės į XML formatą.



Pav.3. Kubas ir cilindras su kamera bei šviesomis

8.3.1 Python skripto naudojimas Blender

Python skriptas (žr. [JMS01]) yra naudojamas pasiekti 3D modelio duomenis Blender modeliavimo priemonėje. Naudojant Python skriptą galima importuoti arba eksportuoti 3D scenų duomenis į kitus formatus.

Blender modeliavimo priemone galima ne tik kurti 3D modelius bei animacijas, bet ir manipuluoti 3D modelio duomenimis. Visi 3D modelio duomenys gali būti pasiekiami naudojant Python skriptus. Python skriptai Blender modeliavimo priemonėje gali būti naudojami šiems tikslams:

1. 3D modelio duomenų modifikavimui bei jų importavimui ar eksportavimui į kitus formatus.
2. 3D žaidimuose Python skriptas gali būti naudojamas kontroliuoti žaidimo eigą įvykus tam tikrai numatytai situacijai.

Išorinis Python skriptas gali būti įdiegtas į Blender modeliavimo priemonę kaip papildomas komponentas (plug-in). Taip pat Blender turi specialų skriptų langą, kuriame galima rašyti bei vykdyti skriptus.

Blender moduliai turi būti importuoti skripto pradžioje, kad juos būtų galima naudoti skripto vykdymo metu. Blender modeliavimo priemonė turi šiuos modulius:

- **Blender** – pagrindinis Blender modulis
- **Types** – Blender tipų modulis
- **Nmesh** – žemo lygio prieiga prie Mesh modulio
- **Draw** – atvaizdavimo modulis
- **BGL** – OpenGL modulis
- **Object** – objektų modulis
- **Lamp** – apšvietimo modulis
- **Camera** – kameros modulis
- **Material** – medžiagų modulis
- **World** – pasaulio modulis
- **IPO** – animacijų modulis

Blender moduliai gali būti importuoti į skriptą dviem būdais:

- **import Blender.** Jei skripto pradžioje importavimo eilutė atrodys taip, tuomet visame skripte šio modulio globalių funkcijų ar klasių priekyje reikės naudoti modulio vardo (šiuo atveju Blender).

- **from Blender import ***. Jei skripto pradžioje importavimo eilutė atrodys taip, tuomet vartoti modulio vardą prieš visas šio modulio globalias funkcijas ar klases nebūtina.

Python skriptą galima įterpti į Blender modeliavimo priemonės meniu. Išoriniai Python skriptai turi būti išsaugoti ~/blender/scripts/ direktorijoje. Visi skriptai esantys šioje direktorijoje yra įkraunami į Blender modeliavimo priemonę jos paleidimo metu. Įdėjus šį papildomą kodą į Python skriptą, skriptas bus įkeltas į Blender meniu kaip papildomas įrankis:

```
#!BPY
"""
Name: 'Export to M3G (.M3G)'
Blender: 242
Group: 'Export'
Tooltip: 'Exporting to M3G file format for mobile applications'
"""
```

Pirmoji eilutė nurodo, kad tai yra Blender skriptas ir Blender modeliavimo priemonės paleidimo metu, šis skriptas bus įkeltas į Blender. Sekančios eilutės skirtos:

- nurodyti papildomo įrankio vardui, kuris bus matomas pagrindiniame Blender meniu;
- patalpinti Python skriptą į Blender modeliavimo priemonės pagrindinį meniu prie grupės „Export“;
- rodyti paaiškinamąjį tekstą užvedus pelę ant papildomo įrankio pavadinimo pagrindiniame meniu.

8.3.2 Python skripto programavimas

Python skriptas pirmiausia sukuria Scene klasės objektą iškviečiant metodą (žr. [PYT01]) GetCurrent(), kuris gražina visų 3D scenos objektų sąrašą, kurie buvo sukurti Blender modeliavimo priemonėje:

```
objList = Scene.GetCurrent()
```

Python skriptas eksportuoja trijų tipų 3D duomenis: kamerą, šviesas bei Mesh tipo 3D objektus. Kiekvienas 3D duomenų tipas yra apdorojamas skirtingai. Kiekvienas objekto tipas iš anksčiau nuskaityto 3D scenos sąrašo yra patikrinamas naudojant tipo nustatymo metodus:

```
Blender.isCamera(name)
Blender.isLamp(name)
Blender.isMesh(name)
```

Pirmiausia skriptas nuskaito ir apdoroja kameras. Eksportuojama tik aktyvi scenos kamera. Pasiesti kameros objektą iš 3D scenos Blender modeliavimo priemonėje galima šiuo kodu:

```
camobj = scene.getCurrentCamera()  
camera = Blender.getCamera(camobj.data)
```

Įvykdžius antrąją kodo eilutę galima pasiekti visus šio Camera klasės objekto parametrus (žr. [CAM01]):

- **Lens** - matomumo kampas. Kai šio parametro reikšmė nedidelė (pvz. 10), kamera yra toliau nuo objekto. Kuo ši reikšmė didesnė, tuo objektas yra arčiau kameros.
- **ClSta** - nurodo minimalią kameros matomumo ribą. Tai atstumas, nuo kurio artimiausi objektai bus matomi kameros lange.
- **ClEnd** - nurodo maksimalią kameros matomumo ribą. Kuo šis parametras didesnis, tuo daugiau objektų pasiekia kamera.

Galiausiai nuskaitoma kameros pozicija 3D scenoje: padėtis X, Y, Z ašių atžvilgiu, posūkis erdvėje bei mastelio informacija.

Apdorojus bei eksportavus kameros objektą su jo parametrais, toliau eksportuojamos scenos šviesos. Šviesos objektas bei jo parametrai gali būti pasiekti įvykdžius šį kodą:

```
lampobj = Blender.getObject(name)  
lamp = Blender.getLamp(lampobj.data)
```

Lamp klasės objektas turi šiuos parametrus:

- **Type** - šviesos tipas. Gali būti vienas iš šių tipų:
 - **Lamp** – tai taškinė šviesa, kuri šviečia visomis kryptimis vienodu intensyvumu.
 - **Sun** – tai pastovaus intensyvumo šviesa, šviečianti nurodyta kryptimi bei imituojanti saulės šviesą.
 - **Spot** – tai kūginė šviesa su spinduliu generuojamu iš šviesos šaltinio pozicijos su nurodyta kryptimi. Tai vienintelė šviesa Blender modeliavimo priemonėje, kuri generuoja objektų šešėlius.
 - **Hemi** – tai specifinė šviesa, skirta imituoti šviesą, sklindančią iš debesuoto dangaus. Jos pozicija nesvarbi, tačiau svarbi orientacija. Ši šviesa panaši į saulės šviesą.
- **Energ** – šviesos intensyvumas.
- **R** – šviesos raudonos spalvos komponentė.
- **G** – šviesos žalios spalvos komponentė.
- **B** – šviesos mėlynos spalvos komponentė.
- **Dist** – taškinės šviesos parametras, kuris nurodo atstumą, iki kurio šviesa išlieka pilno intensyvumo, o po kurio ji silpnės.

- **SpoSi** – kūginės šviesos parametras, kuris nurodo šviesos sklidimo kampą.
- **SpoBl** – kūginės šviesos parametras, kuris nurodo kaip susilieja kūginė šviesa su supančia neapšviesta zona.
- **Quad1, Quad2** – kūginės šviesos parametrai, skirti reguliuoti šviesos intensyvumo silpnėjimą su atstumu.
- **HaInt** – kūginės šviesos parametras, skirtas reguliuoti aureolės dydį.
- **ClipSta** – nurodo minimalią apšvietimo ribą, nuo kurios artimiausi objektai bus apšviesti.
- **ClipEnd** – nurodo maksimalią apšvietimo ribą, nuo kurios tolimesni objektai nebus apšviesti.

Galiausiai nuskaitoma šviesos pozicija 3D scenoje: padėtis X, Y, Z ašių atžvilgiu, posūkis erdvėje bei mastelio informacija.

Eksportavus kameras bei šviesas galima eksportuoti Mesh tipo objektų duomenis, t.y. trikampus ir keturkampus. Trikampiai ir keturkampiai yra saugomi Mesh klasės objektuose. Norint pasiekti Mesh objektą ir jo duomenis, reikia įvykdyti kodą:

```
meshobj = Blender.getObject(name)
mesh = Blender.getMesh(meshobj.data)
```

Eksportuosime šiuos Mesh klasės objekto parametrus:

- **Vertices** – saugo viršūnių sąrašą, kurį galima naudoti Python skripte. Face parametras naudoja viršūnių indeksus, taip tas pačias viršūnes gali naudoti keli skirtingi trikampiai ar keturkampiai.
- **Normals** – saugo sąrašą viršūnių normalių, kuriuos gali būti naudojamos detaliam (smooth) atvaizdavimui. Elementų skaičius šiame sąrašė turi atitikti viršūnių skaičių sąrašė.
- **Faces** – saugo paviršių sąrašą. Paviršių sudaro 6 integer tipo parametrai: pirmieji keturi parametrai nurodo keturkampio ar trikampo viršūnes. Jei tai trikampis, tuomet ketvirtasis parametras yra lygus 0. Penktasis parametras nurodo, ar trikampis, ar keturkampis turi būti atvaizduojamas detaliam (smooth), naudojant viršūnių normalių sąrašą. Paskutinis parametras nurodo material indeksą, kuri yra naudojama šiam paviršiui.
- **Texcoords** – saugo tekstūrų koordinačių sąrašą. Sąrašas gali būti tuščias, jei tekstūros nenaudojamos. Jei tekstūros naudojamos, tuomet tekstūrų sąrašas turi atitikti paviršių sąrašą. Tekstūrų koordinatės saugojamos kiekvienam paviršiui kaip sąrašas 4 masyvų (jei

naudojami trikampiai, tuomet naudojami tik 3 masyvai). Kiekvienas masyvas turi 2 narius – u ir v kryptis.

