

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
MULTIMEDIJOS INŽINERIJOS KATEDRA

Audrius Bučinskas

**Miesto erdvinio vaizdo generavimas pagal  
topografinius duomenis**

Magistro darbas

Darbo vadovas  
dr. Antanas Šimaitis

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
MULTIMEDIJOS INŽINERIJOS KATEDRA

## **Miesto erdvinio vaizdo generavimas pagal topografinius duomenis**

**Magistro darbas**

Kalbos konsultantas

Lietuvių k. katedros lekt.

Jonas Jonušas

2006-05

Vadovas

dr. Antanas Šimaitis

2006-05

Recenzentas

doc.dr. K. Motiejūnas

2006-05

Atliko

IFM- 0/3 gr. stud.

Audrius Bučinskas

2006-05-18

Kaunas, 2006

# TURINYS

1. ĮVADAS	5
2. SISTEMOS ANALIZĖ	7
2.1. Tikslas	7
2.2. Programinės įrangos vieta tarp pasaulinių galiūnų	7
2.3. Rinkos programinės įrangos produktų analizė	7
2.4. Programinės įrangos kūrimo priemonės	10
2.5. Taikomųjų uždavinių modelis	12
2.6. Vartotojai	13
2.7. Projekto apribojimai	14
2.7.1. Įpareigojantys apribojimai	14
2.7.2. Sistemos kūrimo biudžetas	15
3. PROGRAMINĖS ĮRANGOS PROJEKTAS	16
3.1. Projekto tikslas ir sudėtis	16
3.2. Veiklos sudėtis	16
3.3. Veiklos padalinimas	17
3.4. Sistemos ir vartotojo galimybės	18
3.5. Sistemos ir vartotojo galimybių analizė	19
3.6. Funkciniai reikalavimai ir reikalavimai duomenims	24
3.6.1. Funkciniai reikalavimai	24
3.6.2. Reikalavimai duomenims	29
3.7. Reikalavimai sistemos išvaizdai	30
3.8. Sistemos panaudojamumo reikalavimai	30
3.9. Vykdomo charakteristikų reikalavimai	32
3.10. Veikimo sąlygų reikalavimai	33
3.11. Sistemos priežiūros reikalavimai	33
3.12. Sistemos saugumo reikalavimai	34
3.13. Kultūriniai-politiniai reikalavimai	34
3.14. Teisiniai reikalavimai	35
3.15. Projekto išeiga	36
3.15.1. Atviri klausimai	36
3.15.2. Egzistuojantys sprendimai	36

3.15.3. Naujos problemos	36
3.15.4. Uždaviniai	37
3.15.5. Vystymo etapai	37
3.16. Rizikos	37
3.16.1. Projekto rizikų aptarimas	37
3.16.2. Rizikų sprendimų planas	38
3.17. Vartotojo apmokymas	39
3.18. Programinės įrangos architektūra	40
3.18.1. Paskirtis	40
3.18.2. Apžvalga	40
3.18.3. Architektūros pateikimas	40
3.18.4. Architektūros tikslai ir apribojimai	41
3.18.5. Panaudojimo atvejai	41
3.18.6. Sistemos loginis vaizdas ir klasių detalizavimas	44
3.18.7. Procesu vaizdas	49
3.18.8. Bendradarbiavimo ir sekų diagramos	50
3.18.9. Sistemos išdėstymo vaizdas	51
3.18.10. Duomenų vaizdas	52
3.18.11. Vartotojo sąsaja	53
3.18.12. Testavimas	54
4. VARTOTOJO DOKUMENTACIJA	55
5. KOKYBĖS ĮVERTINIMAS	57
6. IŠVADOS	60
7. LITERATŪRA	61
8. TERMINŲ IR SANTRUMPŲ ŽODYNAS	62
9. SANTRAUKA ANGLŲ KALBA	63

# 1. ĮVADAS

Šiuo metu pasaulyje žmonės vis giliau skverbiasi į virtualią erdvę. Įvairiose srityse realūs eksperimentai, analizės yra atliekamos panaudojant sukurtas galingas virtualią erdvę pavaizduojančias sistemas. To priežastis – mažesni kaštai, sutaupytas laikas, sumažėjusi tarša. Virtualios sistemos neaplenkė ir tokių sričių kaip geodezija, architektūra. Iki šių dienų yra vis dar naudojami paprasti žemėlapiai, miesto maketai. Tačiau jiems „ant kulnų“ lipa virtualūs dvimačiai, trimačiai žemėlapiai su preciziškai atvaizduotais objektais, kurie bet kada realiame laike gali būti tikslinami ar modifikuojami. Tai labai patogu tiek geodezininkams, tiek architektams analizuojant naujų ar tiesiog esamų objektų įtaką kraštovaizdžiui, miesto planui. Besiplėtojančios informacinės technologijos rinkai pateikia visą eilę programinės įrangos, kurių galimybės iš tiesų yra nemažos. Dėl šių priemonių ne vienos įmonės ar įstaigos darbas tapo efektyvesnis, nes sumažėjo nuostoliai dėl padarytų klaidų ir sugaišto laiko jas sprendžiant. Dauguma šių „pagalbininkų“ sieja pasaulyje gerai žinomos ir plačiai naudojamos CAD sistemos vardas. Čia galima paminėti kiekvienam šiuolaikiniam architektui ar geodezininkui nesvetimas erdvinių vaizdų modeliavimo sistemas, skirtas Microsoft Windows (AutoCAD, ArchiCAD, GeoCAD, 3DS Max) ar Linux (LinuxCAD) terpei. Visos jos tinka atskirų objektų ar ištisų miesto vaizdų projektavimo ir generavimo darbams. Pasiiekti platų panaudojimo spektrą leidžia daugelis įrankių, kurie yra įdiegti šiose programinėse įrangose. Universalumas yra labai gera šių sistemų savybė, tačiau daugybė įrankių šiek tiek apsunkina darbą, todėl be rimto specialisto pagalbos galime neišsiversti. Tai gali būti aktualu tiek stiprioms didelį pelną gaunančioms įmonėms, tiek smulkesnėms, ypač pradedančioms įstaigoms, kurių pelnas yra nedidelis.

Taigi šio darbo esminis tikslas yra nesudėtinga, įvairiose operacinėse sistemose pritaikoma naudoti bei nebrangi trimačio miesto ar atskirų vaizdų generavimo sistema. Pagrindinis duomenų šaltinis - plačiai paplitusios projektavimo sistemos AutoCAD ar mažiau populiarios sistemos LinuxCAD \*.dxf failai. Dažnai turime topografinius duomenis, kuriuose matome tik iš surinktų pastatų, kelių ir reljefo duomenų nubraižytą dvimatį vaizdą, sudarytą iš elementarių viršūnių ir briaunų. Panaudojus specialiai įgytas AutoCAD ar kitos projektavimo programos žinias galima nubraižyti trimatį vaizdą, tačiau šio darbo rezultatas, t.y. programinė įranga, mums leis labai paprastai ir greitai sugeneruoti trimatį miesto vaizdą tik pasirinkus atitinkamus topografinius duomenis. Tai yra lyg pagalbinis priedas, skirtas minimalias projektavimo žinias turinčiam vartotojui.

Darbo metu buvo panaudota Java programavimo kalba su integruotomis OpenGL bibliotekomis. Ši kalba buvo pasirinkta dėl kelių priežasčių: pirma, jos kompiliatorius yra nemokamas; antra, parašyta programa gali būti nesunkiai modifikuojama ir pritaikoma įvairioms operacinėms sistemoms (Linux, Solaris). Darbo uždaviniai:

1. Apžvelgti pagrindinį kuriamos sistemos tikslą.
2. Paanalizuoti šiuo metu pasaulyje plačiai naudojamą projektavimo-modeliavimo programinę įrangą.
3. Išanalizuoti, kokios priemonės buvo panaudotos, kuriant trimačius vaizdus generuojančią programinę įrangą.
4. Apžvelgti dėl sistemos ir vartotojų poreikių atsirandančius projekto apribojimus.
5. Išskirti ir paanalizuoti funkcinius, kurie aprašo programinės įrangos funkcionalumą, t.y. kaip sistema reaguoja į vartotojo veiksmus, ir nefunkcinius, kurie apibrėžia sistemos savybes ir apribojimus, reikalavimus.
6. Įvertinti sistemos rizikas ir sudaryti jų sprendimo planą.
7. Sudaryti kuriamos programinės įrangos architektūrinį modelį, kuris padės iš arčiau pažvelgti į sistemos struktūrą bei atskleisti atskirų posistemių funkcionalumą.
8. Paanalizuoti sistemos duomenų srautus bei pateikti bendrą duomenų vaizdą, kuris atspindi kokio tipo duomenys yra naudojami viename ar kitame kuriamos programos etape.
9. Aptarti kuriamos programinės įrangos vartotojo sąsajos, kuri dažnam vartotojui yra tarsi „sistemos veidrodis“, ypatumus.
10. Aptarti sistemos testavimo, kuris padės sumažinti kuriamos sistemos klaidų skaičių, ypatumus.
11. Pateikti programinės įrangos vartotojo dokumentaciją, kuri leis vartotojui gana greitai įsisavinti sistemos veikimo bei darbo su ja esmę.

## 2. SISTEMOS ANALIZĖ

### 2.1 Tikslas

Paprasta ir nebrangi priemonė, skirta generuoti trimatį miesto vaizdą panaudojant topografinius duomenis. Programinės įrangos veikimo esmė:



1 pav. Tikslas realizacija

### 2.2 Programinės įrangos vieta tarp pasaulinių „galiūnų“

Žinoma, kad rinka yra užpildyta profesionalia tokių gigantų kaip Autodesk, Graphisoft produkcija. Apie ją pakalbėsime vėliau. Šio darbo programinė įranga ieškos savo nišos tarp neprofesionalių ar norinčių sutaupyti vartotojų. Be to, ją galima bus nesunkiai modifikuoti ir pritaikyti Linux terpei, o tai gali sudominti atitinkamos operacinės sistemos vartotojus, nes kolkas visas „galiūnų“ dėmesys sutelktas būtent tik Microsoft Windows terpei.

### 2.3 Rinkos programinės įrangos produktų analizė

Prieš pradėdant bet kokios programinės įrangos kūrimą svarbu išanalizuoti, kas jau yra sukurta. Apžvelgsiu kelias profesionalias programas, kurias naudojamos projektuoti, generuoti, modeliuoti trimačiams vaizdams.

#### **AutoCAD - Autodesk Inc.**

*Operacinė sistema:* Microsoft Windows 9x/NT/2000/ME/XP

*Kaina:* ~10000Lt

*Privalumai:* universali programinė įranga, skirta daugiau profesionaliems vartotojams.

Objektai kuriami iš grafinių primityvų: 2D – tiesės, apskritimai, elipsės, lankai ir kt.; 3D - kubai, sferos, cilindrai, piramidės ir kt. Braižymo ir projektavimo produktyvumas padidintas naudojant įvairius įrankius:

- Sheet Set Manager – brėžinių menedžeris, kuris valdo susijusių brėžinių rinkinius.
- Brėžinių rinkinyje greitas atidarymas, peržiūrėjimas.
- Brėžinių tvarkos organizavimas – grupavimas kategorijomis.
- Automatinis nuorodų ir ryšio sukūrimas tarp brėžinių ir vaizdų.
- Brėžinių rinkinių archyvavimas.
- OLE objektai – automatinis įterpimas ir orientavimas.

- CAD standartai – efektyvus fiksavimas ir atitikimo standartams pranešimas.

*Trūkumai:* gana aukšta kaina bei gana sudėtinga paprastam vartotojui

### **Autodesk MAP 3D - Autodesk Inc.**

*Operacinė sistema:* Microsoft Windows 9x/NT/2000/ME/XP

*Kaina:* ~13000Lt

*Privalumai:* naujos kartos kompanijos “Autodesk” produktas skirtas GIS duomenims kaupti, administruoti, naudoti ir analizuoti. Kadangi ši programinė įranga veikia kartu su nauja AutoCAD platforma, joje yra ir naujų šios programinės įrangos galimybių. Autodesk Map 3D skirta inžinieriams, vadybininkams, topografams ir kitų sričių specialistams kuriantiems ir valdantiems žemėlapius. Autodesk Map suteikia ir galimybę dirbti su 3D objektais – paviršiais. Paviršiai gali būti pavaizduojami izolinijomis, atliekama jų analizė ir vizualizacija nurodytame taške bei paimama ir pavaizduojama altitudė (elipsoidinis aukštis). Paviršius sukuriamas importuojant LandXML, DEM, TIN failus, naudojant taškus, taškų failus, lūžių linijas, esamus programos AutoCAD objektus, ribas. Taip pat sukuriamas paviršiaus atvaizdavimo stilius, kuriame yra nurodoma, kaip turi būti pavaizduojamas paviršius. Keičiant paviršiaus savybes pasirenkamas paviršiaus pavaizdavimo stilius ir atsižvelgiant į stiliaus nustatymus paviršius gali būti pavaizduojamas kitaip.

*Trūkumai:* gana aukšta kaina bei gana sudėtinga paprastam vartotojui.

### **Autodesk VIZ - Autodesk Inc.**

*Operacinė sistema:* Microsoft Windows 9x/NT/2000/ME/XP

*Kaina:* ~6000Lt

*Privalumai:* programa skirta trimačiam modeliavimui ir vizualizacijai. Programoje yra pateikiami visų tipų architektūrinių statinių conceptualaus projektavimo instrumentai, įskaitant ir visiškai paprastą aplinkos reljefo modeliavimą vietovės skaitmeninių ir foto duomenų importo būdu. Didelė landšafto objektų: medžių, krūmų ir pan. biblioteka. Šie objektai turi tokius parametrus kaip aukštis, šakų ir lapų kiekis, vėjo apkrovos ir svyravimai. Taip pat yra didelė parametrinių statybinių elementų (sienų, langų, durų, laiptų ir t.t.) biblioteka, kuri suteikia galimybę nepaprastai lengvai modeliuoti eksterjerus ir interjerus. Būdingas trimačių modelių kūrimas dvimačių brėžinių pagrindu. Tam programa turi visus reikiamus instrumentus, leidžiančius uždarus ir atvirus kontūrus pertvarkyti į trimačius objektus – sienas, langus, duris, laiptus ir kt. Programa dar turi vizualizacijos funkcijas, kurios padeda greitai ir kokybiškai vizualizuoti projektą.



*Trūkumai:* gana aukšta kaina bei gana sudėtinga paprastam vartotojui.

### **ArchiCAD - GRAPHISOFT**

*Operacinė sistema:* Microsoft Windows 9x/NT/2000/ME/XP

*Kaina:* ~15000Lt

*Privalumai:* programa kuriama specialiai architektams. Dirbdant ArchiCAD'u automatiškai kuriamas virtualaus pastato ar kito objekto modelis: vietoj linijų, kreivių ar apskritimų generuojamos sienos, dedami langai ir durys, klojamos grindys, konstruojami stogai ir pan. ArchiCAD generuoja 3D modeliui reikalingą informaciją, pagal kurią bet kuriame projektavimo etape sukuria pjūvius, fasadus, medžiagų sąrašus, specifikacijas ir t.t. ArchiCAD dirba su daugybe paplitusių failų formatų - DXF, DWG, DGN, PDF, IFC, SVF, DWF, HPGL, WMF, PICT, PMK, VRML, 3DStudio, OBJ (Wavefront), Art•lantis Render, Lightscape Technology, FACT (Electric Image), 3DMF, ADS 3D, ZOOM, Strata, ASCII, TEXT, RTF, GDL, JPEG, TIF, GIF, BMP, BMS, PICS, QuickTime VR, RealVR, QuickTime Mov, AVI, FLI (Animator Pro).

*Trūkumai:* gana aukšta kaina bei gana sudėtinga paprastam vartotojui.

### **3D Studio Max – Autodesk Inc.**

*Operacinė sistema:* Microsoft Windows 9x/NT/2000/ME/XP

*Kaina:* ~11000 Lt

*Privalumai:* tai labiausiai pasaulyje parduodama profesionalaus trimačio modeliavimo, animacijos ir vaizdų generavimo programinė įranga, kuria galima kurti vizualinius efektus, animuoti personažus ir kurti naujos kartos žaidimus. 3DS Max leidžia dirbti visiškai integruotoje trimatėje erdvėje ir atlikti ypač greitą interaktyvų vaizdų generavimą. Ši programa turi daug objektų modifikavimo, transformavimo priemonių, kurios palengvina tiek naujų objektų sukūrimą, tiek jau sukurtų objektų modifikavimą. Taip pat šis produktas turi daug priemonių, kurios padeda sukurti įvairius aplinkos efektus: rūką, sniegą ir t.t.

*Trūkumai:* aukšta kaina bei gana sudėtinga paprastam vartotojui.