- **Texture** – saugo naudojamos tekstūros pavadinimą ir kelią iki tos tekstūros.
- **Materials** - 3D objekto Material klasės objekto parametrus galima pasiekti įvykdžius šį Python kodą:

```
Blender.getMaterial(name)
```

Eksportuosime šias Material klasės objekto savybes:

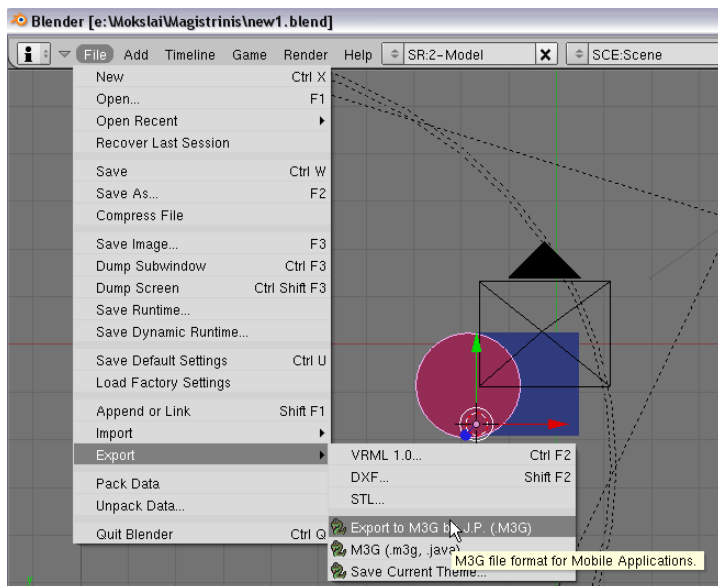
- **Name** – medžiagos vardas.
- **Ambient** – tai spalva, kurią objektas atspindi nuo savo paviršiaus. Spalva aprašoma RGB komponentėmis.
- **Emissive** – tai spalva, kurią objektas sugeria. Spalva aprašoma RGB komponentėmis.
- **Diffuse** – tai objekto spalva, kuri matoma tiesioginėje dienos šviesoje. Spalva aprašoma RGB komponentėmis
- **Specular** – tai objekto švytėjimo spalva, kuri išryškina šviesos atspindžius nuo objekto paviršiaus. Spalva aprašoma RGB komponentėmis.

Paruošus visus scenos duomenis eksportavimui, sukuriamas naujas XML failas, į kurį surašoma visa 3D scenos informacija.

8.3.4 Python skripto vykdymas

Python skriptas gali būti vykdomas tiesiogiai iš Blender modeliavimo priemonės kaip papildomas komponentas (plug-in). Skriptą galima redaguoti Blender modeliavimo priemonės skriptų lange.

Įdiegti Python eksportavimo skriptą į Blender modeliavimo priemonę galima įdėjus skriptą į Blender *Blender\blender\scripts* direktoriją. Tuomet skriptas gali būti tiesiogiai pasiekiamas iš Blender pagrindinio meniu *File* → *Export* → *Export to M3G (.M3G)*.



Pav.4. Blender eksportavimo priemonės iškvietimo langas

Paleidus skriptą iš pagrindinio meniu, skriptas nuskaity .blend faile esančią objektų informaciją ir ją įrašo į XML failą kartu su žymėmis hierarchinio medžio pavidalu. XML failas yra tarpinė 3D scenos duomenų laikmena, kurią vėliau apdoros Java programavimo kalba parašyta eksportavimo priemonė.

Sukūręs XML failą ir įrašęs eksportuotus duomenis, skriptas iškviečia Java kalba parašytą programavimo priemonę, kuri leidžia vartotojui eksportuoti duomenis į M3G standartą.

8.4 3D scenos duomenų saugojimas XML formatu

XML (eXtensible Markup Language) yra W3C konsorciumo rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba (žr. [WIK01]). Pagrindinė XML kalbos paskirtis yra užtikrinti lengvesnį duomenų keitimąsi tarp skirtingo tipo sistemų ar aplikacijų. XML formate duomenys saugojami žymėse - taguose, pvz. <name>, <coord> ir t.t. XML dokumento medis konstruojamas nuo tėvinio mazgo šakomis link lapų. Mazgai gali turėti dukterinių mazgų, o mazgas, neturintis dukterinių mazgų, yra vadinamas lapu. Naudojant W3C konsorciumo standartą DOM Java ekportavimo priemonėje galima nesunkiai nuskaityti visus XML duomenis. DOM (Document Object model, žr. [DOM01]) – dokumento objektų modelis yra nepriklausomas nei nuo

platformos, nei nuo programavimo kalbos, bei leidžia kitoms programoms ar skriptams dinamiškai pasiekti ir atnaujinti dokumento turinį, struktūrą ar stilių.

3D scenos duomenys iš Blender modeliavimo priemonės yra eksportuojami į XML formatą hierarchinio medžio pavidalu. Hierarchijos viršūnėje yra **<world3D>** tagas. Toliau hierarchiniame XML medyje gali būti šie elementai:

- **<cameras>**. Šis XML elementas saugo 3D scenos kamerų informaciją. 3D scena gali turėti daugiau nei 1 kamerą, todėl šis XML elementas turi tiek dukterinių **<camera>** elementų, kiek yra kamerų scenoje. **<camera>** elementas gali turėti 2 atributus: kameros pavadinimą bei jos tipą – 0, jei **persp** kameros tipas; 1, jei **ortho** kameros tipas. Kameros elementas gali turėti šiuos dukterinius elementus:
 - **<Lens>** – kameros matomumo kampas.
 - **<ClipStart>** - nurodo minimalią kameros matomumo ribą.
 - **<ClipEnd>** - nurodo maksimalią kameros matomumo ribą.
 - **<Transform>** - nurodo kaip kamera yra nutolusi nuo atskaitos taško. Šis elementas gali turėti dukterinius elementus: **scale** (mastelio keitimas pagal X, Y ir Z ašis), **rotate** (kameros posūkis pagal vektorių nurodytu kampu) bei **translate** (kameros postūmis pagal X, Y, Z ašis).
- **<lights>**. Šis XML elementas saugo 3D scenos šviesų informaciją bei gali turėti tiek dukterinių **<light>** elementų, kiek šviesų yra 3D scenoje. **<light>** elementas gali turėti 2 atributus: šviesos šaltinio pavadinimą bei tipą - **Lamp** (taškinė šviesa), **Spot** (kūginė šviesa), **Sun** (saulės šviesa) bei **Hemi** (šviesa iš debesuoto dangaus). Šviesos elementas gali turėti šiuos dukterinius elementus:
 - **<Energy>** – šviesos intensyvumas.
 - **<lineardist>**, **<quaddist>** – kūginės šviesos parametrai, skirti reguliuoti šviesos intensyvumo silpnėjimą su atstumu.
 - **<Color>** – šviesos spalva, kurioje nurodoma kiekviena RGB spalvų modelio komponentė
 - **<ClipStart>** – nurodo minimalią apšvietimo ribą, nuo kurios artimiausi objektai bus apšviesti.
 - **<ClipEnd>** – nurodo maksimalią apšvietimo ribą, nuo kurios tolimesni objektai nebus apšviesti.