### **ArCon - mb Software AG**

*Operacinė sistema:* Microsoft Windows 9x/NT/2000/ME/XP

*Kaina:* 4000 Lt

*Privalumai:* ši programa suteikia dideles galimybes architektams. Programa leidžia bet kuriame darbo etape matyti projekto trimatį modelį, jo pjūvį ar perspektyvą, parinkti

tinkamiausias medžiagas ir suskaičiuoti jų išėigą. Architektas ar dizaineris gali lengvai sukurti multiplikacinę filmuką, pavaizduodamas pastatą konkrečioje vietovėje.

*Trūkumai:* aukšta kaina.

### **LinuxCAD - Software Forge Inc.**

*Operacinė sistema:* Linux (bet kuri versija)

*Kaina:* ~ 500 Lt

*Privalumai:* šioje programoje realizuotos pagrindinės AutoCAD 2D ir 3D projektavimo komandos. Palaiko pagrindinius AutoCAD formatus (\*.dwg, \*.dxf), todėl galima failų tarp skirtingų operacinių sistemų migracija. Be to, programinės įrangos kaina yra artimesnė paprastam vartotojui.

*Trūkumai:* didelių trūkumų, ypač naujose versijose, nepastebėta.

Taigi pastebime, kad dauguma programinės įrangos pavyzdžių yra tikrai kokybiški ir profesionalūs. Tačiau dėl jų beribio funkcionalumo ir universalumo išauga kaina.

Atsižvelgiant į tai, šiame darbe sukūrėme paprastą ir nebrangų produktą, kuris realiu laiku generuoja trimatį vaizdą panaudodamas dvimačius topografinius duomenis.

## **2.4 Programinė įrangos kūrimo priemonės**

Kuriant programą buvo naudojama Java programavimo kalba. Kaip jau minėjau, ši kalba pasirinkta dėl to, kad būtų sumažinti programinės įrangos kūrimo kaštai, nes Java kūrėjai nemokamai leidžia naudotis šios kalbos kompiliatoriumi ir transliatoriumi. Bet tai nėra vienintelis teigiamas aspektas. Java platforma „pralaužia“ barjerą tarp skirtingų operacinių sistemų. Todėl šio darbo programos kodas ateityje gali būti pritaikomas tiek Windows, Linux, Solaris ar kitai terpei. Tai suteiks programai universalumo ir lankstumo. Vartotojas nėra priverstinai pririšamas prie vienos operacinės sistemos.

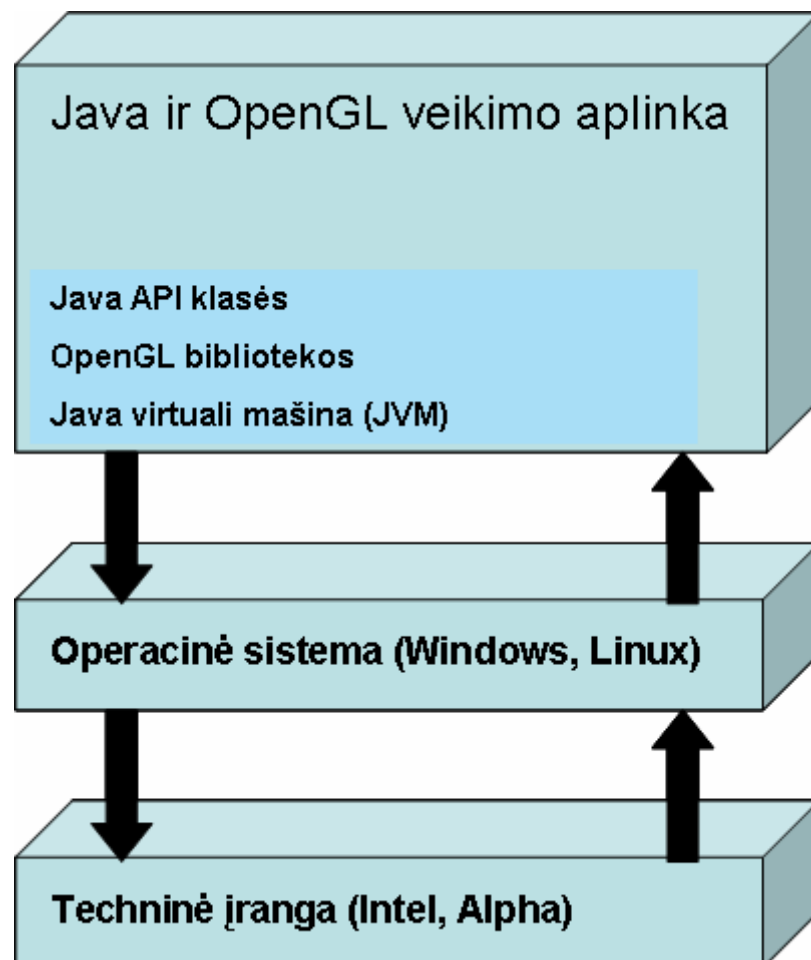
Kuriant trimačius vaizdus generuojančią programinę įrangą taip pat labai svarbu pasirinkti tinkamą trimatės grafikos apdorojimo standartą. Prieš atliekant praktinę šio darbo dalį buvo analizuojami du grafiniai standartai: Java 3D ir OpenGL. Bandytu metu paaiškėjo tai, kad programinė įranga parašyta panaudojant Java 3D bibliotekas reikalauja žymiai didesnių kompiuterio resursų nei panaudojant OpenGL bibliotekas. Todėl ir buvo pasirinktas OpenGL grafikos standartas. Trumpai apžvelgsime OpenGL (Open Graphics Library) – atvirą grafinę biblioteką. Programos, sukurtos panaudojant šį standartą, taip pat nėra pririštos prie vienokios ar kitokios platformos. Jos turinį sudaro visa eilė įrankių, t.y. apie 150 komandų,

leidžiančių programuotojui naudojant primitivus (taškus, linijas, daugiakampius) generuoti trimačius vaizdus. Pagrindinės savybės:

- grafiniai (linijos, poligonai) ir rastriniai primitivai;
- B – splainų panaudojimas;
- darbas su spalvomis;
- Z – buferizacija;
- darbas su tekstūromis;
- apšvietimo panaudojimas;
- atmosferių efektų panaudojimas.

O šiame darbe panaudodami grafinius primitivus, tekstūras, apšvietimą sukursime trimačius vaizdus generuojančią programinę įrangą.

Taigi bendrą Java ir OpenGL veikimą galima pavaizduoti tokia schema:



2 pav. Java ir OpenGL veikimas

## 2.5 Taikomųjų uždavinių modelis

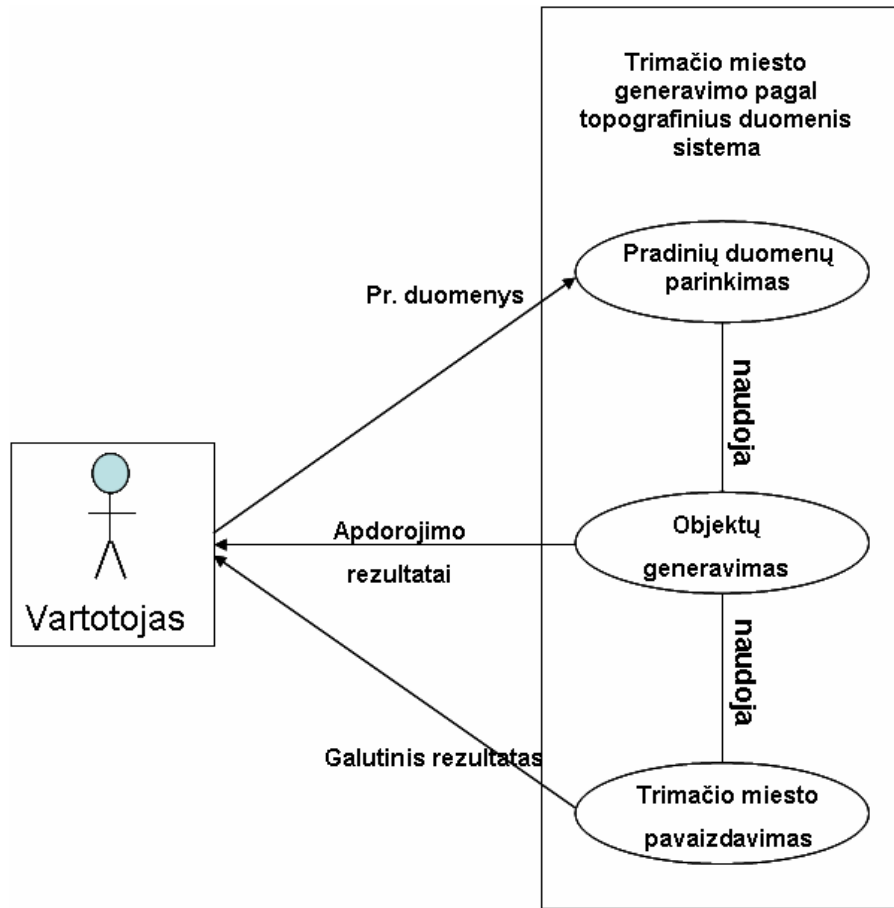
Paanalizuosime kuriamos sistemos bei vartotojo sąveiką, informacinius srautus ir visą tai pavaizduosime Use Case diagrama.

**1 lentelė.** Taikomųjų uždavinių modelio „Trimačio miesto generavimas panaudojant topografinius duomenis“ veiklos dalyviai.

Veiklos dalyvis	Veiklos dalyvio aprašymas
Vartotojas	Asmuo, naudojantis trimačio miesto generavimo sistema. Tai gali būti architektas, geodezininkas ar bet kurios kitos srities žmogus. Sistema skirta neprofesionaliam vartotojui. Kalbant apie neprofesionalumą, turima omenyje tai, kad vartotojui reikia turėti bent minimalių braižymo žinių.
Trimačio miesto generavimo panaudojant topografinius duomenis sistema	Tai produktas, veikiantis Microsoft Windows aplinkoje. Tačiau jį modifikavus, galima pritaikyti ir darbui Linux terpėje. Iš vartotojo gavusi atitinkamus duomenis ir komandą sistema vykdo apskaičiavimus ir trimačio miesto vaizdo generavimą.

**2 lentelė.** „Trimačio miesto generavimas panaudojant topografinius duomenis“ modelio taikomųjų uždavinių aprašymas

Uždavinys	Aprašymas
Pradinių duomenų parinkimas	Vartotojas pasirenka *.dxf failuose saugomus pradinius duomenis(apie reljefą, pastatus, kelius) ir pateikia juos sistemai.
Objektų generavimas	Pagal pateiktus duomenis sugeneruojami trimačiai pastatai, keliai ir reljefas.
Trimačio miesto pavaizdavimas	Sugeneruoti objektai sujungiami į bendrą vaizdą, t.y. trimatį miestą, kuris ir pateikiamas vartotojui.



3 pav. Taikomųjų uždavinių modelis

**3 lentelė.** „Trimačio miesto generavimo pagal topografinius duomenis“ taikomųjų uždavinių informacinių srautų aprašymas

Informacinis srautas	Aprašymas
Pr. duomenys	*.dxf failuose išsaugoti dvimačiai duomenys.
Apdorojimo rezultatai	*.txt faile sugrupuoti duomenys, skirti generuoti grafinius elementus.
Galutinis rezultatas	trimatis miesto vaizdas.

## 2.6 Vartotojai

Trumpai apžvelgsime, kam bus skirtas šio darbo produktas.

**4 lentelė.** Programinės įrangos vartotojai

Vartotojo kategorija:	Geodezininkai, architektai, projektuojančios įmonės, miesto kraštovaizdžio analitikai
Vartotojo sprendžiami uždaviniai:	Dvimačių duomenų panaudojimas generuojant trimatį miesto modelį.
Patirtis dalykinėje srityje:	Patyręs

Patirtis informacinėse technologijose:	Pagrindai
Papildomos vartotojo charakteristikos:	Vartotojas turi būti minimaliai susipažinęs ir dirbęs su CAD projektavimo-modeliavimo sistemomis (AutoCAD pagrindai).

## 2.7. Projekto apribojimai

### 2.7.1. Įpareigojantys apribojimai

#### *Kompiuterio architektūros apribojimai*

Programinė įranga turi veikti nepriklausomai nuo kompiuteryje įdiegtos platformos (operacinės sistemos). Be to, programa turi veikti tokiuose kompiuteriuose, kurių architektūros charakteristikos neprastesnės už toliau išvardintas:

Procesoriaus dažnis	ne mažesnis negu 1500 MHz.
Operacinė sistema	MS Windows 98/2000/XP.
Atmintis	ne nemažesnė negu 512 MB.
Ekranas	Spalvotas ir raiška ne mažesnė negu 1024*768, 32 bitų spalvos.
Spausdintuvas	Galimybė spausdinti didesnio formato lapus, patartina lazerinis.

#### *Diegimo aplinka*

Kadangi produktas kuriamas Java kalbos pagrindu, reikalingas kompiuteris, kuriame būtų įdiegta naujausia Java virtuali mašina (JVM).

#### *Bendradarbiaujančios sistemos*

Pati sistema nebendradarbiauja su jokia kita sistema. Nebent vartotojas, norėdamas pakoreguoti \*.dxf duomenų failus, gali pasinaudoti atitinkama redagavimo priemone (AutoCAD, LinuxCAD, Notepad).

#### *Numatoma darbo vietos aplinka*

Numatomai darbo vietai specialių reikalavimų nėra. Fizinės darbo vietos charakteristikos atitinka elementarios darbo su kompiuteriu vietos charakteristikas.

#### *Sistemos kūrimo terminai*

### 5 lentelė. Sistemos kūrimo terminai

Eil. Nr.	Produktas	Terminas
1.	Reikalavimų specifikacijos	2005 m. Rugsėjis
2.	Projektavimas	2005 m. Gruodis
3.	Projekto atskirų komponentų realizavimas	2006 m. Vasaris
4.	Komponenčių sujungimas į visumą	2006 m. Kovas
4.	Projekto testavimas	2006 m. Balandis.
5.	Programinės įrangos diegimas	2006 m. Gegužė.

### **2.7.2. Sistemos kūrimo biudžetas**

Projektui reikalingi ištekliai ir finansavimas dar nėra numatyti.

### 3. PROGRAMINĖS ĮRANGOS PROJEKTAS

#### 3.1. Projekto tikslas ir sudėtis

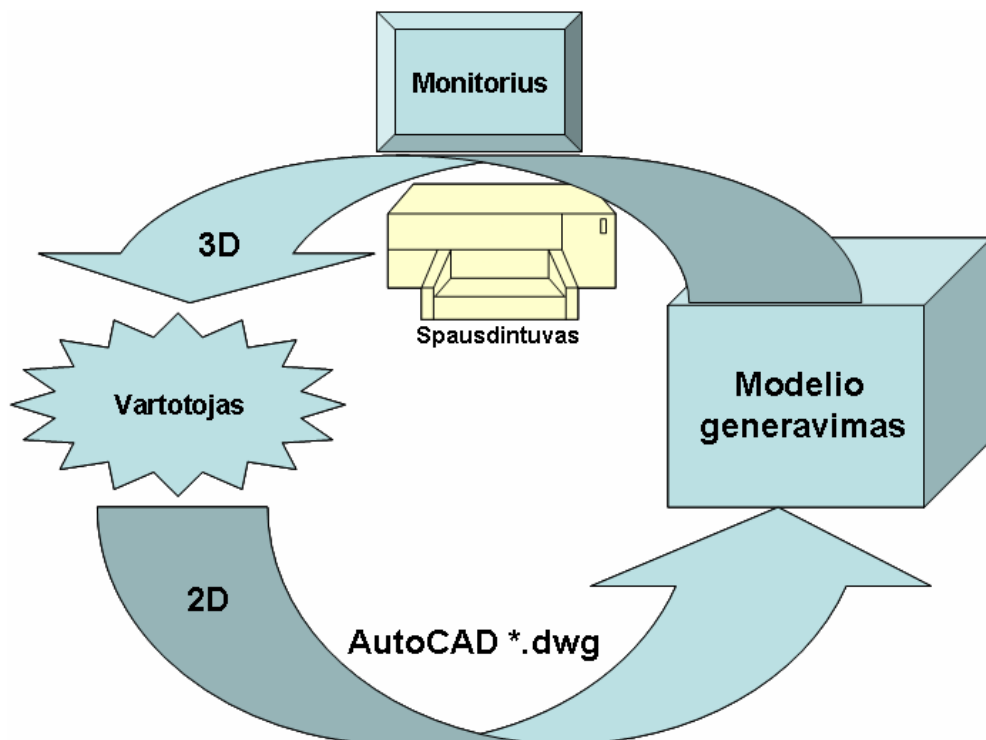
Sukursime paprastą, nebrangią, trimačiams miesto vaizdams generuoti skirtą programinę įrangą. Vartotojui reikės tik pasirinkti reljefo, pastatų ir kelių topografinius duomenis.