- **<Dist>** – taškinės šviesos parametras, kuris nurodo atstumą, iki kurio šviesa išlieka pilno intensyvumo, o po kurio ji silpnės.
- **<SpotSize>** – kūginės šviesos parametras, kuris nurodo šviesos sklidimo kampą.
- **<SpotBlend>** – kūginės šviesos parametras, kuris nurodo kaip susilieja kūginė šviesa su supančia neapšviesta zona.
- **<HaInt>** – kūginės šviesos parametras, skirtas reguliuoti aureolės dydį.
- **<Transform>** - nurodo kaip šviesos šaltinis yra nutolęs nuo atskaitos taško. Šis elementas gali turėti dukterinius elementus: **scale** (mastelio keitimas pagal X, Y ir Z ašis), **rotate** (šviesos šaltinio posūkis pagal vektorių nurodytu kampu) bei **translate** (šviesos šaltinio postūmis pagal X, Y, Z ašis).
- **<objects>**. Šis XML elementas saugo informaciją apie Mesh tipo 3D objektus bei gali turėti tiek dukterinių elementų kiek Mesh tipo objektų yra scenoje. **<objects>** tagas atributų parametru neturi, tačiau jo dukteriniai elementai **<object>** gali turėti atributą – **name**, kuris nusako objekto pavadinimą, duotą Blender modeliavimo priemonėje. **<object>** elementas gali turėti šiuos dukterinius elementus:
 - **<geometryinfo>** – šis elementas saugo objekto koordinates. **<geometryinfo>** elementas atributų neturi, tačiau turi dukterinį elementą **<coordinates>**, o šis dukterinį elementą **<array>**, kuris saugo objekto viršūnių masyvą. **<array>** elementas turi atributus *vertex*, kuris saugo trikampių, į kuriuos yra sudalintas objektas, skaičių, bei *texture* atributą, kuris saugo kelią iki naudojamos tekstūros. **<array>** elementas turi **<triangle>** dukterinį elementą, kuris saugo trikampio viršūnės informaciją X, Y, Z ašyje, normalių informaciją X, Y, Z ašyje bei UV informaciją.
 - **<transform>** - šis elementas nurodo kaip objektas yra nutolęs nuo atskaitos taško. Šis elementas gali turėti dukterinius elementus: **<scale>** (mastelio keitimas pagal X, Y ir Z ašis), **<rotate>** (objekto posūkis pagal vektorių nurodytu kampu) bei **<translate>** (objekto postūmis pagal X, Y, Z ašis).
 - **<appearance>** - šis elementas saugo objekto išvaizdos informaciją. Jis gali turėti dukterinį elementą **<material>**. Šis XML elementas turi **name** atributą, kuris nusako išvaizdos vardą, duotą Blender modeliavimo priemonėje. **<material>** elementas gali turėti dukterinius elementus **<shine>** ir **<color3f>**. **<shine>** elementas saugo objekto išvaizdos švytejimo informaciją. **<color3f>** elementas turi atributą **type**, nurodantį ar tai ambient, emissive, diffuse ar specular spalva bei

dukterinius elementus <red>, <green> ir <blue>, kurie nurodo kiekvieną RGB spalvų modelio komponentę.

Visi 3D scenos duomenys, kurie yra saugomi XML formatu, yra išdėstyti hierarchiniu medžiu (žr. Priedą Nr. 1).

8.5 Eksportavimo priemonės kūrimas Java programavimo kalba

Eksportavimo priemonės kūrimui buvo pasirinkta netBeans IDE 6.0 programinė aplinka dėl patogaus Mobility Pack, kuris naudojamas kurti, testuoti, kompiliuoti aplikacijas, skirtas mobiliesiems įrenginiams, turintiems J2ME platformą.

Java programavimo kalba suprogramuota aplikacija nuskaityto Python skripto sugeneruotą XML failą, apdoroja jo duomenis ir eksportuoja į M3G formatą. Tarpinis XML failas tarp Blender modeliavimo priemonės ir Java aplikacijos buvo pasirinktas dėl patogaus duomenų saugojimo ir nuskaitymo būdo. Eksportavimo priemonė nuskaityto XML failo duomenis ir paruošia 3D scenos elementų hierarchinį vaizdą bei jį pavaizduoja vartotojui. Eksportavimo priemonė leidžia tiesiogiai manipuluoti 3D scenos duomenimis ir prieš eksportuojant pakeisti eksportuojamus objektus.

Paspaudus eksportavimo mygtuką pradedamas eksportavimo procesas: sukuriamas naujas M3G failas ir į jį įrašoma apdorota 3D scenos informacija. Eksportavimo metu pateikiamas informatyvus atliekamų žingsnių aprašymas bei įvykusių klaidų pranešimas.

Iš XML formato eksportuotas M3G failas yra įkraunamas į mobiliąją aplikaciją ir atidaromas Java emuliatoriumi – mobiliuoju telefonu.

8.5.1 Projekto struktūra

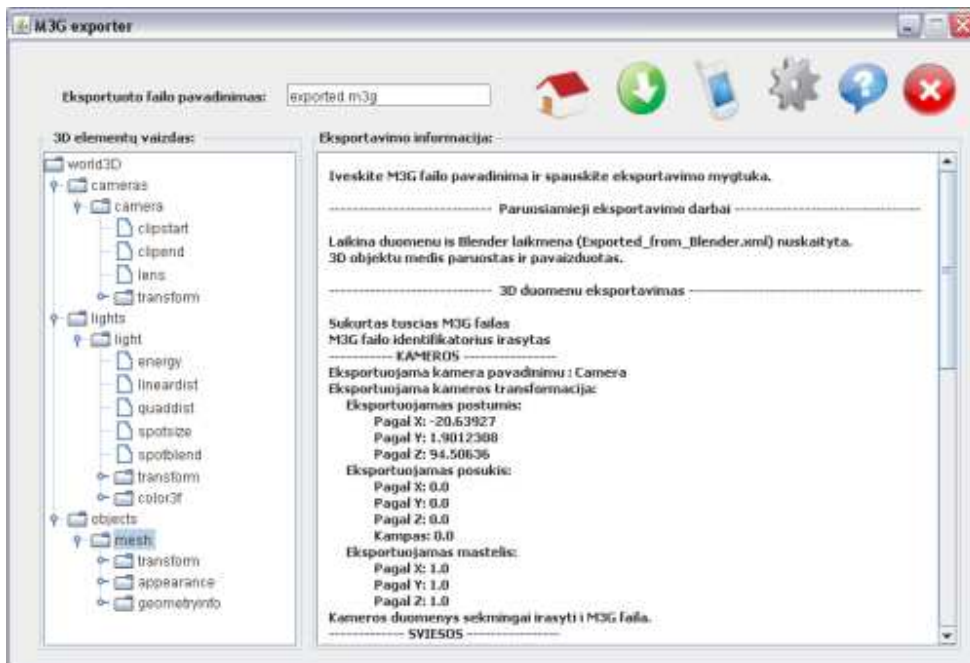
Eksportavimo priemonę sudaro dvi dalys:

- **Eksportavimo priemonės Java klasės.** Ši projektą sudaro vartotojo sąsajos, darbo su XML failu bei eksportavimo ir naujo M3G failo sukūrimo Java klasės.
- **Eksportuoto M3G failo testavimo projektas.** Ši projektą sudaro 2 Java klasės, kurios sukuria mobiliąją aplikaciją, įkrauna eksportuotą M3G failą ir jį paleidžia Java emuliatoriuje – mobiliojo telefono imitatoriuje. Naudojantis mobiliojo telefono mygtukais galima priartinti ar pasukti eksportuotą 3D modelį.

8.5.2 Aplikacijos vartotojo sąsaja

Aplikacijos vartotojo sąsają sudaro šie skydėliai:



- *Viršutinis valdymo skydelis* – meniu juosta, kurią sudaro pagrindinių funkcijų greito iškvietimo mygtukai bei failo pavadinimo įvedimo funkcija. Kiekvienos greitosios funkcijos mygtuko platesnis aprašymas yra rodomas užvedus pelę ant mygtuko.
- *3D scenos elementų skydelis* – 3D scenos hierarchinis medis. Kiekvieną medžio tėvinį mazgą galima praplėsti iki lapų lygio.
- *Eksportavimo informacijos skydelis* – vartotojui pateikiama informacija apie eksportavimo procesą bei atliekamus veiksmus.







Pav.5. Eksportavimo priemonės vartotojo sąsaja.

8.5.3 Aplikacijos valdymas

Aplikacijos pagrindinių funkcijų greitojo iškvietimo mygtukų paaiškinimas:

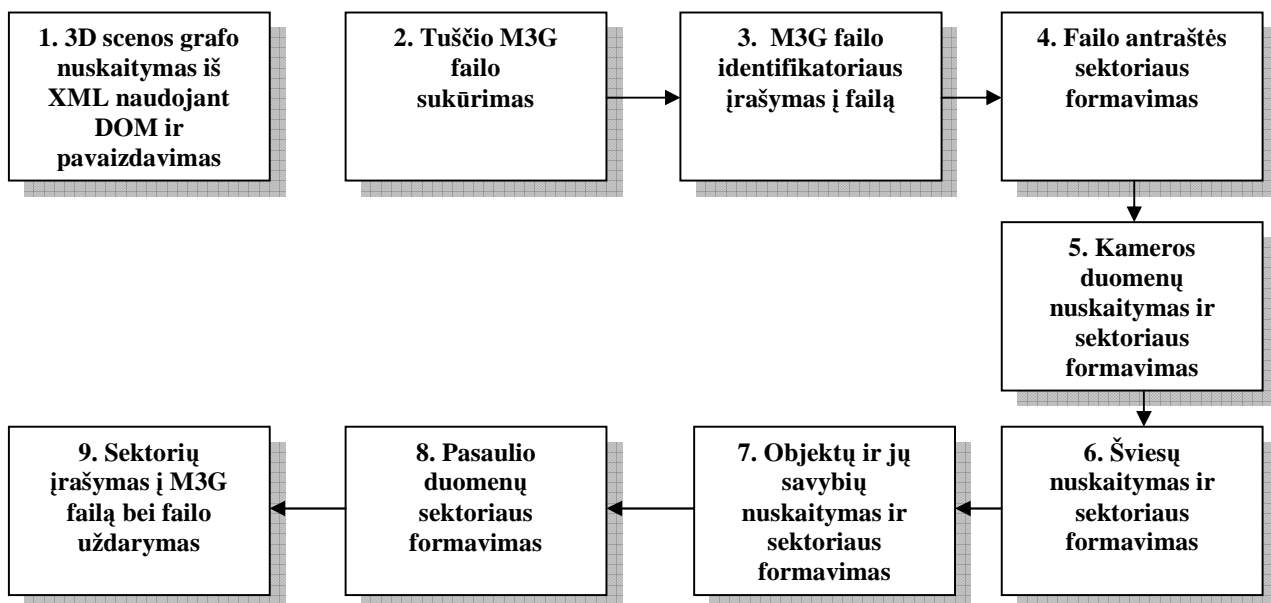
-  - atidaromas direktorijos pasirinkimo langas, kuris leidžia pasirinkti direktoriją, kurioje turėtų būti išsaugotas eksportuotas M3G failas.
-  - eksportavimo mygtukas. Jį paspaudus pradedamas duomenų eksportavimas į M3G standartą.