Be to, pasistengsime išanalizuoti programos kūrimo ypatumus, visos sistemos ir atskirų posistemų įtaką programinės įrangos funkcionalumui. Visa tai, padės geriau suprasti trimačio miesto generavimo esmę. Pagrindiniai šios dalies etapai:

- sistemos veiklos apibendrinimas;
- funkcinų reikalavimų išgavimas;
- reikalavimų duomenims apibendrinimas;
- nefunkcinių reikalavimų aptarimas;
- atvirų klausimų, problemų, rizikų išskyrimas;
- programinės įrangos architektūros analizė bei detali specifikacija;

#### 3.2. Veiklos sudėtis

Pažvelgsime į projektą truputį išsamiau ir vaizdingiau. Jo veikimo esmę galime pavaizduoti taip:



4 pav. Veiklos sudėtis



### 3.3. Veiklos padalinimas

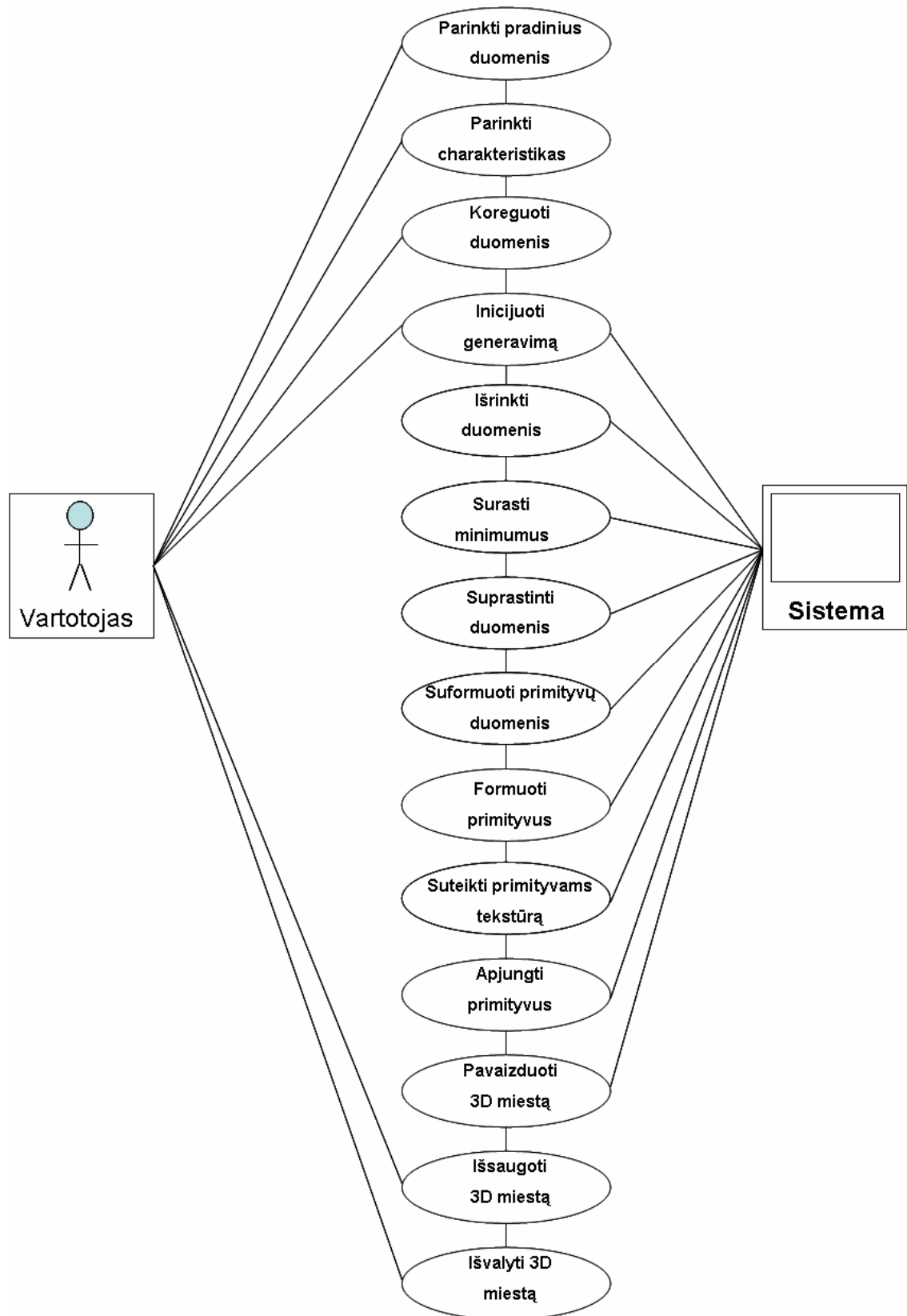
Vistik programinė įranga susideda ne vien tik iš duomenų pateikimo ir rezultatų grąžinimo. Yra dar keletas labai svarbių tarpinių stadijų, kurios paprastam vartotojui gal ir nėra tokios svarbios, tačiau programuotojas be jų negalėtų suprasti programos veikimo esmės.

Informaciją apie jas surasime sekančioje lentelėje:

**6 lentelė.** Programinės įrangos veiklos stadijos.

Stad. Nr.	Stadijos aprašymas	Įeinantys(in)/išeinantys(out) informacijos srautai
1.	Vartotojas pasirenka ir perduoda *.dxf failuose esančius topografinius duomenis. Informacija apie pastatus, kelius ir reljefą yra skirtinguose failuose.	Pradiniai duomenys (in)
2.	Vartotojas pasirenka kelių, pastatų ir reljefo tekstūras.	Vaizdo parametrai (in)
3.	Vartotojas koreguoja pradinius duomenis. Tai jis gali atlikti su tam skirta redagavimo priemone (LinuxCad, AutoCAD), arba tiesiog Notepad'o pagalba pakeisdamas koordinates ar kitas charakteristikas (pastato tipą, aukštį).	Koreguoti parametrai (in)
4.	Sistema nuskaito vartotojo pateiktus pradinius duomenis, juos patikrina ir apdoroja. Apdorojimas susideda iš smulkesnių etapų: išrinkimas, minimumų paieška, prastinimas, grupavimas ir kt. Šioje stadijoje netvarkingi, išmėtyti duomenys paruošiami ir pateikiami tiek vartotojui, tiek generavimo sistemai patogesne forma.	Darbiniai duomenys (out)
5.	Sistema peržiūri pilnai sutvarkytus duomenis, vartotojo parinktus vaizdo bei koregavimo parametrus, sugeneruoja grafinius primityvus, padengia juos tekstūromis ir tik tuomet viską apjungia į vieną trimatį miesto vaizdą (sistemą) ir pavaizduoja gautą rezultatą vaizduoklio ekrane arba atspausdina ant popieriaus.	Sugeneruotas vaizdas (out)

### 3.4. Sistemos ir vartotojo galimybės



5 pav. Sistemos ir vartotojo galimybės

### 3.5. Sistemos ir vartotojo galimybių analizė

7 lentelė. Vartotojo galimybė „Parinkti pradinis duomenis“

Nr.	1
Pavadinimas:	Parinkti pradinis duomenis
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Vartotojas pasirenka ir perduoda sistemai *.dxf failuose esančius topografinius duomenis.
Prieš sąlyga:	Vartotojas turi turėti paruoštus topografinius duomenis.
Sužadinimo sąlyga:	Vartotojas pasirenka duomenų failus.
Po sąlyga:	Parinkti duomenų failai yra pateikiami trimačių vaizdų apdorojimo sistemai.

8 lentelė. Vartotojo galimybė „Parinkti charakteristikas“

Nr.	2
Pavadinimas:	Parinkti charakteristikas
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Vartotojas savo nuožiūra pasirenka vaizdines charakteristikas, t.y. parenka kiekvienam objektui (reljefas, keliai, pastatai) tekstūrą.
Prieš sąlyga:	Vartotojas turi turėti pasiruošęs atitinkamų objektų tekstūrų failus.
Sužadinimo sąlyga:	Vartotojas pasirenka miesto objektų tekstūroms reikalingus failus.
Po sąlyga:	Parinkti tekstūrų failai yra pateikiami trimačių vaizdų apdorojimo sistemai.

9 lentelė. Vartotojo galimybė „Koreguoti duomenis“

Nr.	3
Pavadinimas:	Koreguoti duomenis
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Vartotojas koreguoja tik pradinis duomenis, t.y. keičia koordinatas ar kitas charakteristikas (pastato tipą, aukštį, sienų ilgį ir pan.). Tai jis gali atlikti su tam skirta specialia redagavimo priemone (LinuxCad, AutoCAD), arba tiesiog Notepad'o pagalba.
Prieš sąlyga:	Vartotojas turi būti pasiruošęs atitinkamus pradinis duomenų failus.
Sužadinimo sąlyga:	Vartotojas panaudodamas atitinkamas redagavimo priemones koreguoja pradinis duomenis.
Po sąlyga:	Koreguoti pradinis duomenys vėl pateikiami trimačių vaizdų generavimo sistemai.

**10 lentelė.** Vartotojo galimybė „Inicijuoti generavimą“

Nr.	4
Pavadinimas:	Inicijuoti generavimą
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Vartotojas, parinkęs naujus ar koreguotus duomenis, taip pat tekstūras ir nustatęs kitas charakteristikas, inicijuoja generavimą. Sistema pradeda savo darbą.
Prieš sąlyga:	Vartotojas turi būti pasirinkęs pradinis duomenis bei objektų tekstūras.
Sužadinimo sąlyga:	Vartotojas nuspaudžia funkcijos, kuri inicijuoja generavimą mygtuką.
Po sąlyga:	Sistema pradeda darbą.

**11 lentelė.** Sistemos galimybė „Išrinkti duomenis“

Nr.	5
Pavadinimas:	Išrinkti duomenis
Vartotojo pavadinimas:	Sistema
Aprašas:	Sistema, gavusi iš vartotojo pradinis duomenis, vykdo išrinkimo procesą. Pradiniai duomenys išrenkami ir sutalpunami į masyvų sąrašus.
Prieš sąlyga:	Sistema iš vartotojo turi gauti naujus ar koreguotus pradinis duomenis.
Sužadinimo sąlyga:	Kai tik gaunami duomenys ir inicijuojamas generavimas iš karto pradamas išrinkimo procesas.
Po sąlyga:	Sistema po išrinkimo turi visą eilę neapdorotų duomenų, kurie yra sutalpinti į masyvų sąrašus.

**12 lentelė.** Sistemos galimybė „Surasti minimumus“

Nr.	6
Pavadinimas:	Surasti minimumus
Vartotojo pavadinimas:	Sistema
Aprašas:	Sistema, gavusi atskirų objektų (pastatų, kelių, reljefo) pradinis atskiruose masyvų sąrašuose sutalpintus duomenis, juos analizuoja ir ieško minimalių reikšmių.
Prieš sąlyga:	Sistema turi gauti bent vieno objekto išrinktų duomenų masyvo sąrašą.
Sužadinimo sąlyga:	Kai tik gaunami išrinktų duomenų masyvų sąrašai pradama koordinačių minimumo paieška.
Po sąlyga:	Sistema turi koordinačių minimumus.

**13 lentelė.** Sistemos galimybė „Suprastinti duomenis“

Nr.	7
Pavadinimas:	Suprastinti duomenis
Vartotojo pavadinimas:	Sistema
Aprašas:	Sistema, radusi koordinačių minimumus, atlieka koordinačių prastinimo veiksmus, o tada naujos jau suprastintos koordinatės vėl sutalpinamos į masyvų sąrašus.
Prieš sąlyga:	Sistema turi gauti pradinius nekoreguotus duomenis bei koordinačių minimumus.
Sužadinimo sąlyga:	Sistema gauna koordinačių minimumus.
Po sąlyga:	Po apdorojimo sistema turi naujus jau apdorotus taškų koordinačių duomenis.

**14 lentelė.** Sistemos galimybė „Formuoti primityvų duomenis“

Nr.	8
Pavadinimas:	Formuoti primityvų duomenis
Vartotojo pavadinimas:	Sistema
Aprašas:	Sistema, gavusi jau pakeruotus duomenis, analizuoja ir formuoja grafinių primityvų duomenis. Kiekvienam objektui atskirai sistema iš gautų duomenų suformuoja primityvų duomenis ir talpina į kiekvienam objektui atskirai (pastatai, reljefas, keliai) skirtą masyvą.
Prieš sąlyga:	Sistema turi gauti jau pilnai apdorotus kiekvieno objekto duomenis.
Sužadinimo sąlyga:	Kai tik gaunami duomenys iš karto ir pradamas primityvų duomenų formavimo procesas.
Po sąlyga:	Po formavimo sistema turi visų objektų grafinių primityvų duomenis, sutalpintus į kiekvienam objektui priklausančius masyvų sąrašus.

**15 lentelė.** Sistemos galimybė „Generuoti primityvus“

Nr.	9
Pavadinimas:	Generuoti primityvus
Vartotojo pavadinimas:	Sistema
Aprašas:	Sistema, gavusi suformuotus primityvų duomenis, pereina į grafinių primityvų generavimą. Kiekvienam objektui atskirai yra generuojami daugiakampiai.
Prieš sąlyga:	Sistema turi gauti suformuotus primityvų duomenis.
Sužadinimo sąlyga:	Kai tik gaunami suformuoti primityvų duomenys iš karto ir pradamas generavimo procesas.
Po sąlyga:	Po generavimo sistema turi visą eilę daugiakampių, kurie vėliau sudarys trimatį miesto vaizdą.

**16 lentelė.** Sistemos galimybė „Suteikti primityvams tekstūrą“

Nr.	10
Pavadinimas:	Suteikti primityvams tekstūrą
Vartotojo pavadinimas:	Sistema
Aprašas:	Sistema jau turi visą eilę sugeneruotų daugiakampių. Tačiau visi jie yra vienspalviai. Tad sistema, panaudodama vartotojo tekstūras, padengia jomis daugiakampius.
Prieš sąlyga:	Sistema turi turėti sugeneruotus daugiakampius ir vartotojo parinktas tekstūras.
Sužadinimo sąlyga:	Jei buvo sugeneruoti daugiakampiai, sistema iš karto pradeda jų dengimą tekstūromis.
Po sąlyga:	Po generavimo sistema turi visą eilę tekstūromis padengtų atskirų daugiakampių.

**17 lentelė.** Sistemos galimybė „Apjungti primityvus“

Nr.	11
Pavadinimas:	Apjungti primityvus
Vartotojo pavadinimas:	Sistema
Aprašas:	Sistema sugeneruotus ir padengtus tekstūromis daugiakampius apjungia. Daugiakampių visuma sudaro pastatus, reljefą ir kelius.
Prieš sąlyga:	Sistema turi gauti sugeneruotus ir padengtus tekstūromis daugiakampius.
Sužadinimo sąlyga:	Pakanka gauti sugeneruotus daugiakampius ir sistema pradeda vykdyti jų apjungimą. Daugiakampiai gali būti ir be tekstūrų.
Po sąlyga:	Po daugiakampių apjungimo sistema turi jau sugeneruotą bet dar neatvaizduotą trimatį miesto vaizdą.

**18 lentelė.** Sistemos galimybė „Pavaizduoti 3D miestą“

Nr.	12
Pavadinimas:	Pavaizduoti 3D miestą
Vartotojo pavadinimas:	Sistema
Aprašas:	Sistema, sugeneravusi ir apjungusi daugiakampius, atvaizduoja juos vaizduoklio ekrane. Vartotojui pateikiamas erdvinis miesto vaizdas
Prieš sąlyga:	Sistema turi turėti sugeneruotus ir apjungtus daugiakampius.
Sužadinimo sąlyga:	Trimatis vaizdas vaizduoklio ekrane pavaizduojamas automatiškai tik sugeneravus ir apjungus daugiakampius.
Po sąlyga:	Pavaizdavus trimatį miesto vaizdą, vartotojas jį mato vaizduoklio ekrane.

**19 lentelė.** Vartotojo galimybė „Išsaugoti 3D miestą“

Nr.	13
Pavadinimas:	Išsaugoti 3D miestą
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Kuomet sistema sugeneruoja trimatį miesto vaizdą, vartotojas gali jį išsaugoti paveikslėlio pavidalu.
Prieš sąlyga:	Sistema turi būti sugeneravusi trimatį vaizdą.
Sužadinimo sąlyga:	Išsaugojimo funkciją vartotojas iškviečia meniu pasirinkdamas atitinkamą funkciją.
Po sąlyga:	Turime išsaugotą trimatį miesto vaizdą.

**20 lentelė.** Vartotojo galimybė „Išvalyti 3D miestą“

Nr.	14
Pavadinimas:	Išvalyti 3D miestą
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Jeigu vartotojas, panaudodamas naujus duomenis, nori sugeneruoti naują trimatį vaizdą, tuomet jis turi panaikinti dar prieš tai sugeneruotą vaizdą.
Prieš sąlyga:	Sistema turi būti sugeneravusi trimatį vaizdą.
Sužadinimo sąlyga:	Išvalymo funkciją vartotojas iškviečia meniu išsirinkdamas atitinkamą funkciją.
Po sąlyga:	Turime išvalytą tuščią langą, kuriame vartotojas gali sugeneruoti kitą trimatį vaizdą.

### 3.6. Funkciniai reikalavimai ir reikalavimai duomenims

#### 3.6.1. Funkciniai reikalavimai

Reikalavimas #:	1	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 1
Aprašymas:	Galimybė vartotojui pasirinkti pradinis duomenis. Duomenų pasirinkimo langas išskviečiamas nuspaudus atitinkamą meniu mygtuką.		
Pagrindimas:	Galimybė vartotojui pasirinkti *.dxf failuose, esančiu pradinis duomenis ir pateikti juos sistemai.		
Šaltinis:	Vartotojas		
Tikimo kriterijus:	Tuo pačiu būti pradėtas naujo vaizdo generavimo procesas.		
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2005 liepos 23.		

Reikalavimas #:	2	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 2
Aprašymas:	Vartotojas gali pasirinkti kiekvienam objektui(pastatams, keliams, reljefui) tinkamą tekstūrą. Tekstūrų pasirinkimo langas išskviečiamas nuspaudus atitinkamą meniu tipo mygtuką.		
Pagrindimas:	Vartotojas gali pasirinkti kokius nori *.png tipo failus.		
Šaltinis:	Vartotojas		
Tikimo kriterijus:	Vartotojas gali keisti tekstūras, kurios naudojamos generuoti vaizdus.		
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2005 liepos 26.		

Reikalavimas #:	3	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 3
Aprašymas:	Vartotojas gali koreguoti tik pradinis duomenis, t.y. keisti koordinatas ar kitas charakteristikas (pastato tipą, aukštį, sienų ilgį ir		



	pan.). Tai jis gali atlikti su tam skirta specialia redagavimo priemone (LinuxCad, AutoCAD), arba tiesiog Notepad'o pagalba.
Pagrindimas:	Vartotojas gali koreguoti *.dxf tipo failus.
Šaltinis:	Vartotojas
Tikimo kriterijus:	Yra galimybė koreguojant pradiniu duomenis, keisti erdvinio miesto vaizdą.
Priklausomybės	Konfliktai: Nėra
Papildoma medžiaga:	Nėra
Istorija:	Užregistruotas 2005 liepos 30.

Reikalavimas #:	4	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 4
Aprašymas:	Sistema pati negali pradėti generavimo proceso. Jį inicijuoja vartotojas, nuspausdamas generavimo pradžią skelbiantį meniu mygtuką.		
Pagrindimas:	Vartotojui suteikiama galimybė savo nuožiūra paleisti generavimo procesą.		
Šaltinis:	Vartotojas		
Tikimo kriterijus:	Rankinis generavimo proceso paleidimas suteikia vartotojui savarankiškumo.		
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2005 rugpjūčio 15.		

Reikalavimas #:	5	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 5
Aprašymas:	Sistema pati išsirenka generavimo procesui reikalingus duomenis ir pateikia juos vartotojui.		
Pagrindimas:	Be pradinių duomenų *.dxf failuose yra daug nereikalingų šalutinių duomenų.		
Šaltinis:	Sistema		
Tikimo kriterijus:	Vartotojui pačiam išrinkti reikamus pradinius duomenis būtų labai		

	sudėtinga.	
Priklausomybės	Nėra	Konfliktai: Nėra
Papildoma medžiaga:	Nėra	
Istorija:	Užregistruotas 2005 rugpjūčio 20.	

Reikalavimas #:	6	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 6
Aprašymas:	Sistema iš generavimo duomenų išsirenka minimalias reikšmes. Jos panaudojamos apdorojant pradinis duomenis.		
Pagrindimas:	Pradinių duomenų vertės gali būti labai didelės, o tai apsunkina grafinių primityvų duomenų paskaičiavimą bei trimačio miesto pavaizdavimą.		
Šaltinis:	Sistema		
Tikimo kriterijus:	Vartotojui pačiam išrinkti minimalias reikšmes būtų labai sudėtinga.		
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2005 rugpjūčio 25.		

Reikalavimas #:	7	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 7
Aprašymas:	Sistema, panaudodama minimalias reikšmes, suprastina pradinis duomenis.		
Pagrindimas:	Pradinių duomenų vertės gali būti labai didelės, o tai apsunkina grafinių primityvų duomenų paskaičiavimą bei trimačio miesto pavaizdavimą.		
Šaltinis:	Sistema		
Tikimo kriterijus:	Vartotojui pačiam suprastinti pradines reikšmes būtų labai sudėtinga.		
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2005 rugpjūčio 26.		

Reikalavimas #:	8	Reikalavimo tipas:	Įvykis / panaudojimo atvejis
-----------------	---	--------------------	------------------------------

#: 8

Aprašymas:	Sistema formuoja primityvų (daugiakampių) duomenis.	
Pagrindimas:	Pradinės reikšmės yra suprastintos, tačiau nesugrupuotos, todėl sistema pagal pasirinktus kriterijus pati suformuoja kiekvienam primityvui reikalingus duomenis.	
Šaltinis:	Sistema	
Tikimo kriterijus:	Vartotojui pačiam formuoti primityvų duomenis būtų labai sudėtinga.	
Priklausomybės	Nėra	Konfliktai: Nėra
Papildoma medžiaga:	Nėra	
Istorija:	Užregistruotas 2005 rugpjūčio 28.	

Reikalavimas #:	9	Reikalavimo tipas:	Įvykis / panaudojimo atvejis
			#: 9
Aprašymas:	Sistema, panaudodama grafines bibliotekas, formuoja atskirų objektų primityvus.		
Pagrindimas:	Kadangi trimatis vaizdas bus sudarytas iš primityvų, tai juos visų pirma reikia sugeneruoti.		
Šaltinis:	Sistema		
Tikimo kriterijus:	Primityvus gali formuoti sistema, vartotojas yra tik stebėtojas.		
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2005 rugsėjo 15 .		

Reikalavimas #:	10	Reikalavimo tipas:	Įvykis / panaudojimo atvejis
			#: 10
Aprašymas:	Iš pradžių sukurti primityvai yra vienspalviai. Todėl sistema juos padengia tekstūromis.		
Pagrindimas:	Vaizdas be tekstūrų yra nuobodus ir netikroviškas.		
Šaltinis:	Sistema		
Tikimo kriterijus:	Vartotojas gali tekstūras pasirenka, o sistema jomis padengia grafinius primityvus.		
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		

Istorija:	Užregistruotas 2005 rugsėjo 30 .
-----------	----------------------------------

Reikalavimas #:	11	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 11
Aprašymas:	Kiekvieno objekto primityvai sujungiami į visumą.		
Pagrindimas:	Primityvų visuma sudaro trimatį miesto vaizdą.		
Šaltinis:	Sistema		
Tikimo kriterijus:	Norint užtikrinti sklandų sistemos darbą, vartotojas į procesą nesikiša.		
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2005 spalio 10 .		

Reikalavimas #:	11	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 11
Aprašymas:	Trimatis vaizdas pavaizduojamas vaizduoklio ekrane.		
Pagrindimas:	Tik sugeneruoto, bet neatvaizduoto grafinio vaizdo, vartotojas dar nemato.		
Šaltinis:	Sistema		
Tikimo kriterijus:	Norint užtikrinti sklandų sistemos darbą, vartotojas į procesą nesikiša.		
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2005 spalio 10 .		

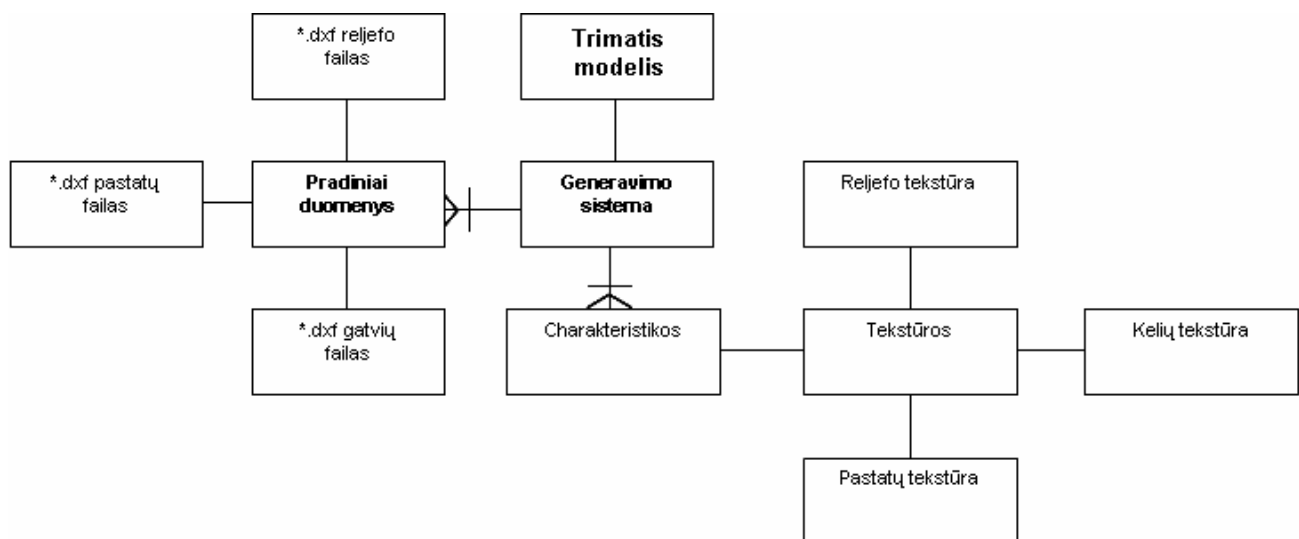
Reikalavimas #:	12	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 12
Aprašymas:	Sugeneruotą trimatį miesto vaizdą vartotojas gali išsaugoti paveikslėlio pavidalu. Išsaugojimo procesas inicijuojamas, nuspaudus atitinkamą meniu mygtuką.		
Pagrindimas:	Vartotojui patogiu vienu metu turėti kelis miesto vaizdus. Tuomet jis gali nesunkiai juos palyginti.		
Šaltinis:	Vartotojas		
Tikimo kriterijus:	Erdvinių vaizdų išsaugojimas vartotojui leidžia juos kaupti, analizuoti		

	bei lyginti su kitais vaizdais.	
Priklausomybės	Nėra	Konfliktai: Nėra
Papildoma medžiaga:	Nėra	
Istorija:	Užregistruotas 2005 spalio 30 .	

Reikalavimas #:	13	Reikalavimo tipas:	Įvykis / panaudojimo atvejis #: 13
Aprašymas:	Vartotojas, norėdamas iš naujo atlikti generacijos procesą, privalo išvalyti esamą grafinį vaizdą. Išsaugojimo procesas inicijuojamas, nuspaudus atitinkamą meniu mygtuką.		
Pagrindimas:	Tai padeda atlaisvinti kompiuterio resursus bei pagreitinti tolesnį sistemos darbą.		
Šaltinis:	Vartotojas		
Tikimo kriterijus:	Naujas grafinis vaizdas bus generuojamas tik išvalius jau sugeneruotą. Priešingu atveju sistema gali būti labai apkrauta.		
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2005 lapkričio 20 .		

### 3.6.2. Reikalavimai duomenims

Duomenų ypatybes puikiai nurodo ši schema



6 pav. Reikalavimai duomenims

### 3.7. Reikalavimai sistemos išvaizdai

Reikalavimas #:	1	Reikalavimo tipas:	Įvykis / panaudojimo atvejis	1-5
			#:	
Aprašymas:	Lengvai valdoma sąsajos(meniu, mygtukai ir kt).			
Pagrindimas:	Vartotojas neturėtų ilgai mokytis, kad suprastų programos darbą. Tai turi būti paprastai suvokiama sistema.			
Šaltinis:	Vartotojas			
Tikimo kriterijus:	Lengvai suprantama ir patogi sąsaja.			
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	4	
Priklausomybės	Nėra	Konfliktai:	Nėra	
Papildoma medžiaga:	Nėra			
Istorija:	Užregistruotas 2006 vasario 5d.			

Reikalavimas #:	2	Reikalavimo tipas:	Įvykis / panaudojimo atvejis	1-5
			#:	
Aprašymas:	Nesudėtingas meniu.			
Pagrindimas:	Vartotojui turi būti paprasta pasirinkti sistemos meniu punktus. Jis turi lengvai atsimentti kokiam punkte bus konkreti komanda.			
Šaltinis:	Vartotojas			
Tikimo kriterijus:	Lengvai suprantama ir patogi sąsaja			
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	4	
Priklausomybės	Nėra	Konfliktai:	Nėra	
Papildoma medžiaga:	Nėra			
Istorija:	Užregistruotas 2006 vasario 5d.			

### 3.8. Sistemos panaudojamumo reikalavimai

Reikalavimas #:	1	Reikalavimo tipas:	11 Įvykis / panaudojimo atvejis	1-5
			#:	
Aprašymas:	Sistema minimizuoti vartotojo klaidų tikimybę.			

Pagrindimas:	Sistemoje turi veikti tam tikros trimačių vaizdų generavimo taisyklės, kurios ir sumažintų klaidų tikimybę.		
Šaltinis:	Vartotojas		
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	5
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2006 vasario 5d.		

Reikalavimas #:	2	Reikalavimo tipas:	11	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Sistema įsisavinamui nereikia specialaus apmokymo.				
Pagrindimas:	Vartotojas turi išmanyti bent braižymo pagrindus.				
Šaltinis:	Vartotojas				
Tikimo kriterijus:	Sistema galima dirbti turint minimalias braižymo žinias.				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	5		
Priklausomybės	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2006 vasario 5d.				

Reikalavimas #:	3	Reikalavimo tipas:	11	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Sistemoje turi būti įdiegta visapusiška pagalba vartotojui.				
Pagrindimas:	Jei vartotojas nežino ar neranda tam tikros informacijos apie sistemą, jis ją gali realiu laiku surasti sistemos pagalbos vadove, kuris pateikiamas kartu su sistema.				
Šaltinis:	Vartotojas				
Tikimo kriterijus:	Pagalba vartotojui.				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	3		
Priklausomybės	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				

Istorija:	Užregistruotas 2006 vasario 5d.
-----------	---------------------------------

### 3.9. Vykdyto charakteristikų reikalavimai

Reikalavimas #:	1	Reikalavimo tipas:	12	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Efektyvus resursų panaudojimas.				
Pagrindimas:	Sistema neturi trukdyti kitoms sistemoms efektyviai naudoti resursus. Turi būti nustatyti tam tikri resursų naudojimo prioritetai kitų sistemų atžvilgiu.				
Šaltinis:	Vartotojas				
Tikimo kriterijus:	Efektyviai paskirstyti resursai				
Užsakovo tenkinimas:	3			Užsakovo netenkinimas:	3
Priklausomybės	Nėra			Konfliktai:	Nėra
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2006 vasario 5d.				

Reikalavimas #:	2	Reikalavimo tipas:	12	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Efektyvus užduočių vykdymas				
Pagrindimas:	Sistema turi atlikti užduotis per kiek galima trumpesnę laiką. Sistema netūrėtų sukelti įtarimo vartotojui, kad užduotis nevykdoma.				
Šaltinis:	Vartotojas				
Tikimo kriterijus:	Kuo greičiau vykdomos užduotys.				
Užsakovo tenkinimas:	5			Užsakovo netenkinimas:	5
Priklausomybės	Nėra			Konfliktai:	Nėra
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2006 vasario 5d.				



Reikalavimas #:	4	Reikalavimo tipas:	12	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Išplečiamumas				
Pagrindimas:	Sistema turi turėti papildymo naujais komponentais galimybę.				
Šaltinis:	Vartotojas				
Tikimo kriterijus:	Priklausomai nuo vartotojo norų, sistema gali būti lengvai išplečiamama				
Užsakovo tenkinimas:	2			Užsakovo netenkinimas:	2
Priklausomybės	Nėra			Konfliktai:	Nėra
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2006 vasario 5d.				

### 3.10. Veikimo sąlygų reikalavimai

Reikalavimas #:	1	Reikalavimo tipas:	13	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Paprastas produkto įdiegimas				
Pagrindimas:	Vartotojui sistemos įdiegimas kompiuteryje privalo būti nesudėtingas				
Šaltinis:	Vartotojas				
Tikimo kriterijus:	Lengvas sistemos diegimas, sumažina vartotojo klaidų tikimybę.				
Užsakovo tenkinimas:	5			Užsakovo netenkinimas:	4
Priklausomybės	Nėra			Konfliktai:	Nėra
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2006 vasario 5d.				

### 3.11. Sistemos priežiūros reikalavimai

Reikalavimas #:	1	Reikalavimo tipas:	14	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Visada vykdomas sistemos atnaujinimas.				
Pagrindimas:	Dirbti su sistema labai padeda sistemingi atnaujinimai ir rekomendacijos.				

Šaltinis:	Vartotojas		
Tikimo kriterijus:	Padeda išvengti galimų sistemos „lūžių“		
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	4
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2006 vasario 5d.		