-  - emuliacijos paleidimo mygtukas. Paspaudus šį mygtuką atidaromas emuliacijos langas, kuris parodo kaip atrodys eksportuotas M3G failas mobiliojoje aplikacijoje.
-  - nustatymų mygtukas. Paspaudus šį mygtuką, galima keisti eksportavimo parametrus.
-  - pagalbos mygtukas. Paspaudus šį mygtuką, parodomas langas su eksportavimo priemonės veikimo bei reikalavimų aprašymu.
-  - išėjimo mygtukas. Paspaudus šį mygtuką, eksportavimo priemonė uždaroma.

8.5.4 Eksportavimo metu atliekami veiksmai

Eksportavimo priemonė iškviečiama iš Blender modeliavimo priemonės pagrindinio meniu, kai Python skriptas išsaugo duomenis į laikiną laikmeną – XML failą. Įvedus eksportuoto M3G failo pavadinimą ir paspaudus eksportavimo mygtuką, pirmiausia nuskaitymas XML failas bei naudojant DOM sukuriama XML elementų hierarchinis medis ir atvaizduojamas eksportavimo priemonės 3D scenos elementų skydelyje. Atlikti veiksmai aprašomi eksportavimo informacijos skydelyje.

Eksportavimo metu atliekamų veiksmų schema:



Sch.3. Eksportavimo metu atliekamų veiksmų schema

Platesnis eksportavimo schemos veiksmų aprašymas:

1. Paspaudus eksportavimo mygtuką pradedamas XML failo nuskaitymas naudojant DOM. Nuskaicius visus XML faile esančius elementus, jie pavaizduojami kaip hierarchinis medis kairiajame eksportavimo priemonės lange.
2. Sukuriamas tuščias M3G failas, kuris yra pavadinamas vardu, įvestu eksportuoto failo pavadinimo lauke.
3. Į sukurtą tuščią failą įrašomas M3G failo identifikatorius – 12 baitų seka, pagal kurią identifikuojamas M3G failas. M3G specifikacijoje (žr. [FFM01]) nurodytas identifikatorius: Byte[12] FileIdentifier = { 0xAB, 0x4A, 0x53, 0x52, 0x31, 0x38, 0x34, 0xBB, 0x0D, 0x0A, 0x1A, 0x0A }
4. Šiame žingsnyje sukuriama failo antraštės sektorius. Antraštės sektorius susideda iš sektoriaus antraštės, kurioje nurodomas sektoriaus tipas ir sektoriaus dydis, bei sektoriaus objekto, kuris susideda iš šių dalių:
 - a. M3G standarto versija (dabartinė 1.0);
 - b. Loginė reikšmė, kuri nurodo ar objektai turi nuorodų į išorinius failus;
 - c. Failo dydis. Šis laukas yra naudojamas įkraunant failą į mobiliąją aplikaciją, todėl privalo būti teisingai nurodytas;
 - d. Apytikslis failo dydis. Šis laukas gali būti naudojamas apytiksliam eksportavimo laikui apskaičiuoti.
 - e. Autoriaus laukas, kuriame galima nurodyti failo autoriaus duomenis.
5. Kameros duomenų nuskaitymas. Kameros duomenis saugo sukurta *Camera* klasė. Naudojant šią klasę kameros savybės bei transformacija yra įrašoma į sektoriaus objektą, o sektoriaus antraštę sudaro sektoriaus tipas bei sektoriaus dydis. Jei scenoje nėra nė vienos kameros, tuomet parodomas klaidos pranešimas, kuris paaiškina, kad eksportavimas negalimas, jei scenoje nėra kameros.
6. Šviesos duomenų nuskaitymas. Šviesos duomenis saugo sukurta *Light* klasė. Naudojant šią klasę šviesos savybės, spalva bei transformacija yra įrašoma į sektoriaus objektą, o sektoriaus tipas bei dydis įrašomas į sektoriaus antraštę. Jei scenoje nėra nė vienos šviesos, parodomas informacinis pranešimas, kuris leidžia pasirinkti, ar pridėti į sceną numatytąsias šviesas.
7. Objektų nuskaitymas. Šiame žingsnyje nuskaityma visa objektų informacija bei formuojamas sektoriaus objektas tokia tvarka: geometrijos informacija – viršūnės bei normalės, objektų išvaizdos informacija bei transformacija. Sektoriaus antraštę sudaro sektoriaus tipas bei dydis.

8. Šiame žingsnyje kuriamas pasaulio sektorius. Šio sektoriaus objektas saugo visų scenos objektų indeksus, aktyvios kameros indeksą bei fono (Background) indeksą. Sektoriaus antraštę sudaro sektoriaus tipas bei dydis.
9. Pabaigus sektorių formavimą, visi sektoriai įrašomi į M3G failą ir failas uždaromas. Eksportavimas baigtas.

Ekportavus visus duomenis į M3G formatą, galima šį failą įkrauti į suprogramuotą aplikaciją ir paleisti ją Java emuliatoriumi (paspaudus telefono mygtuką eksportavimo priemonėje) bei priartinti arba pasukti 3D modelį. Paspaudus telefono mygtuką iškviečiamas Java procesas, kuris atnaušina emuliatoriaus JAR paketą ir įdeda eksportuotą M3G failą į JAR paketą. Tuomet iškviečiamas Java emuliatorius ir jame pavaizduojamas eksportuotas M3G failas.

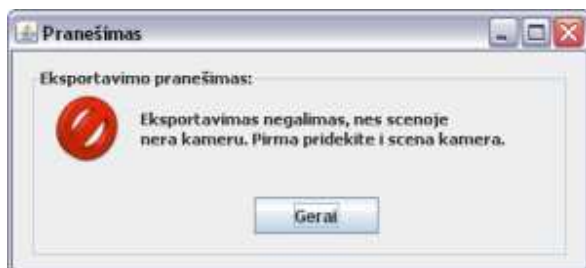
8.5.5 Eksportavimo informacijos bei klaidų pranešimai

Eksportavimo priemonė iš karto pateikia eksportavimo metu atliekamų veiksmų informaciją rezultatų lange. Atliekamo veiksmo rezultatas nedelsiant išvedamas į rezultatų langą programai dar nebaigus eksportavimo. Rezultatų išvedimą valdo sukurta *TextAreaOutputStream* klasė, kuri paveldi *java.io.OutputStream* klasę, atsakingą už baitų srauto išvedimą. Sukurtos klasės pagalba, naudojant *System.out.println* komandą, visas baitų srautas yra nukreipiamas į rezultatų langą (JTextArea objektą) vietoj įprasto baitų srauto išvedimo.

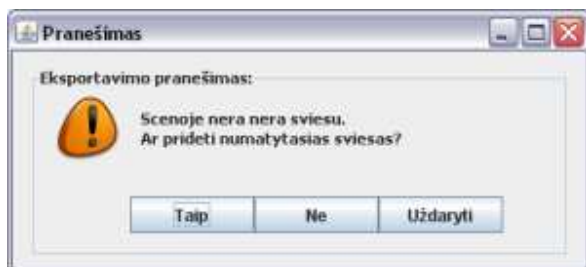
Klaidų valdymui bei pranešimui buvo sukurta klasė *Warning*, kuri atidaro naują pranešimo langą (JFrame objektą). Šios klasės objektą galima sukurti kitose klasėse, bei nustačius pranešimo tipą bei informacinį pranešimą, pavaizduoti pranešimą vartotojui.

Klaidų pranešimai gali būti dviejų tipų:

1. Klaida, dėl kurios eksportavimas negalimas. Pranešimas aprašo kodėl atsirado klaida ir kaip ją galima ją ištaisyti.



2. Įspėjimas apie eksportavimo metu aptiktus 3D scenos trūkumus. Eksportavimo procesas gali būti pratęstas paspaudus mygtuką „Taip“. Paspaudus mygtuką „Ne“ arba „Uždaryti“ eksportavimo procesas yra nutraukiamas.



9. Eksportavimo rezultatų palyginimas

Eksportavimo priemonės rezultatams įvertinti buvo pasirinktos dvi komercinės eksportavimo priemonės, skirtos Maya ir MilkShape 3D modeliavimo priemonėms, ir pagal pasirinktus kriterijus palyginti gauti rezultatai.

9.1 Eksportavimo priemonių pasirinkimas

Eksportavimo priemonės rezultatų palyginimui buvo pasirinktos populiariausios eksportavimo priemonės į M3G standartą. Eksportavimo rezultatai buvo palyginti su šiomis eksportavimo priemonėmis:

- Populiariausia komercinė eksportavimo priemonė yra integruota į kompanijos Autodesk modeliavimo priemonę 3ds Max (versija 9.0, žr. [AUT01]). Ji veikia kaip papildomas įrankis ir gali būti pasiekta per pagrindinį meniu. Prieš eksportuojant į M3G, ši eksportavimo priemonė leidžia pakeisti eksportavimo nustatymus bei pačius duomenis – nustatyti aktyvią kamerą, pergrupuoti eksportuojamus objektus, eksportuoti tik dalį visos 3D scenos objektų ir t.t. Eksportavimo priemonė taip pat turi galingą tektūrų valdymo mechanizmą – leidžia pakeisti objekto tekstūras bei jas redaguoti specialiaame tektūrų redagavimo lange.
- M3G eksportavimo priemonė (versija 1.3.3, žr. [MAY01]) skirta Autodesk Maya modeliavimo priemonei (versijai Maya 2008, žr. [MAY02]). Ši eksportavimo priemonė, kaip ir pats Maya modeliavimo paketas, yra komercinė, tačiau kūrėjai leidžia išbandyti

apribotą M3G eksportavimo priemonės versiją nemokamai. Kai kurios šios versijos funkcijos yra uždraustos nemokamoje versijoje:

- Kadangi priemonė nėra skirta komerciniams tikslams, todėl kiekvienas failas yra pažymimas specialiais simboliais;
 - Priemonė neleidžia pasirinkti jokių nustatymų prieš eksportuojant į M3G;
 - Neleidžiama trikampių trianguliacija (paviršiaus skaidymas į sujungtus trikampius, kurie padeda tiksliau ir lengviau atvaizduoti paviršių);
 - Duomenys M3G faile nėra spaudžiami;
 - Viršūnės yra saugojamos Short tipu (bet ne Byte);
 - Vartotojo objektams priskirti UserID parametrai nėra eksportuojami;
 - Viršūnių spalvos neeksportuojamos;
 - Leidžia eksportuoti animacijas.
-
- M3G eksportavimo priemonė (versija 1.0, žr. [MLK01]) skirta MilkShape3D modeliavimo priemonei (versijai 1.8.2, žr. [MLK02]) ir veikia kaip papildomas įrankis, pasiekiamas iš pagrindinio meniu (File → Export → M3G). Šią eksportavimo priemonę pasiūlė kompanija Sony Ericsson ir ji yra nemokama. MilkShape 3D modeliavimo paketas buvo sukurtas kompanijos chUmbaLum sOft. MilkShape 3D yra gana populiarus modeliavimo paketas, nors iš pradžių jis buvo sukurtas tik žaidimo Half-Life 3D scenoms kurti. Vėliau modeliavimo paketui buvo suteikiama vis daugiau funkcionalumo, jis pradėjo palaikyti net išorinius failų formatus, skirtus kitoms modeliavimo priemonėms. Šiuo metu MilkShape 3D modeliavimo paketas palaiko 70 skirtingų failo formatų. Ši eksportavimo priemonė pasižymi šiomis savybėmis:
 - Priemonė leidžia pasirinkti nustatymus prieš eksportuojant į M3G;
 - Leidžiama pasirinkti, ar naudoti trikampių trianguliaciją;
 - Eksportuojama kartu su fonu (background) objektu;
 - Jei scenoje nebuvo sukurta kamerų ar šviesų, tuomet automatiškai pridedama kamera ir šviesos ir jos yra eksportuojamos kartu su modeliu.
 - Leidžia eksportuoti animacijas.
-
- „Nelson Games” siūloma M3G eksportavimo priemonė (versija 0.8, žr. [NEL01]) skirta Blender modeliavimo priemonei (versijai 2.4.4) ir veikia kaip papildomas įrankis, pasiekiamas iš pagrindinio meniu (File → Export → M3G (.m3g, java)). Ši eksportavimo

priemonė yra nemokama kaip ir pats Blender modeliavimo paketas, o ją pradėjo kurti keli programuotojai, kurie nenorėjo pirkti komercinių eksportavimo priemonių. Ši eksportavimo priemonė nuolat tobulinama daugumos programuotojų, kurie siūlo savo idėjas bei patobulinimus šiai eksportavimo priemonei. Eksportavimo priemonė pasižymi šiomis savybėmis:

- 3D objektai scenoje nėra spaudžiami (planuojama realizuoti sekančioje versijoje);
- Priemonė leidžia pasirinkti nustatymus prieš eksportuojant į M3G;
- Vartotojo objektams priskirti UserID parametrai yra eksportuojami;
- Leidžia eksportuoti animacijas.

9.2 Eksportavimo rezultatų palyginimo kriterijai

Eksportavimo priemonės rezultatams palyginti buvo sukurti keli vienodi modeliai trijose skirtingose modeliavimo priemonėse 3D Max, Blender, Maya ir Milkshape 3D. Visi objektai 3D scenoje visose modeliavimo priemonėse buvo išdėstyti tose pačiose vietose bei jų savybės yra vienodos. Visi modeliai buvo eksportuoti į M3G formatą bei palyginimui pasirinkti šie kriterijai:

- Neatitikimai palyginus su originaliu failu ir eksportuotu M3G failu (pvz. pozicija erdvėje, objektų savybės bei transformacijos, apšvietimas, kameros);
- Vartotojo sąsaja;
- Eksportavimo nustatymų keitimo galimybė;
- Failo dydis;
- Informacijos bei klaidų pranešimų pateikimas vartotojui bei eksportavimo priemonės patikimumas.

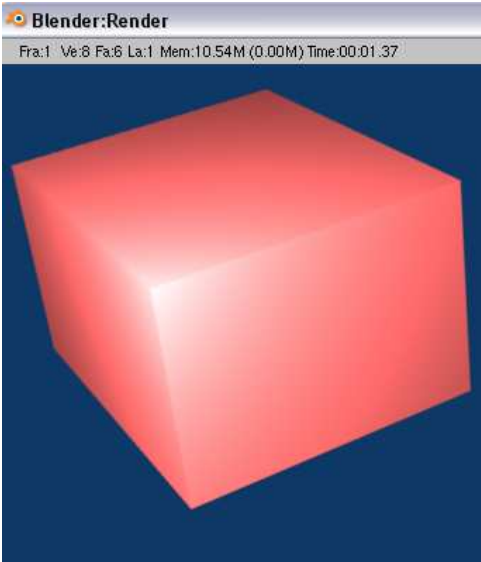
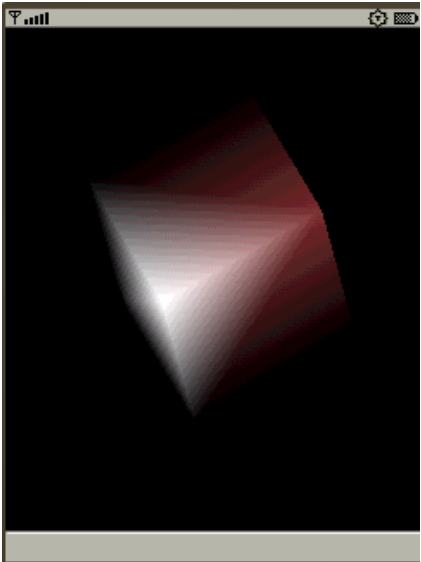
9.3 Eksportavimo rezultatų palyginimas

Eksportavimo priemonių rezultatams palyginti buvo sukurti 3 skirtingi modeliai kiekvienoje iš modeliavimo priemonių. Trumpas kiekvieno modelio aprašymas:

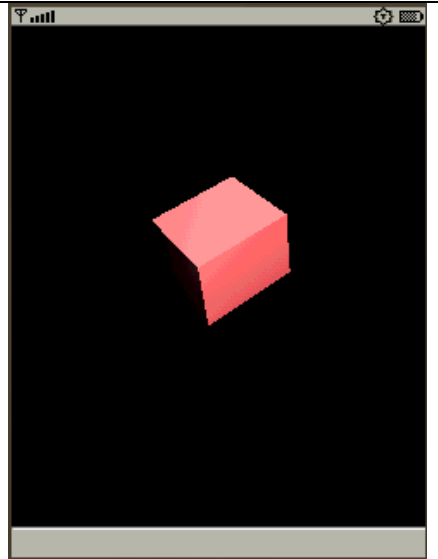
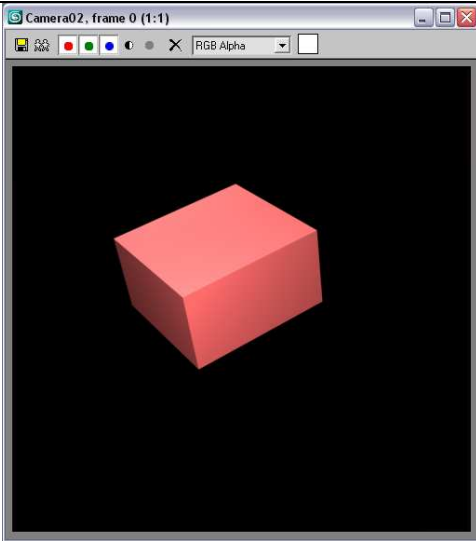
1. Modelis atvaizduoja 3D kubą, pasuktą bei pastumtą nuo atskaitos taško per tam tikrą atstumą. Kubas turi priskirtą raudoną išvaizdą. Modelis turi taškinio tipo šviesą bei kamerą, nukreiptą tiesiai į kubą. 3D modelio detalus aprašas: 8 viršūnių, 12 trikampių.