### 3.12. Sistemos saugumo reikalavimai

Saugumo reikalavimai tiesiogiai susiję su rizika, todėl sistema turi atitikti nurodytus saugumo reikalavimus.

Reikalavimas #:	1	Reikalavimo tipas:	15	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Sistema neleis įvesti klaidingos informacijos.				
Pagrindimas:	Kadangi modelio generavimo metu atliekama daug paskaičiavimų, klaidingai įvesta informacija gali labai apkrauti ar net „užlaužti“ kompiuterį.				
Šaltinis:	Vartotojas				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	4		
Priklausomybės	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2006 vasario 5d.				

### 3.13. Kultūriniai-politiniai reikalavimai

Reikalavimas #:	1	Reikalavimo tipas:	16	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Sistemoje naudojama korektiška lietuvių kalba, stengiamasi išvengti žargoninių terminų panaudojimo.				
Pagrindimas:	Vartotoją gali atbaidyti programoje panaudoti žargoninės kalbos terminai.				
Šaltinis:	Vartotojas				
Tikimo kriterijus:	Sukurta sistema atitinka lietuvių kalbos etikos normas.				

Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	3
Priklausomybės	Nėra	Konfliktai:	Nėra
Papildoma medžiaga:	Nėra		
Istorija:	Užregistruotas 2006 vasario 5d.		

### 3.14. Teisiniai reikalavimai

Reikalavimas #:	1	Reikalavimo tipas:	17	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Teisėtas projektas				
Pagrindimas:	Kuriama sistema turi būti teisėta. Be autoriaus sutikimo ji negali būti kopijuojama ar naudojama kitais tikslais.				
Šaltinis:	Vartotojas				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	5		
Priklausomybės	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2006 vasario 5d.				

Reikalavimas #:	2	Reikalavimo tipas:	17	Įvykis / panaudojimo atvejis	1-5
				#:	
Aprašymas:	Išeities kodo atvirumas.				
Pagrindimas:	Produkto išėities kodas nėra komercinis produktas.				
Šaltinis:	Vartotojas				
Tikimo kriterijus:	Produkto išėities kodas gali būti atskleistas mokomiesiems tikslams, tačiau bet koks jo panaudojimas turi būti suderintas su kūrėju.				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	5		
Priklausomybės	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:	Nėra				
Istorija:	Užregistruotas 2006 vasario 5d.				

### **3.15. Projekto išeiga**

#### **3.15.1. Atviri klausimai**

Kuriamas projektas turėtų kuo tiksliau pavaizduoti trimatį miesto modelį. Nukrypimai ir netikslumai turi būti minimalūs. Todėl darbe atliekami skaičiavimai yra prastinami ir apvalinami atsargiai. Atkreipiame dėmesį į tai, kad erdvinio vaizdo tikslumas yra didinamas kompiuterio resursų sąskaita.

#### **3.15.2. Egzistuojantys sprendimai**

Pasaulyje yra nemažai sistemų (Autodesk Map 3D, ArchiCAD) naudojamų tokiems tyrimams, tačiau jos visos yra galingi ir brangūs produktai ir šiek tiek skiriasi nuo šios sistemos tikslo ir galimybių.

#### **3.15.3. Naujos problemos**

##### **Aplinkos diegimo problemos**

Sistemą diegiant gali kilti problemų tik su JVM versija. Šios problemos sprendimas yra paprastas – kartu su programos paketu bus įdėta sklandžiam darbui reikalinga JVM versija.

##### **Įtaka jau instaliuotoms sistemoms**

Programinė įranga neturi jokios įtakos jau instaliuotoms sistemoms.

##### **Neigiamas vartotojų nusiteikimas**

Galima vartotojų neigiama reakcija tik kaip įprasta vartotojams gavus naują produktą, kol vartotojas nėra prie jo priprastęs, kol neįsisavinti įgūdžiai. Siekiant sumažinti šią neigiamą reakciją bus stengiamasi kad programinė įranga neatitoltų nuo vartotojų darbo specifikos, kad būtų suprantama ir lengvai perprantama.

##### **Kliudantys diegimo aplinkos apribojimai**

Jokių kitų specifinių apribojimų, kliudančių diegimui, numatomoje diegimo aplinkoje nėra.

##### **Naujos sistemos sukeltos problemos**

Kadangi sistema neoperuoja su asmeniniais duomenimis, nenumatytos problemos, kurias galėtų sukelti sistemos naudojimas.

### 3.15.4. Uždaviniai

#### Sistemos pateikimo žingsniai:

- ✓ Reikalavimų nustatymas
- ✓ Reikalavimų analizė
- ✓ Spendimų analizė
- ✓ Projekto plano sudarymas.
- ✓ Sistemos projektavimas.
- ✓ Sistemos realizavimas
- ✓ Sistemos testavimas
- ✓ Sistemos diegimas
- ✓ Sistemos eksploatavimas.

### 3.15.5. Vystymo etapai

- ✓ Duomenų, esančių \*.dxf failuose, nuskaitymas.
- ✓ Parinktų charakteristikų analizė.
- ✓ Duomenų išrinkimas, prastinimas ir grupavimas.
- ✓ Primityvų formavimas ir tekstūrų dengimas.
- ✓ Primityvų apjungimas.
- ✓ Modelio redagavimo priemonių vystymas.

## 3.16. Rizikos

### 3.16.1. Projekto rizikų aptarimas

Aptarsime esmines projekto panaudojimo metu kylančias rizikas bei jų įtaką:

**21 lentelė.** Projekto rizikos.

Nr.	Rizikos faktorius	Tikimybė	Įtaka
1.	Projektą naudojančio vartotojo patirtis ir sugebėjimai	Vidutiniška	Rimta
2.	Reikalavimų pasikeitimas	Vidutiniška	Rimta
3.	Poreikis kardinaliai pakeisti sistemą ar sistemos posistemę, kai jos kodas jau paruoštas	Vidutiniška	Rimta
4.	Techninės įrangos gedimas	Žema	Leistina

### 3.16.2. Rizikų sprendimų planas

Galimi rizikų sprendimo būdai pateikti sekančioje lentelėje:

22 lentelė. Rizikos faktoriai ir problemų sprendimas

Nr.	Rizikos faktorius	Problemos sprendimas
1.	Projektą naudojančio vartotojo patirtis ir sugebėjimai	Parinkti asmenis, kurie turi bent minimalius pagrindus dirbant su modeliavimo ar projektavimo sistemomis.
2.	Reikalavimų pasikeitimas	Reikia su užsakovu derinti reikalavimus ir nuo pradžių juos suskirstyti pagal prioritetus. Nesvarbius reikalavimų pasikeitimus galima atidėti sekančiai programinės įrangos versijai. Ir aišku pasilikti pakankamą laiko rezervą svarbiems reikalavimų pakeitimams įgyvendinti.
3.	Poreikis kardinaliai pakeisti sistemą ar sistemos posistemę kai ji jau sukoduota.	Reikia bandyti išsiaiškinti su užsakovu, ar tokie pakeitimai tikrai labai svarbūs ir ar jų negalima atidėti sekančiai programos versijai. Reikia derinant reikalavimus naudoti oficialius dokumentus ant kurių būtų užsakovo parašas rodantis, kad jis sutinka su reikalavimais ir jei jie keisis tai bus skiriama tam tikrų lėšų.
4.	Techninės įrangos gedimas	Numatyti rezervinę techninę įrangą. Jei tokia rizika didelė, reikėtų papildomai skirti lėšų tos įrangos pakeitimui, perėjimui iš vienos darbo vietos į rezervinę.

### **3.17. Vartotojo apmokymas**

Kartu su sistema planuojama pateikti vartotojo dokumentaciją. Bus paruošti šie dokumentacijos tipai:

- ✓ Vartotojo vadovas, sistemų generavimui;
- ✓ Vartotojo vadovas, dažniausiai išskylančių klaidų taisymui.

Vartotojų apmokymai nebus organizuojami, kadangi programa bus labai paprasta ir turint bent modeliavimo ir projektavimo pagrindus problemų neturėtų iškilti.

### 3.18. Programinės įrangos architektūra

#### 3.18.1. Dokumento paskirtis

Architektūros dokumentas skirtas produkto projektavimo stadijai dokumentuoti, naudojant skirtingus sistemos architektūros aspektus, be to, parodyti sistemos architektūrinius sprendimus, kuriuos galima atlikti šioje sistemoje. Taip pat dokumentas padės geriau suprasti programinės įrangos architektas architektūrinius sprendimus.

Galimi dokumento vartotojai:

- ✓ Atliekamo darbo vadovas – tikslas: susipažinti su projektavimo eiga.
- ✓ Sistemos analitikas – aptikti sistemos architektūros trūkumus.
- ✓ Programuotojas – galimybė koreguoti klaidas ar generuoti papildomą kodą.

#### 3.18.2. Apžvalga

Šis dokumentas aprašo programinės įrangos, kuri generoja trimatį miesto modelį pagal geodezinius duomenis, architektūrą.

Bendrai apžvelgsime šio darbo išdėstymą:

- ✓ Sistemos nefunkciniai reikalavimai ir apribojimai → „Architektūros tikslai ir apribojimai“.
- ✓ Panaudojimo atvejai → „Panaudojimo atvejų vaizdas“.
- ✓ Sistemos išskaidymas bei statinė struktūra → „Loginis vaizdas“.
- ✓ Procesai bei jų aprašymas → „Procesų vaizdas“.
- ✓ Sistemos išdėstymas ir techninė įranga → „Išdėstymo vaizdas“.
- ✓ Sistemos duomenų bazinė struktūra → „Duomenų vaizdas“.
- ✓ Kaip architektūra įtakoja kokybės parametrus → „Kokybė“.

#### 3.18.3. Architektūros pateikimas

23 lentelė. Architektūros nagrinėjimo planas.

Vaizdas	Diagramos
Panaudojimo	<ul style="list-style-type: none"><li>• Panaudojimo atvejų diagrama</li></ul>
Loginis	<ul style="list-style-type: none"><li>• Klasių diagramos</li><li>• bendras sistemos loginis vaizdas</li></ul>
Procesų	<ul style="list-style-type: none"><li>• Būsenų diagramos</li><li>• Sekų diagramos</li></ul>



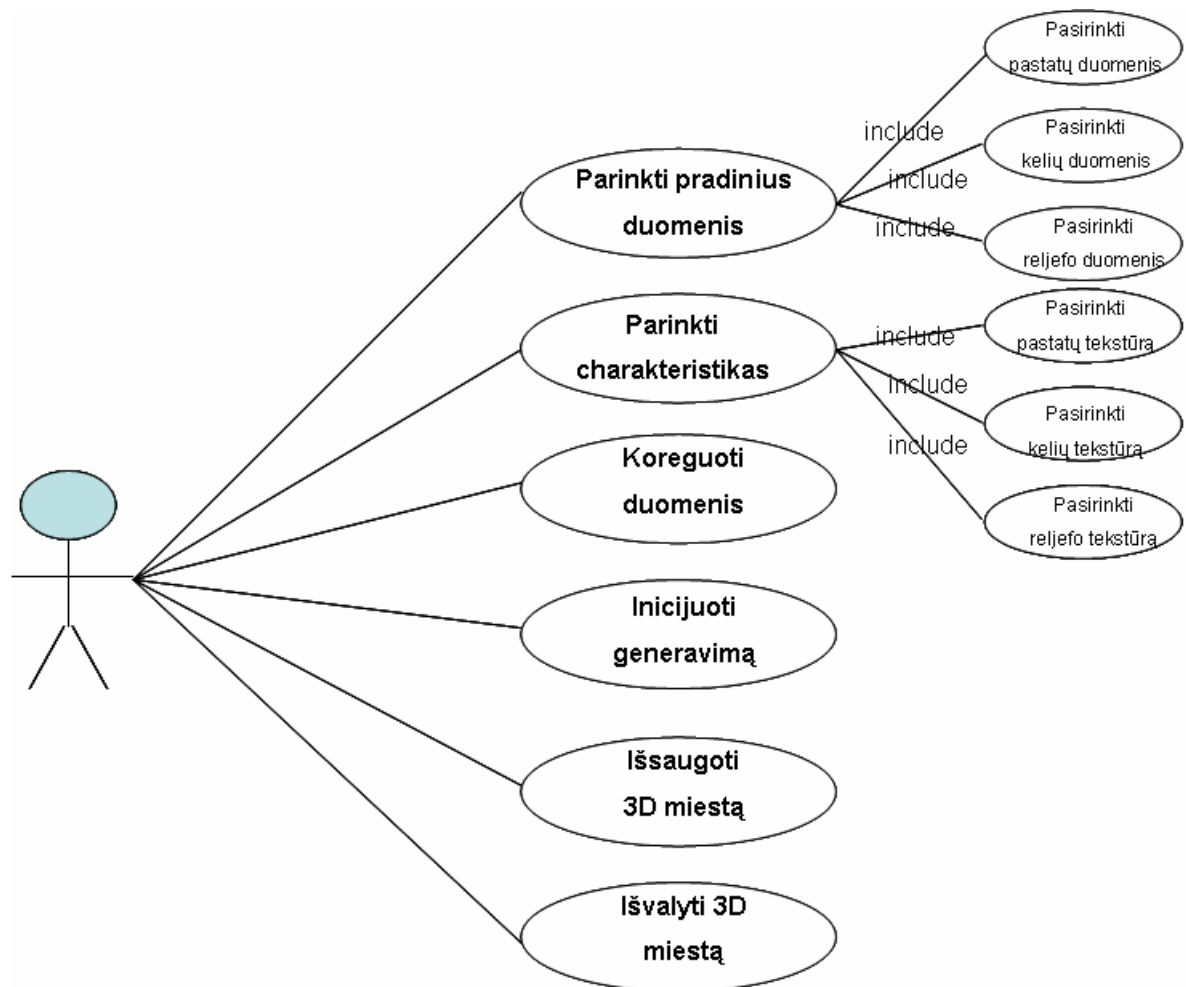
Komponentų išdėstymo	<ul style="list-style-type: none"> <li>• Išdėstymo diagrama</li> </ul>
Duomenų	<ul style="list-style-type: none"> <li>• Duomenų struktūros diagrama</li> </ul>

### 3.18.4. Architektūros tikslai ir apribojimai

- ✓ Tam, kad programinė įranga nebūtų priklausoma nuo platformos, ji bus kuriama naudojant Java virtualiąją mašiną(JVM).
- ✓ Sistema turės pasiimti \*.dxf failuose esančius duomenis.
- ✓ Sistemos rezultatas saugomas grafinio tipo faile.
- ✓ Sistemos architektūra bus parenkama taip, kad ją galima būtų lengva išplėsti ar prijungti naujus komponentus.
- ✓ Pradinių duomenų redagavimui bus gali naudojamas komercinis produktas AutoCAD arba LinuxCAD.

### 3.18.5. Panaudojimo atvejai

*Panaudojimo atvejų diagrama*



7 pav. Panaudojimo atvejai

**24 lentelė.** Panaudojimo atvejis „Parinkti pradinis duomenis“

Nr.	1
Pavadinimas:	Parinkti pradinis duomenis
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Vartotojas pasirenka ir perduoda sistemai *.dxf failuose esančius topografinius duomenis.
Prieš sąlyga:	Vartotojas turi turėti paruoštus topografinius duomenis.
Sužadinimo sąlyga:	Vartotojas pasirenka duomenų failus.
Po sąlyga:	Parinkti duomenų failai yra pateikiami trimačių vaizdų apdorojimo sistemai.

**25 lentelė.** Panaudojimo atvejis „Parinkti charakteristikas“

Nr.	2
Pavadinimas:	Parinkti charakteristikas
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Vartotojas savo nuožiūra pasirenka vaizdines charakteristikas, t.y. parenka kiekvienam objektui (reljefas, keliai, pastatai) tekstūrą.
Prieš sąlyga:	Vartotojas turi turėti pasiruošęs atitinkamų erdvinio vaizdo vaizdo tekstūrų failus.
Sužadinimo sąlyga:	Vartotojas pasirenka tekstūrai reikalingus failus arba nustato reikiamą apšviestumo lygį.
Po sąlyga:	Parinkti tekstūrų failai ir apšviestumo lygis yra pateikiami trimačių vaizdų apdorojimo sistemai.

**26 lentelė.** Panaudojimo atvejis „Koreguoti duomenis“

Nr.	3
Pavadinimas:	Koreguoti duomenis
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Vartotojas koreguoja tik pradinis duomenis, t.y. keičia koordinates ar kitas charakteristikas (pastato tipą, aukštį, sienų ilgį ir pan.). Tai jis gali atlikti su tam skirta specialia redagavimo priemone (LinuxCad, AutoCAD), arba tiesiog Notepad'o pagalba.
Prieš sąlyga:	Vartotojas turi būti pasiruošęs atitinkamus pradinių duomenų failus.
Sužadinimo sąlyga:	Vartotojas panaudodamas atitinkamas redagavimo priemones turi pasirinkti koreguojamus pradinis duomenis.
Po sąlyga:	Koreguoti pradiniai duomenys pateikiami trimačių vaizdų generavimo sistemai.