2. Modelis atvaizduoja du skirtingų tipų 3D objektus su priskirtomis skirtingomis išvaizdomis bei skirtingomis transformacijomis. Modelis turi dviejų skirtingų tipų šviesas (kūginę ir taškine) bei kamerą, nukreiptą į sujungtus objektus. 3D modelio detalus aprašas: 94 viršūnių, 180 trikampių.
3. Modelis atvaizduoja sudėtingą 3D modelį – medį, kuris susideda iš trijų objektų – kamieno, lapų bei šakų. Objektai turi skirtingas transformacijas, 6 skirtingas išvaizdas bei 6 taškines šviesas. 3D modelio detalus aprašas: 614 viršūnių, 1040 trikampių.

Rezultatai, gauti lyginant pasirinktas eksportavimo priemones:

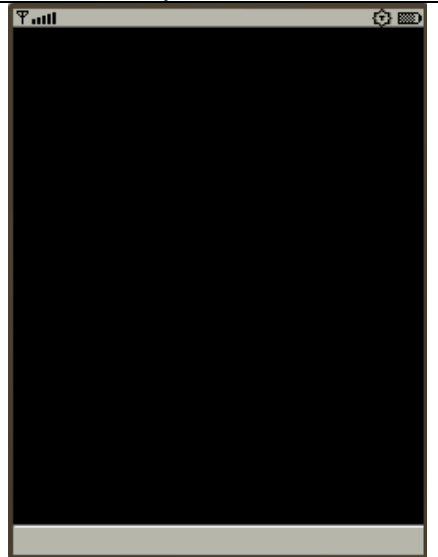
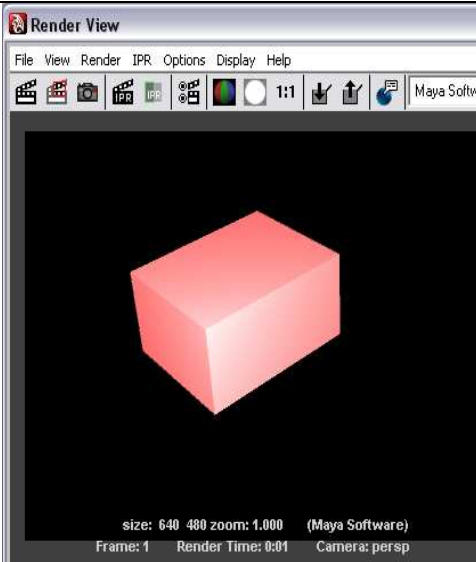
Modelio Nr.	Priemonė	Originalus modelis	Eksportuotas į M3G
1.	Magistro darbo eksportavimo priemonė		 <p data-bbox="1134 1529 1474 1561">M3G failo dydis: 1,15 KB.</p>

Autodesk 3D Max
eksportavimo
priemonė

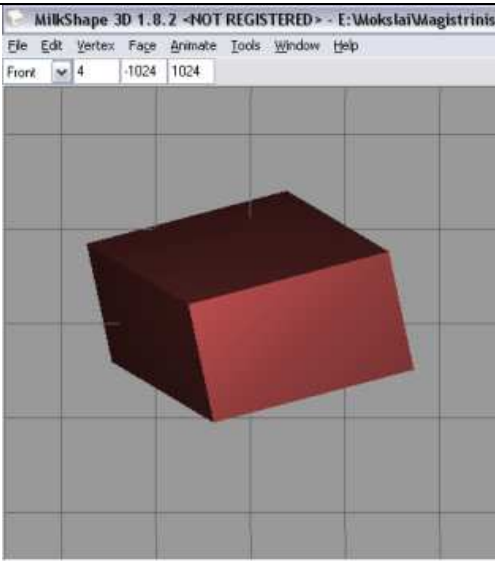
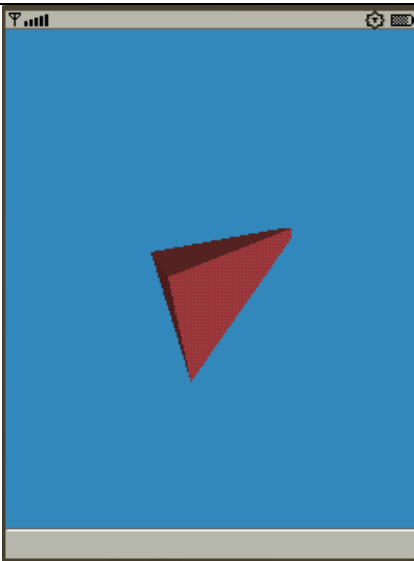
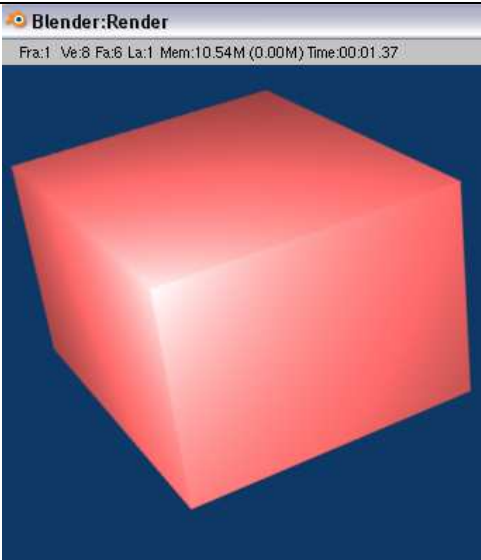
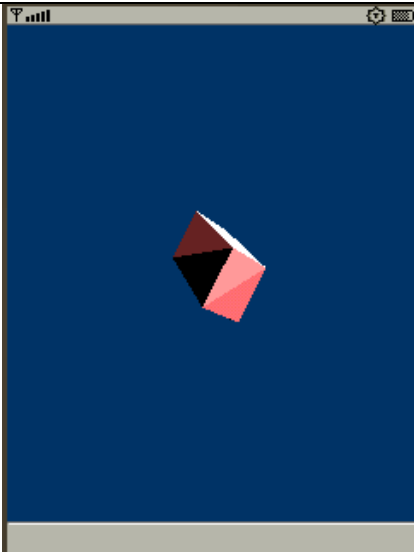


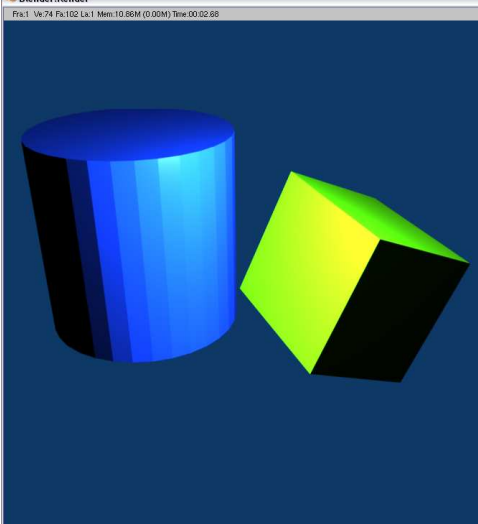
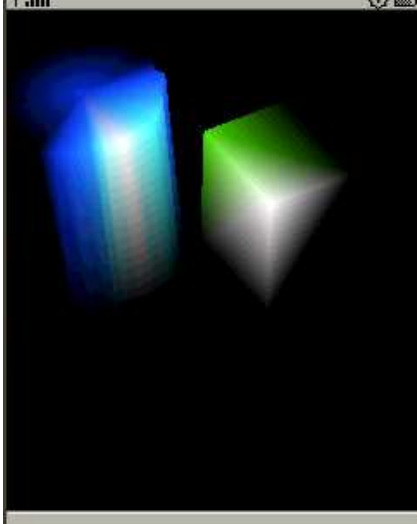
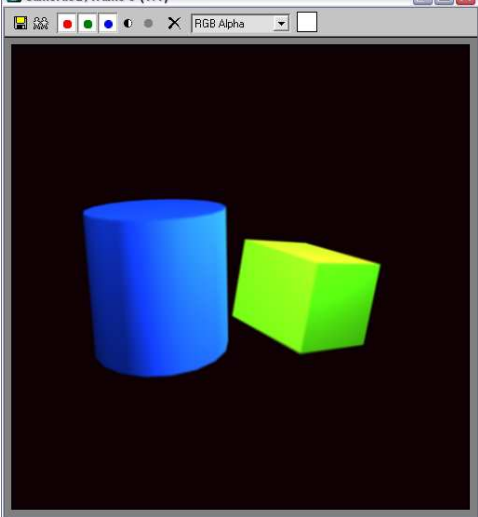
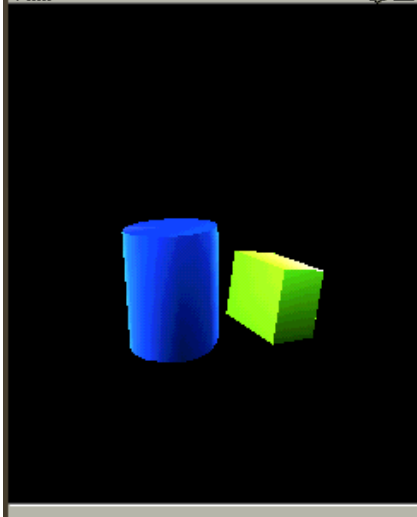
M3G failo dydis: 1,2 KB.