**27 lentelė.** Panaudojimo atvejis „Inicijuoti generavimą“

Nr.	4
Pavadinimas:	Inicijuoti generavimą
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Vartotojas, parinkęs naujus ar koreguotus duomenis, taip pat tekstūras ir nustatęs kitas charakteristikas, inicijuoja generavimą. Sistema pradeda savo darbą.
Prieš sąlyga:	Vartotojas turi būti pasirinkęs pradinius duomenis bei objektų tekstūras.
Sužadinimo sąlyga:	Vartotojas nuspaudžia funkcijos, kuri inicijuoja generavimą, mygtuką.
Po sąlyga:	Sistema pradeda darbą.

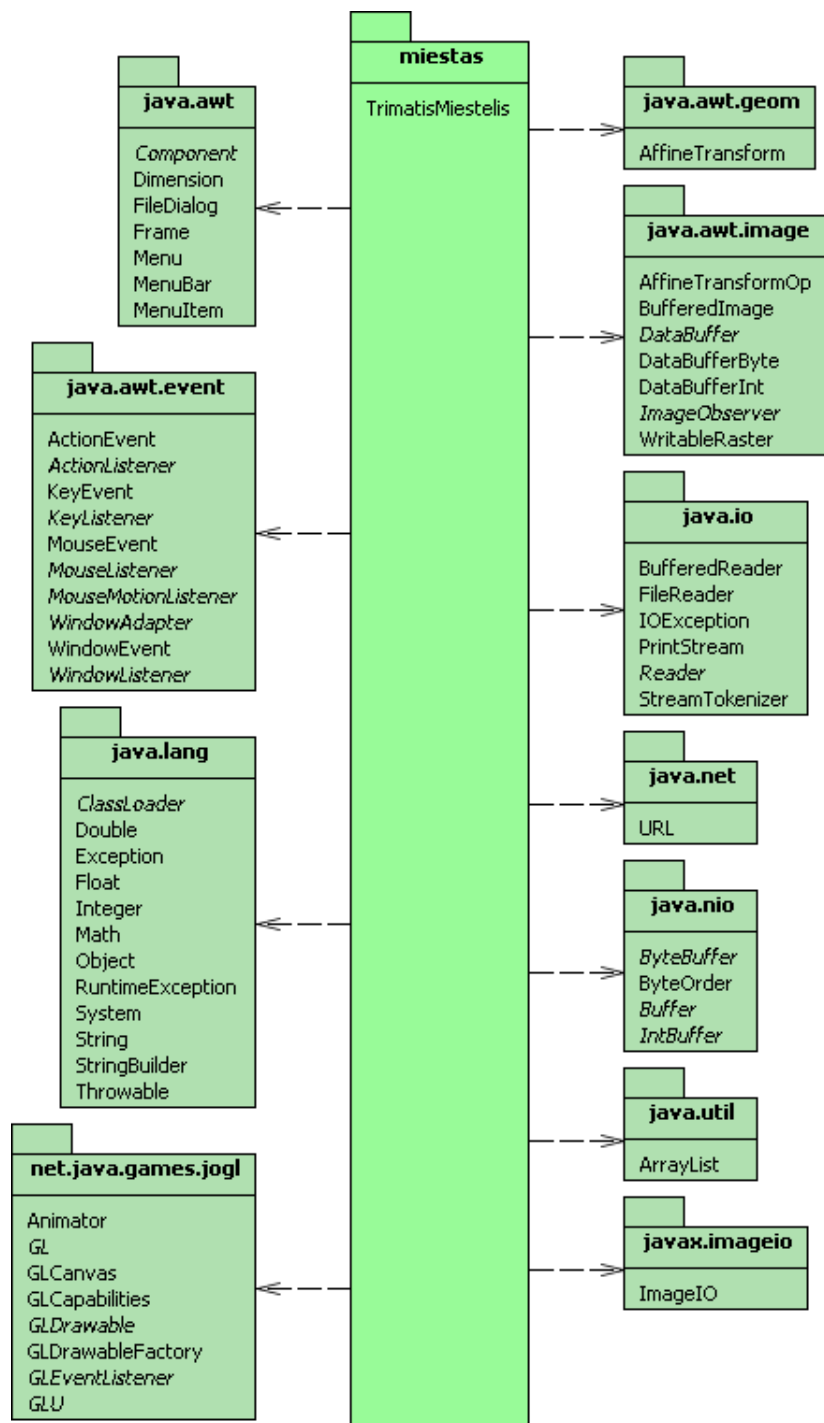
**28 lentelė.** Panaudojimo atvejis „Išsaugoti 3D miestą“

Nr.	13
Pavadinimas:	Išsaugoti 3D miestą
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Kuomet sistema sudgeneruoja trimatį miesto vaizdą, vartotojas gali jį išsaugoti paveikslėlio pavidalu.
Prieš sąlyga:	Sistema turi būti sugeneravusi trimatį vaizdą.
Sužadinimo sąlyga:	Išsaugojimo funkciją vartotojas iškviečia meniu pasirinkdamas atitinkamą meniu mygtuką.
Po sąlyga:	Turime išsaugotą trimatį miesto vaizdą.

**29 lentelė.** Panaudojimo atvejis „Išvalyti 3D miestą“

Nr.	14
Pavadinimas:	Išvalyti 3D miestą
Vartotojo pavadinimas:	Vartotojas
Aprašas:	Jeigu vartotojas, panaudodamas naujus duomenis, nori sugeneruoti naują trimatį vaizdą, tuomet jis dar prieš tai turi išvalyti anksčiau sugeneruotą vaizdą.
Prieš sąlyga:	Sistema turi būti sugeneravusi trimatį vaizdą.
Sužadinimo sąlyga:	Išvalymo funkciją vartotojas iškviečia meniu išsirinkdamas atitinkamą meniu mygtuką.
Po sąlyga:	Turime išvalytą tuščią langą, kuriame galime sugeneruoti kitą trimatį vaizdą.

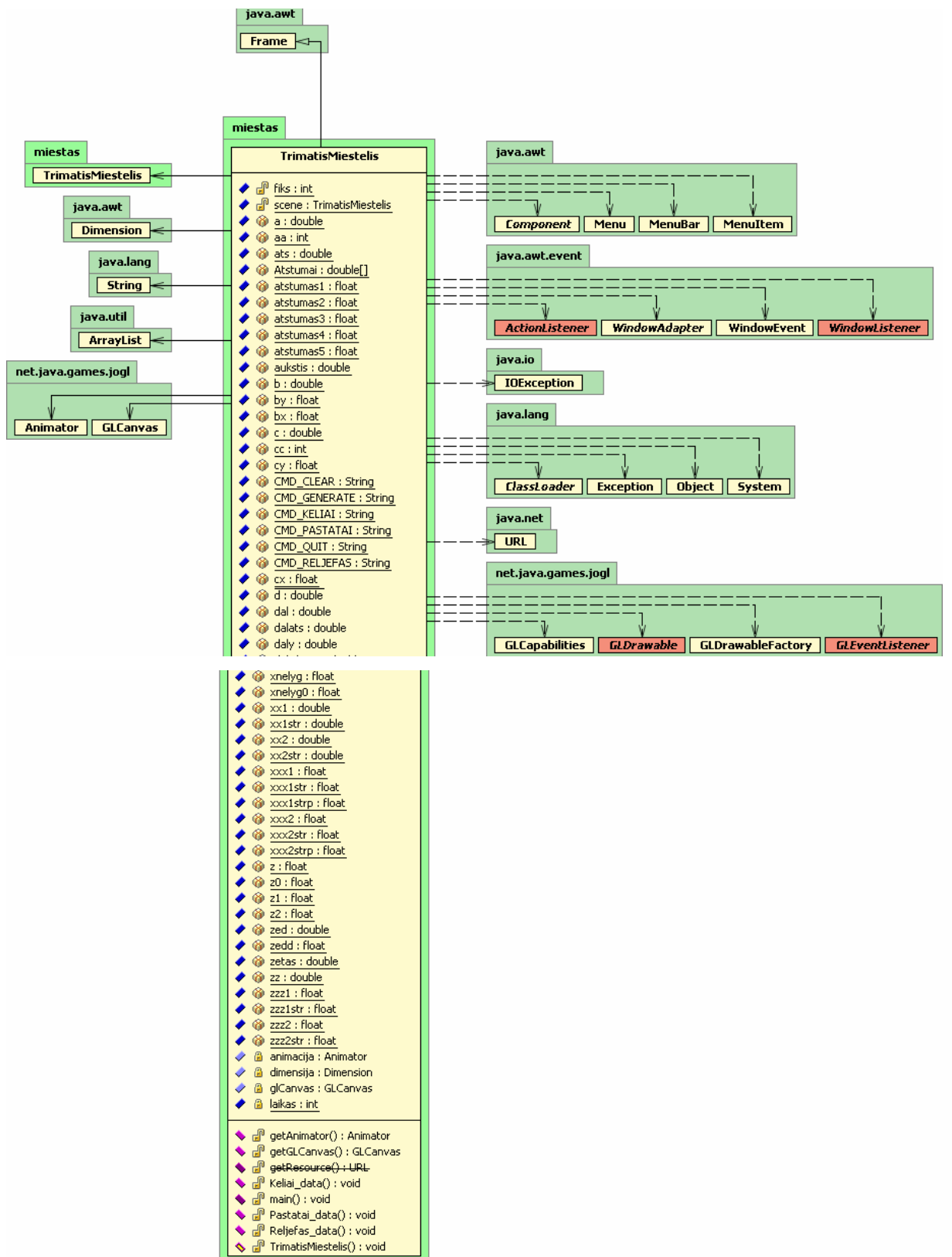
### 3.18.6. Sistemos loginis vaizdas ir klasių detalizavimas



8 pav. Sistemos loginis vaizdas

Šiame darbe buvo panaudotas vienas paketas („Miestas“). Jis turi dvi klases:

1. Klasė „TrimatisMiestelis“. Joje yra metodai, kurie naudojami nuskaitant pradinius duomenis iš \*.dxf failų, sukuriant animatorių, grafinį programos pavidalą ir paleidžiant pačią programą. Šios klasės detalus vaizdas atrodo taip:



9 pav. Klasės „TrimatisMiestelis“ detalus vaizdas

O visą pagrindinę informaciją apie šią klasę :

Pavadinimas – TrimatisMiestelis.

Tikslai – Pradinių duomenų nuskaitymas ir sutalpinimas į masyvų sąrašus.

Animatoriaus, kuris atsakingas už judėjimą, sukūrimas.

Programos išvedimas į darbinį režimą.

Ryšiai – Vienintelė klasė su kuria vyksta duomenų mainai ir bendradarbiavimas yra „Renderer“.

Metodai:

Keliai\_data() – skirtas nuskaityti, sužymėti ir sutalpinti duomenis apie kelius į pradinių duomenų masyvus.

Pastata\_data() - skirtas nuskaityti, sužymėti ir sutalpinti duomenis apie pastatus į pradinių duomenų masyvus.

Reljefas\_data() - skirtas nuskaityti, sužymėti ir sutalpinti duomenis apie reljefą į pradinių duomenų masyvus.

Main() – skirtas generuoti programos interfeisą bei apibendrinti ir paleisti programą.

2. Klasė „Renderer“. Ji turi visus pagrindinius metodus, kurie yra naudojami tiek apdojant duomenis, tiek formuojant primityvų duomenis, tiek generuojant trimatį miesto vaizdą. Informaciją apie šią klasę surasime sekančioje lentelėje:

Pavadinimas – Renderer.

Tikslai – Duomenų apdorojimas ir trimačio miesto vaizdo generavimas.

Ryšiai – Vienintelė klasė su kuria vyksta duomenų mainai ir bendradarbiavimas yra „TrimatisMiestelis“.

Metodai:

Apdoroti() – skirtas suprastinti ir išskirstyti pradinius objektų duomenis bei suformuoti primityvų duomenis. Visų pirma surandami koordinacių minimumai, o tik po to atliekamas koordinacių suprastinimas. Taip pat paskaičiuojami gatvių primityvų bei suformuojami reljefo primityvų duomenys.

Kel\_tex() – skirtas pasirinkti norimą gatvių tekstūrą.

PasDG() – skirtas pasirinkti norimą degių gyvenamų pastatų tekstūrą.

PasDN() - skirtas pasirinkti norimą degių negyvenamų pastatų tekstūrą.

PasMG() – skirtas pasirinkti norimą mūrinių gyvenamų pastatų tekstūrą.

PasMN() - skirtas pasirinkti norimą mūrinių negyvenamų pastatų tekstūrą.

Rel\_tex() - skirtas pasirinkti norimą reljefo tekstūrą.

floor() – skirtas generuoti trimačio miesto pagrindą.

Gatves() – skirtas generuoti trimačio miesto gatves.

Reljefo\_daugiakampiai() – skirtas generuoti trimačio miesto reljefą.

Reljefo\_linijos() - skirtas generuoti trimačio miesto reljefo linijas.

Stogai() - skirtas generuoti trimačio miesto pastatų stogus.

Sudetingi\_Pastatai() - skirtas generuoti trimačio miesto pastatus.

init\_lights() – skirtas inicijuoti apšvietimą bei nustatyti šviesos šaltinio charakteristikas.

lights() – skirtas nustatyti trimačio miesto vaizdo apšvietimo šaltinio padėtį.

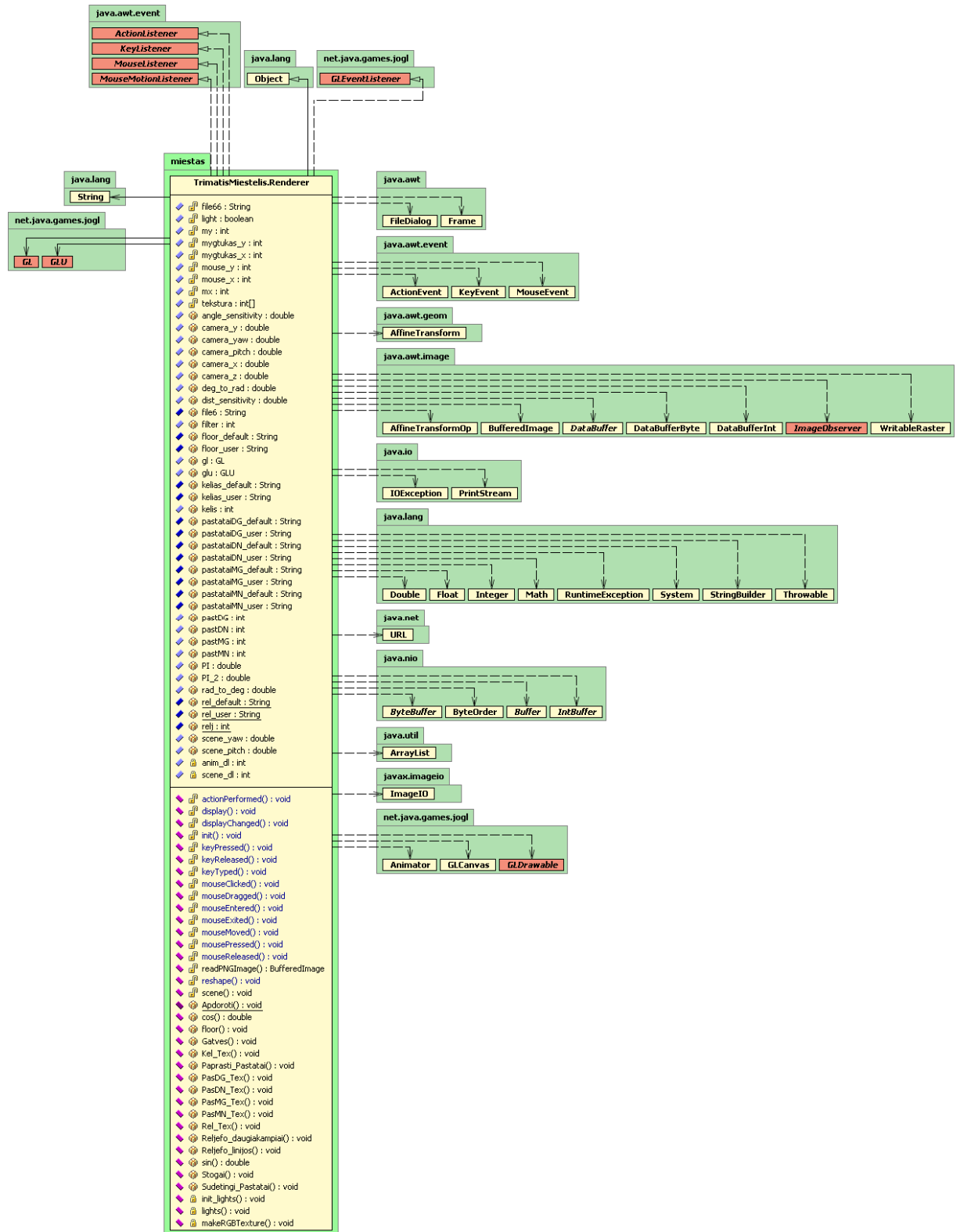
makeRGBTexture() – skirtas suformuoti tekstūrą.

display() – skirtas atvaizduoti sukurtą trimatį vaizdą.

reshape() – skirtas, priklausomai nuo lango dydžio, modifikuoti pavaizduotą trimatį vaizdą.

mouseDragged() – skirtas apdoroti kameros padėtį. Priklausomai nuo to, kuris mygtukas (pirmas, antras arba trečias) yra nuspaustas, judinant pelę yra keičiama kameros padėtis.