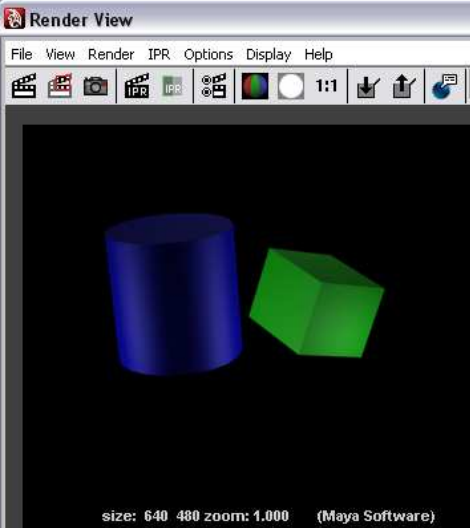
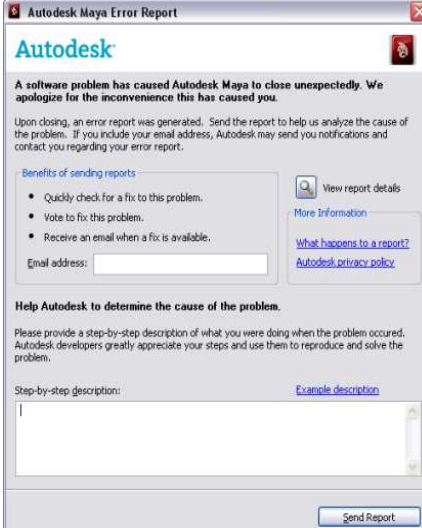
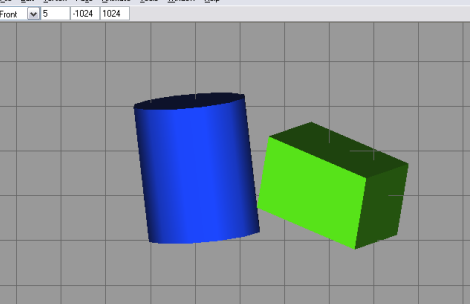
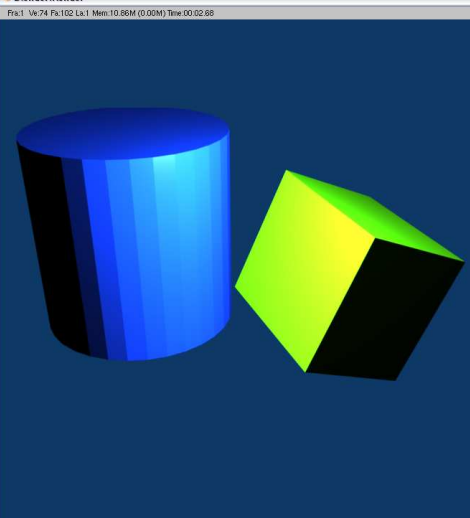
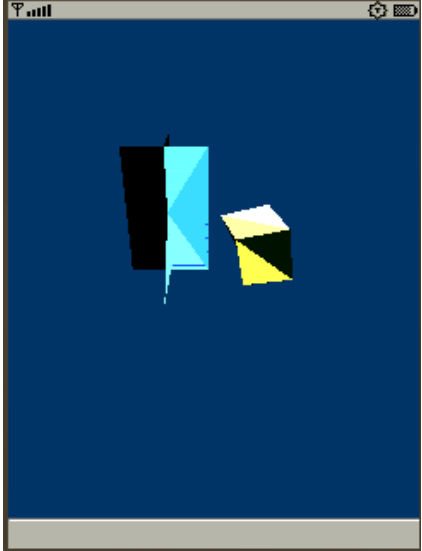
Maya eksportavimo
priemonė


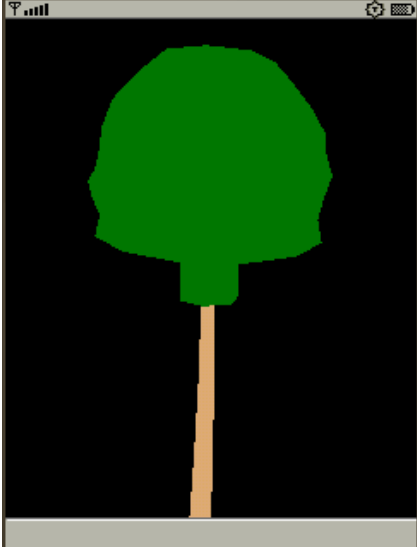
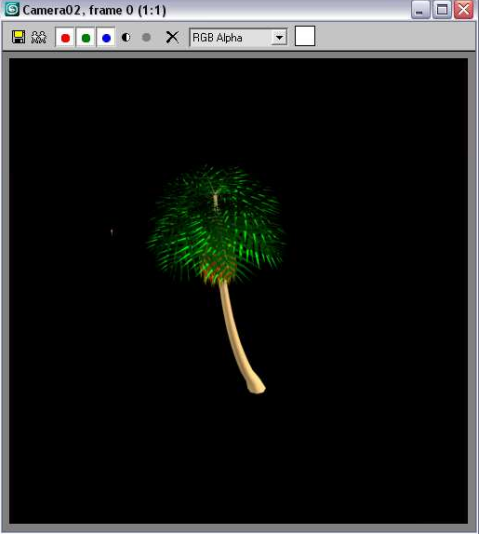



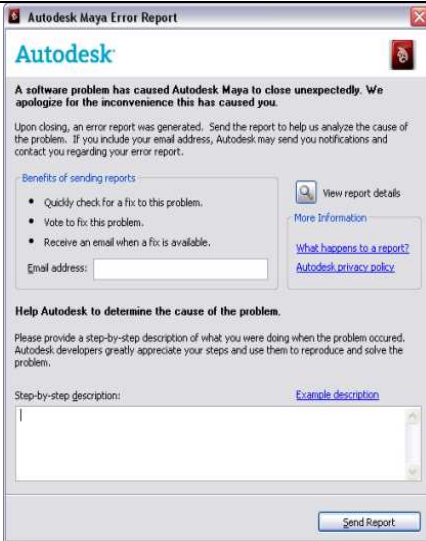


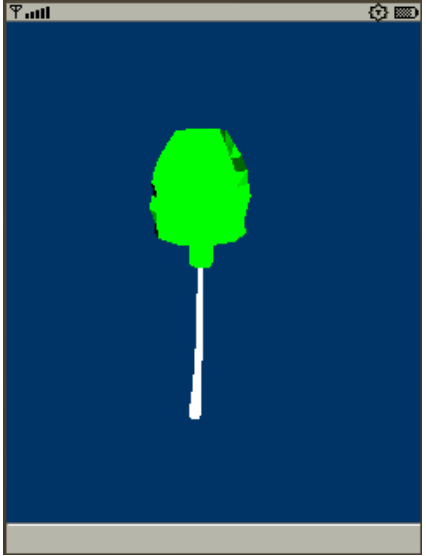
M3G failo dydis: 1,16 KB.

	<p>MilkShape3D eksportavimo priemonė</p>		 <p>M3G failo dydis: 479 baitai.</p>
	<p>Blender eksportavimo priemonė</p>		 <p>M3G failo dydis: 1,03 KB.</p>
<p>2.</p>	<p>Magistro darbo eksportavimo priemonė</p>		

			 <p>M3G failo dydis: 7.17 KB.</p>
	<p>Autodesk 3D Max eksportavimo priemonė</p>		 <p>M3G failo dydis: 6,79 KB.</p>

<p>Maya eksportavimo priemonė</p>		 <p>M3G failo dydis: 0 KB.</p>
<p>MilkShape3D eksportavimo priemonė</p>		<p>Nepalaiko kelių objektų eksportavimo.</p>
<p>Blender eksportavimo priemonė</p>		 <p>M3G failo dydis: 5,45 KB.</p>

<p>3.</p>	<p>Magistro darbo eksportavimo priemonė</p>		 <p>M3G failo dydis: 47,4 KB.</p>
	<p>Autodesk 3D Max eksportavimo priemonė</p>		 <p>M3G failo dydis: 26,6 KB.</p>

<p>Maya eksportavimo priemonė</p>	 <p>size: 640 480 zoom: 1.000 (Maya Software) Frame: 1 Render Time: 0:01 Camera: persp</p>	 <p>Autodesk A software problem has caused Autodesk Maya to close unexpectedly. We apologize for the inconvenience this has caused you. Upon closing, an error report was generated. Send the report to help us analyze the cause of the problem. If you include your email address, Autodesk may send you notifications and contact you regarding your error report.</p> <p>Benefits of sending reports</p> <ul style="list-style-type: none"> Quickly check for a fix to this problem. Vote to fix this problem. Receive an email when a fix is available. <p>Email address: <input type="text"/></p> <p>View report details More Information What happens to a report? Autodesk privacy policy</p> <p>Help Autodesk to determine the cause of the problem. Please provide a step-by-step description of what you were doing when the problem occurred. Autodesk developers greatly appreciate your steps and use them to reproduce and solve the problem.</p> <p>Step-by-step description: Example description</p> <p><input type="text"/></p> <p>Send Report</p>
<p>Milkshape3D eksportavimo priemonė</p>		<p>M3G failo dydis: 0 KB. Nepalaiko kelių objektų eksportavimo.</p>
<p>Blender eksportavimo priemonė</p>		 <p>M3G failo dydis: 43,6 KB.</p>

Eksportavimo rezultatų tyrimas parodė, kad patikimiausia ir daugiausiai galimybių turinti eksportavimo priemonė yra komercinė Autodesk 3D Max eksportavimo priemonė. Šios priemonės eksportavimo rezultatai atitiko originalius 3D modelius, objektų transformacijos eksportavimo metu nebuvo pakeistos, objektų, šviesų bei kamerų savybės po eksportavimo nepakito. Failo dydis net nenaudojant kompresijos buvo gana nedidelis. Eksportavimo priemonė turi savo vartotojo sąsają, kurioje medžio pavidalu pavaizduojami eksportuojamų objektų pavadinimai bei tipai. Prieš eksportavimą leidžiama pakeisti nustatymus, pvz. priskirtą objekto ID, ištrinti objektus ir t.t. Eksportavus duomenis į M3G failą, taip pat sukuriama ir eksportavimo informacijos failas, kuriame objekto vardas, tipas, unikalus ID numeris bei svoris kilobaitais. Klaidų pranešimai pateikiami atskirame lange. Tyrimo metu eksportavimo priemonei pavyko eksportuoti į M3G formatą 90 % patikimumu (2 kartus iš 20 eksportavimų priemonė nutraukė eksportavimą nepranešusi apie klaidas).

Mažiausiai patikima eksportavimo priemonė yra skirta Autodesk Maya modeliavimo priemonei. Ribotai eksportavimo priemonės versijai nepavyko teisingai eksportuoti net paprastų 3D modelių. Tyrimo metu eksportavus pirmąjį modelį, gautas rezultatas parodė, kad eksportavimo priemonė neteisingai eksportuoja šviesas (matyti iš juodo telefono ekrano). Bandant eksportuoti sudėtingesnius 3D modelius, eksportavimo priemonė „pakibdavo“ ir nutraukdavo eksportavimą nepranešusi apie atsiradusias klaidas.

MilkShape 3D eksportavimo priemonė nepalaiko kelių objektų eksportavimo. Eksportuotų objektų savybės bei transformacijos neatitinka pradinio 3D modelio. Eksportavimo priemonė yra patikima, tačiau ją galima naudoti tik paprastų 3D objektų eksportavimui į M3G formatą.

„Nelson Games“ siūloma nemokama eksportavimo priemonė skirta Blender modeliavimo paketui yra patikima, tačiau ji neturi nei vartotojo sąsajos, nei klaidų pranešimų valdymo, o visa minimali eksportavimo informacija yra pateikiama komandinėje eilutėje. Ši eksportavimo priemonė taip pat turi trūkumą - prieš eksportavimą viso 3D modelio mastelis yra pakeičiamas ir eksportavus 3D modelis yra sumažinamas, kad tilptų į emuliatoriaus ekraną. Eksportuojant sudėtingesnius 3D modelius objektų savybės bei transformacijos neatitinka pradinio 3D modelio.

10. Magistro darbo išvados

- M3G standartas yra kompaktiškas pasirenkamasis paketas 3D grafikos vaizdavimui mobiliuosiuose telefonuose. Nors pastaraisiais metais šis standartas tapo ypač populiarus, tačiau tyrimas parodė, kad iš paplitusių modeliavimo priemonių patikimiausia yra tik Autodesk 3D Max eksportavimo priemonė. Kitos gana populiaros eksportavimo priemonės į M3G formatą, skirtos MilkShape3D ir Autodesk Maya priemonės, turi eksportavimo trūkumų bei yra ribotos. Eksportuoti modeliai dažnai neatitinka originalaus 3D modelio, o eksportuojant net ir palyginti nesudėtingus trimačius objektus, eksportavimo priemonės gali „pakibti“ taip ir nepranešusios, kurioje vietoje yra klaida.
- Nors M3G 1.0 standarto versija buvo pasiūlyta daugiau nei prieš 3 metus, tačiau pakankamai aiškios dokumentacijos apie patį M3G failo formatą trūksta. JSR ekspertų pagrindinė dokumentacija labiau aprašo M3G standarto klasių modelį, metodus, manipuliavimą objektais mobiliosiose aplikacijose, tačiau trūksta informacijos apie patį M3G failo formatą bei objektų tvarką M3G faile. Dėl menkiausios klaidos objekto koordinatėse arba M3G failo formavimo struktūroje įkrovus eksportuotą M3G failą į Java emuliatorių – mobiliųjį telefoną, nepavyksta paleisti mobiliosios aplikacijos arba rodomas juodas emulioriaus ekranas.
- M3G 1.0 standartas turi griežtai apibrėžtą M3G failo struktūrą ir nesuteikia vartotojams manipuliavimo standartu galimybes. M3G 1.0 standarto versijoje duomenys gali būti nespaudžiami arba spaudžiami ZLib kompresija. Kiti kompresijos būdai nėra palaikomi bei nesuteikiama galimybė vartotojui juos pritaikyti. Dėl to net ir taikant kompresiją, sudėtingesni 3D modeliai yra ilgai kraunami aplikacijoje mobiliajame telefone, o tokių modelių animacijos stabdo aplikacijos veikimą. JSR ekspertai bando išspręsti aplikacijų užimamos vietos ir sudėtingų skaičiavimų kiekio problemą M3G 2.0 galutinėje versijoje. Pirmieji telefonai, turintys M3G 2.0 standarto versiją, pasirodys rinkoje 2009 m.
- Dauguma komercinių 3D modeliavimo priemonių viešai neskelbia būdų kaip prieiti prie modeliavimo priemonės modelių duomenų, todėl tai apsunkina savo eksportavimo priemonių į M3G formatą kūrimą bei integravimą į modeliavimo priemones.

Literatūros sąrašas

- [FFM01] File Format for Mobile 3D Graphics API
URL: <http://www.j2medev.com/api/jsr184/file-format.html>
- [JSR01] JSR-184 Expert Group. Mobile 3D Graphics API Technical Specification. 2005
- [KIL01] Killer Game Programming in Java
URL: <http://fivedots.coe.psu.ac.th/~ad/jg/>
- [EXP01] First Encounter with 3D Studio MAX version 7 and M3G exporter
URL: <http://www.benhui.net/modules.php?name=Mobile3D&page=First-Encounter-3DStudio-Exporter.html>
- [IBM01] 3D graphics for Java mobile devices, Part 1: M3G's immediate mode
URL: <http://www-128.ibm.com/developerworks/library/wi-mobile1/index.html>
- [IBM02] 3D graphics for Java mobile devices, Part 2: M3G's retained mode
URL: <http://www-128.ibm.com/developerworks/java/library/wi-mobile2/index.html>
- [JMS01] Blender API Python Introduction
URL: http://jmsoler.free.fr/didacticiel/blender/tutor/python_script00_en.htm
- [PYT01] Developer Documentation for the Python API
URL: <http://www.blender.org/documentation/pythonapi.pdf>
- [PYT02] Blender / Python scripting
URL:
http://jmsoler.free.fr/didacticiel/blender/tutor/python_script00_en.htm#Pour%20charger%20un%20script
- [WIK01] XML
URL: <http://lt.wikipedia.org/wiki/XML>
- [SCH01] XML tutorial
URL: <http://www.w3schools.com/xml/default.asp>
- [GRA01] 3D computer graphics software
URL: http://en.wikipedia.org/wiki/3D_computer_graphics_software
- [CAM01] Camera tutorial
URL: <http://argoslabs.com/~malefico/tutor/camera.html>
- [DOM01] XML DOM tutorial
URL: <http://www.w3schools.com/dom/default.asp>
- [TRE01] Creating Jtree component
URL: <http://java.sun.com/docs/books/tutorial/uiswing/components/tree.html>
- [M3G01] JSR 297: Mobile 3D Graphics API 2.0 specifications
URL: <http://jcp.org/en/jsr/detail?id=297>
- [MLK01] New M3G Exporter plug-in for Milkshape 3D by Sony Ericsson
URL:
http://developer.sonyericsson.com/site/global/newsandevents/latestnews/newsjan07/p_milkshape3d_exporter_tool.jsp
- [MLK02] Milkshape3D tool for creating 3D models
URL: <http://chumbalum.swissquake.ch/ms3d/index.html>
- [MAY01] M3G exporter for Maya
URL: <http://www.m3gexport.com/index.jsp>
- [MAY02] Autodesk Maya
URL: <http://usa.autodesk.com/adsk/servlet/index?id=7635018&siteID=123112>
- [MCP01] MascotCapsule Toolkit for M3G

- URL: <http://www.mascotcapsule.com/toolkit/m3g/en/index.php>
- [NEL01] Nelson Games exporter for Blender
URL: <http://www.nelson-games.de/bl2m3g/default.html>
- [AUT01] Autodesk 3ds Max
URL: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5659302>
- [JYT01] Jython project
URL: <http://www.jython.org/Project/index.html>
- [PJC01] Python & Java comparison
URL: http://www.ferg.org/projects/python_java_side-by-side.html

Priedai

1. Hierarchinis 3D duomenų medis XML faile.