Priklausomai nuo vartotojo norų, kitų metodų kodas bus parašytas vėliau.

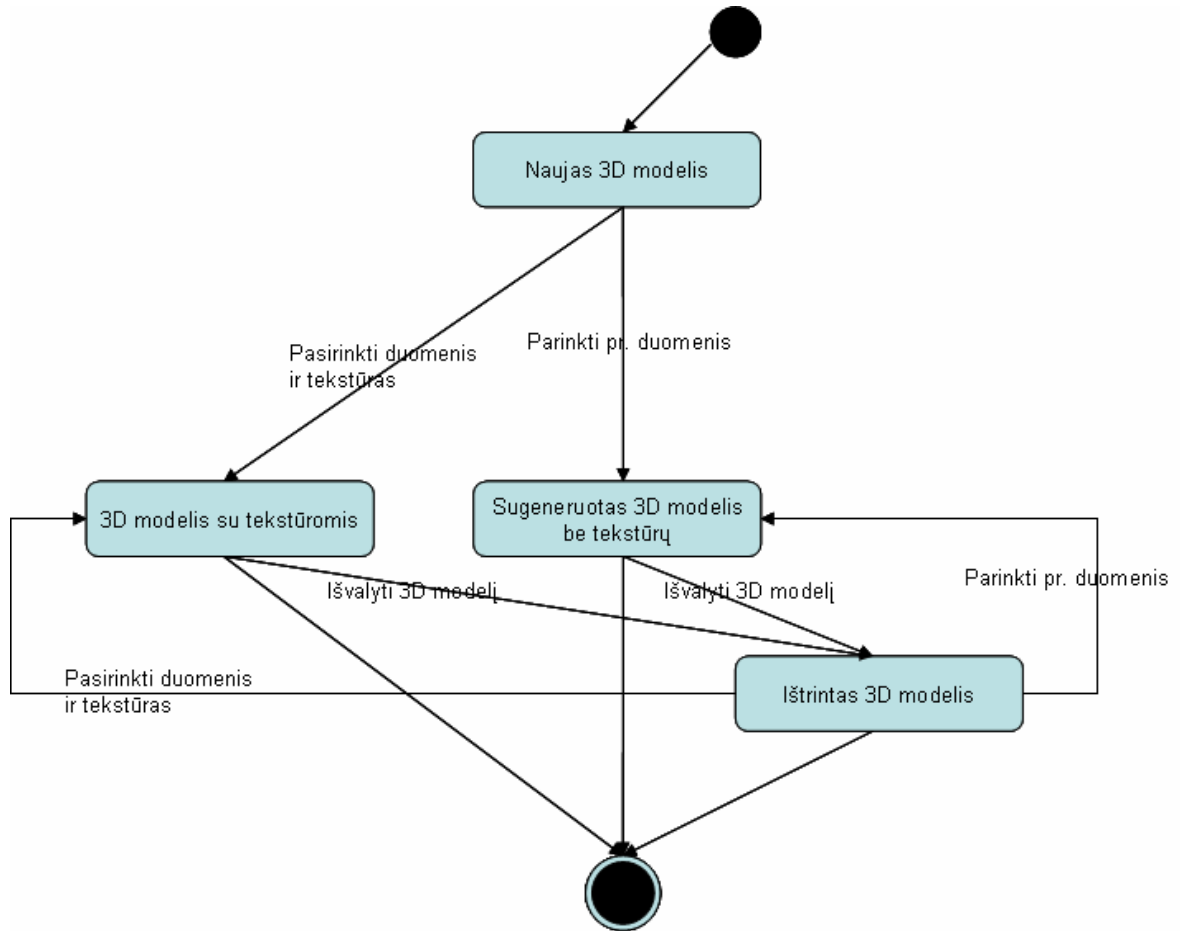


10 pav. Klasės „Renderer“ detalus vaizdas



### 3.18.7. Procesų vaizdas

Trimačiam miesto modeliui sugeneruoti skirta būsenų diagrama



11 pav. Trimačio modelio generavimo būsenų vaizdas

### 3.18.8. Bendradarbiavimo ir sekų diagramos



Prieš sukuriant bendradarbiavimo ir sekų diagramas, trumpai apžvelgsime jų komponentus:

- asmuo, besinaudojantis trimačio miesto generavimo sistema.

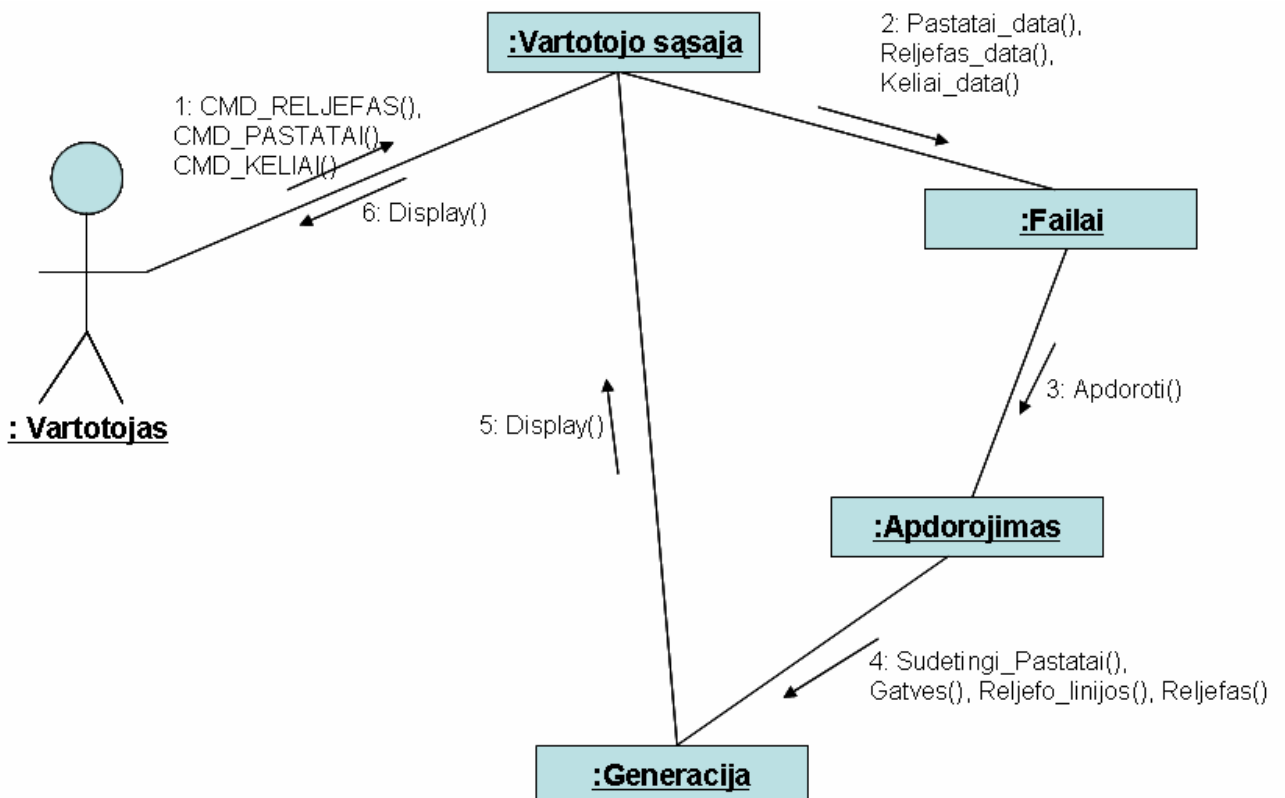
**:Vartotojo sąsaja** - tai tarpininkas tarp vartotojo ir programinio kodo.

**:Failai** - metodai, skirti nuskaityti duomenų failus.

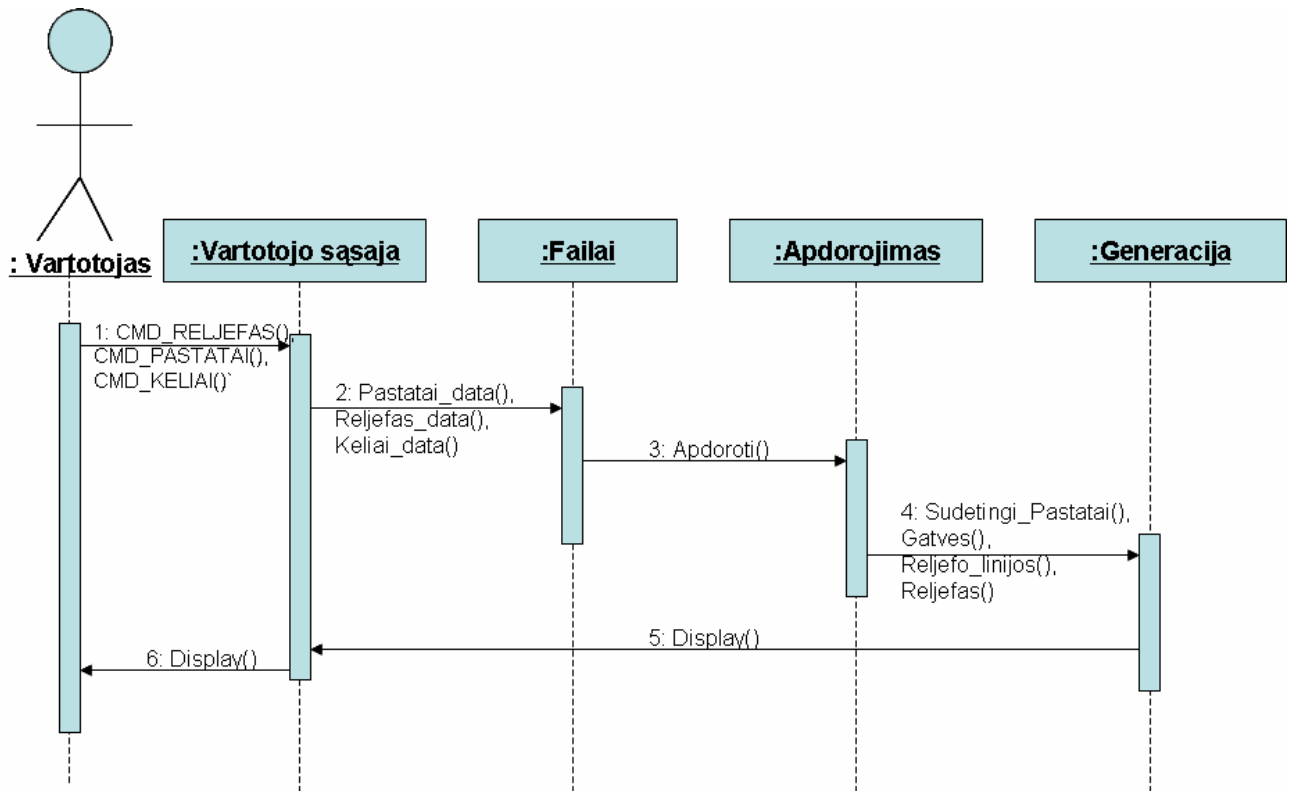
**:Apdorojimas** - metodai skirti apdoroti duomenis.

**:Generacija** - metodai skirti generuoti trimatį vaizdą.

Dabar panaudodami šiuos komponentus sudarysime bendradarbiavimo ir sekų diagramas.

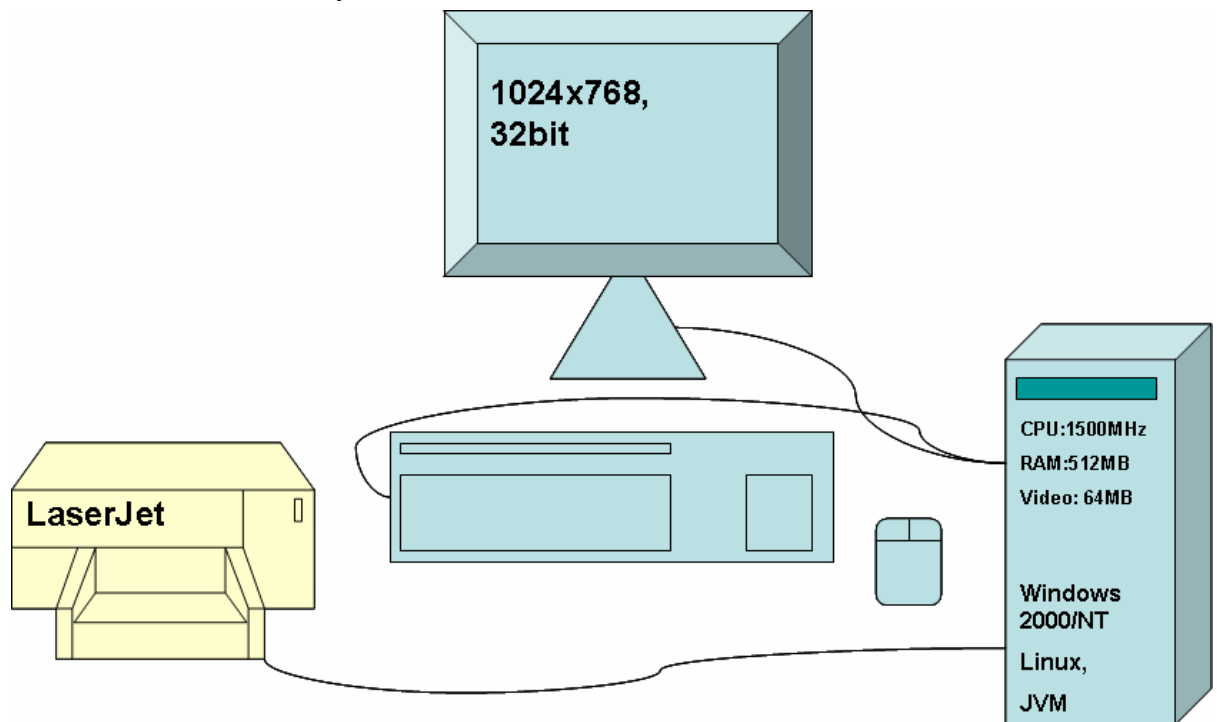


12 pav. Generuoti trimatį miesto vaizdą.



13 pav. Generuoti trimatį miesto vaizdą.

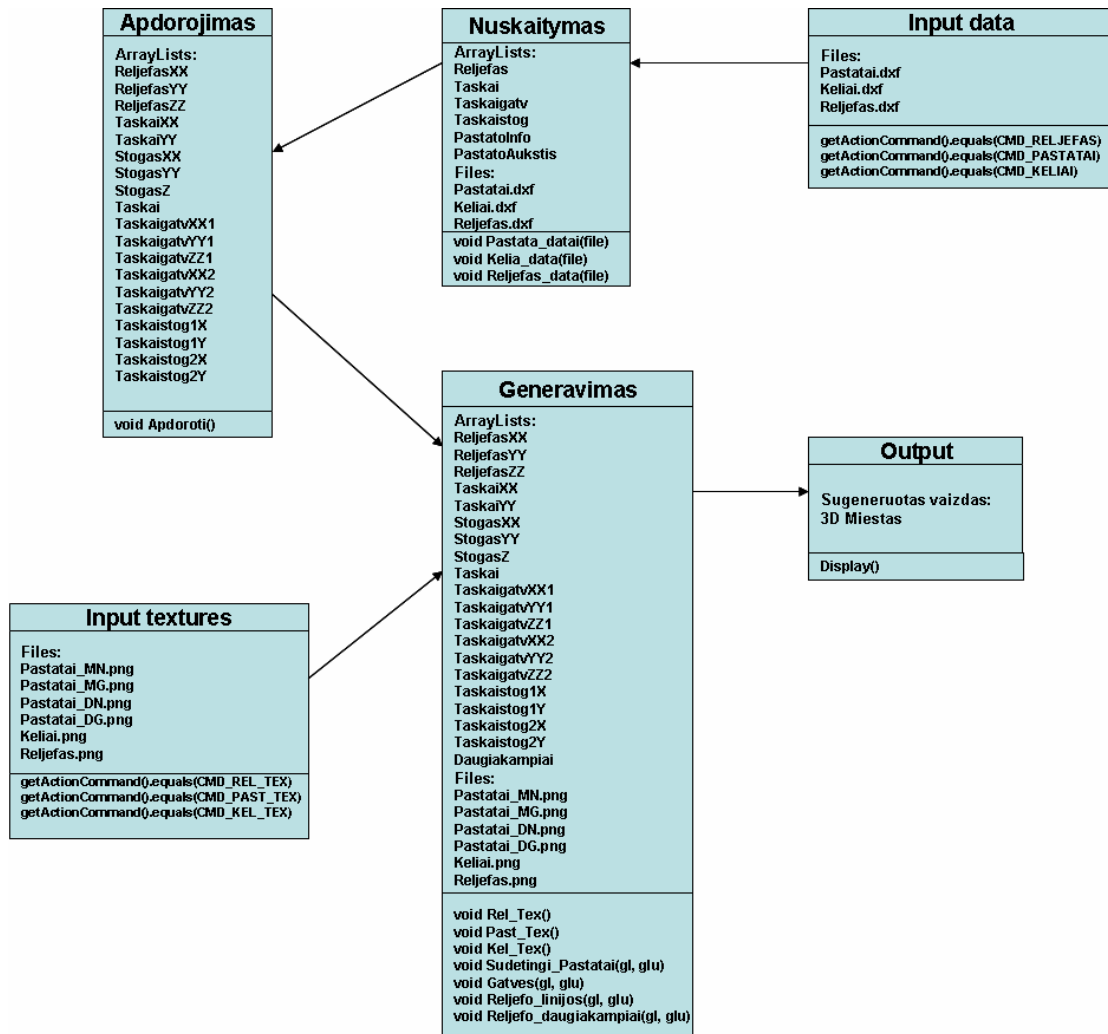
### 3.18.9. Sistemos išdėstymo vaizdas



14 pav. Sistemos vaizdas

### 3.18.10. Duomenų vaizdas

Pateiksime schemą, kurioje nurodysime kokie duomenys ir metodai yra naudojami atskirose programos dalyse (ne klasėse):



15 pav. Duomenų vaizdas

30 lentelė. Duomenų vaizdo komponentų aprašymas

Komponentas	Aprašymas
Input Data	Pradinių generavimo duomenų talpinimas
Input Texture	Objektų tekstūrų talpinimas
Nuskaitymas	Duomenų ištraukimas iš failų ir sutapinimas į masyvus
Apdorojimas	Duomenų prastinimas
Generavimas	Vaizdinių elementų generavimas ir apjungimas
Output	Triamačio vaizdo išvedimas

### 3.18.11. Vartotojo sąsajos specifikacija

#### *Vartotojo sąsajos esmė*

Vartotojo sąsaja yra „atsakinga“ už vartotojui suteikiamą funkcionalumą. Ji yra sudaryta iš klasių, kurios suteikia vartotojui vizualų programos vaizdą.

Vartotojo sąsaja susideda iš:

- ✓ meniu;
- ✓ papildomų mygtukų;
- ✓ vaizdavimo lauko;
- ✓ įspejamų pranešimų.
- ✓ ir kt.

#### *Atsakomybės*

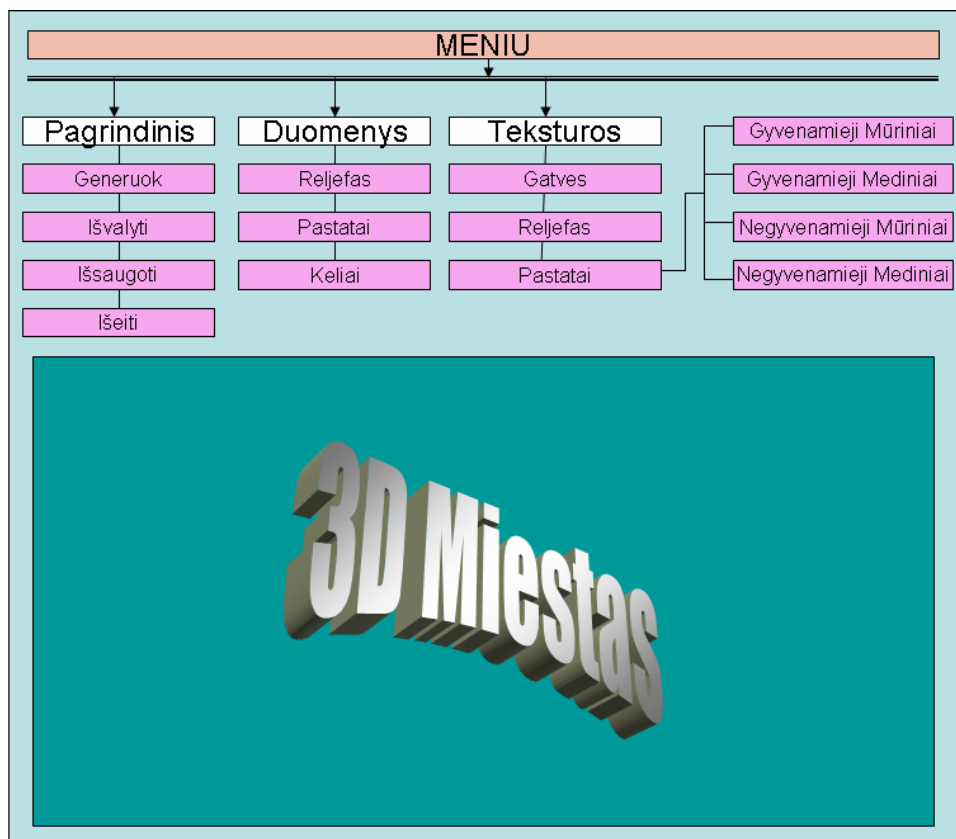
Vartotojo sąsaja turi užtikrinti:

- ✚ kad programos naudotojas galėtų lengvai pasirinkti norimus veiksmus su sistema:
  - išsaugoti
  - redaguoti
  - uždaryti
  - ištrinti
  - ir kt.
- ✚ kad programos naudotojas galėtų lengvai pasirinkti reikiamus duomenis ar charakteristikas bei gautų reikiamą rezultatą;
- ✚ kad vartotojas galėtų atšaukti bet kokį atliktą veiksmą;
- ✚ kad vartotojas galėtų atsidaryti kelis langus vienu metu

#### *Apribojimai*

Vartotojas turi būti apsaugotas nuo atsitiktinių klaidų atliekant svarbius veiksmus, tam, o tai yra realizuota Java programiniame kode naudojant *exception* konstrukciją.

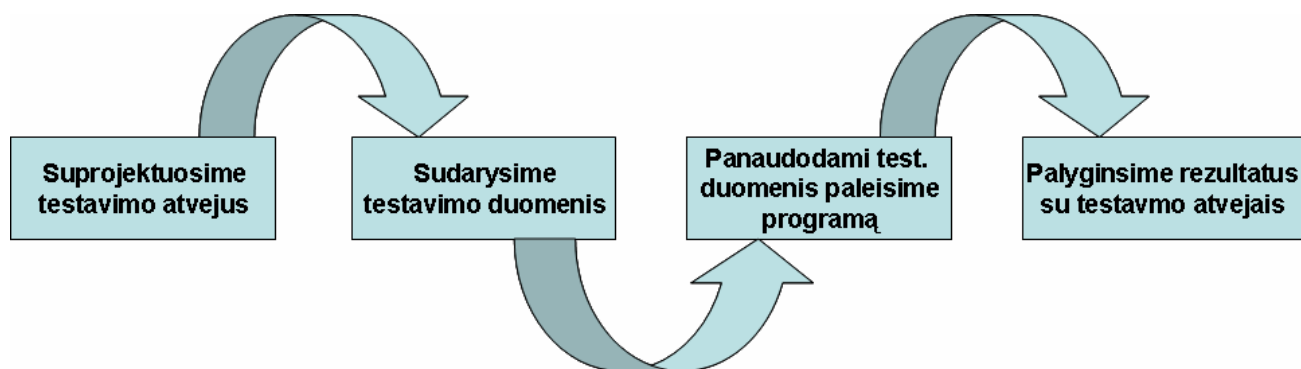
#### *Struktūra*



16 pav. Vartotojo sąsajos struktūra

### 3.18.12 Testavimas

Testavimas – tai visos programos ir jos komponentų vykdymas su pasirinktais pradiniais duomenimis, siekiant nustatyti sistemos klaidas. Visas testavimo procesas planuojamas atlikti



17 pav. Testavimo procesas.

## 4. VARTOTOJO DOKUMENTACIJA

### 4.1. Programinės įrangos aprašymas

Programinė įranga skirta generuoti trimatį miesto vaizdą, panaudojant topografinius duomenis. Vartotojas savo nuožiūra gali pasirinkti pradinis duomenis, tekstūras ir inicijuoti miesto generavimą.

### 4.2. Programinės įrangos panaudojimas

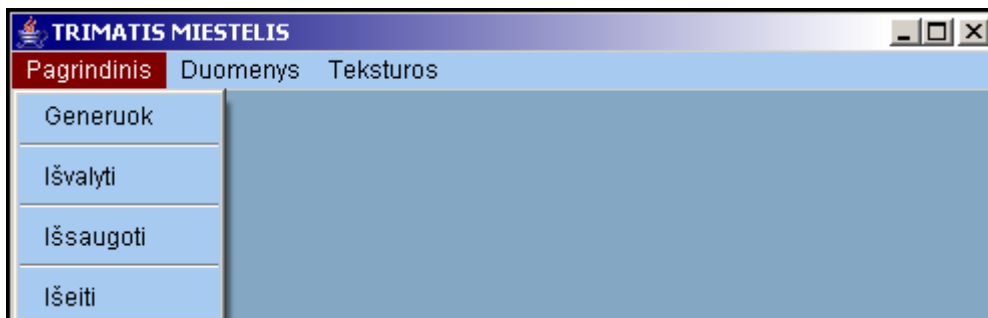
Visas programos valdymas sudėtas į meniu:



18 pav. Meniu

#### Meniu sandara

Meniu - Pagrindinis:



19 pav. Meniu komponentas „Pagrindinis“

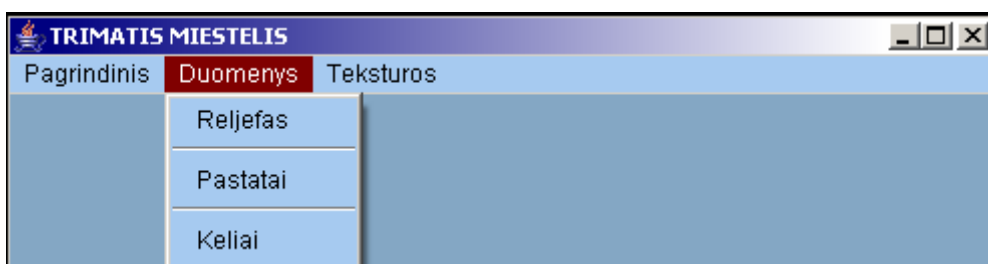
*Generuok* – nuspaudus, inicijuojamas trimačio miesto vaizdo generavimas.

*Išvalyti* – nuspaudus, išvalomas jau sugeneruotas trimačio miesto vaizdas.

*Išsaugotii* – nuspaudus, išsaugomas jau sugeneruotas trimačio miesto vaizdas

*Išeiti* – nuspaudus, uždaroma programa.

Meniu - Duomenys:



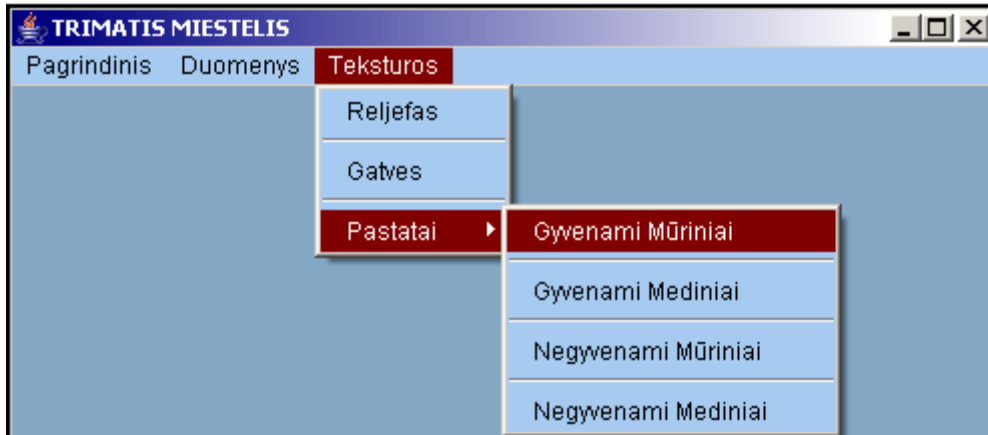
20 pav. Meniu komponentas „Duomenys“

*Reljefas* – nuspaudus, pasirenkamas reljefo duomenų failas (\*.dxf).

*Pastatai* – nuspaudus, pasirenkamas miesto pastatų duomenų failas (\*.dxf).

*Keliai* – nuspaudus, pasirenkamas miesto gatvių duomenų failas (\*.dxf).

Meniu - Tekstūros:



**21 pav.** Meniu komponentas „Tekstūros“

*Reljefas* – nuspaudus, pasirenkamas reljefo tekstūros failas (\*.png).

*Gatves* – nuspaudus, pasirenkamas miesto gatvių tekstūros failas (\*.png).

*Pastatai:*

Gyvenami Mūriniai - nuspaudus, pasirenkamas miesto gyvenamųjų mūrinių pastatų tekstūros failas (\*.png).

Gyvenami Mediniai - nuspaudus, pasirenkamas miesto gyvenamųjų medinių pastatų tekstūros failas (\*.png).

Negyvenami Mūriniai - nuspaudus, pasirenkamas miesto negyvenamųjų mūrinių pastatų tekstūros failas (\*.png).

Negyvenami Mediniai - nuspaudus, pasirenkamas miesto negyvenamųjų medinių pastatų tekstūros failas (\*.png).

Pelės funkcijos – vaizdo apžvalga vykdoma judinant pelę kartu nuspaudus vieną pelės mygtuką:

1 – as mygtukas – sukiojama kamera.

2 – as mygtukas – vaizdas artinamas arba tolinamas.

3 – ias mygtukas – keičiama kameros padėtis.

**Sistemos diegimas:** programinės įrangos specialiai diegti nereikia. Pakanka nusikopijuoti ją į tam tikrą katalogą.

**Sistemos reikalavimai:** CPU > 1500MHz, 512 RAM, 64MB Video, Windows 2000/XP arba Linux.



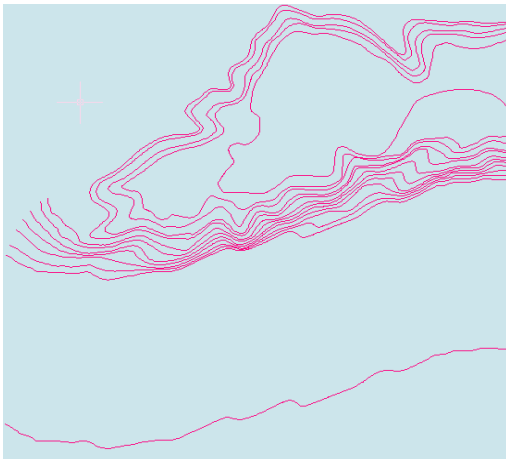
## 5. KOKYBĖS ĮVERTINIMAS

Kuriama sistema nėra pritaikyta naudoti kompiuteriuose, kurių sparta yra mažesnė nei nurodyta paveikslėlyje nr. 14. Taip yra dėl šių priežasčių:

- naudojama daug pradinių duomenų;
- atliekama daug paskaičiavimų;
- generuojami trimačiai vaizdai;
- naudojama animacija.

Trimatis miesto vaizdas sugeneruojamas gana kokybiškai ir tiksliai. Tai patvirtinsime atlikdami nedidelį testuką. Tiesiog paimsime pradinis duomenis (topografinius reljefo, gatvių, pastatų topografinius duomenis) ir parodysime po generacijos gautą trimatį miesto vaizdą.

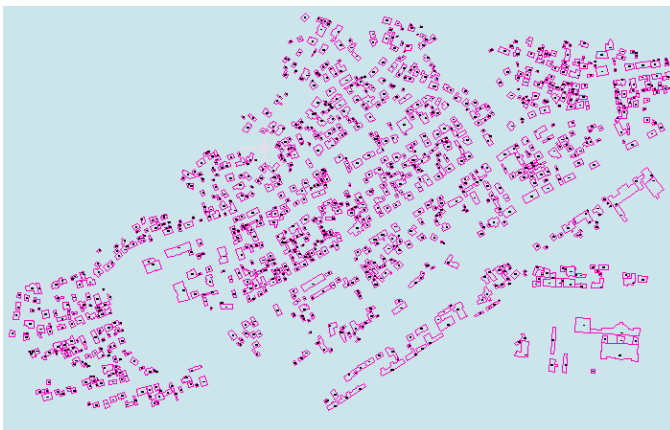
\*.dxf failuose esantys pradiniai duomenys:



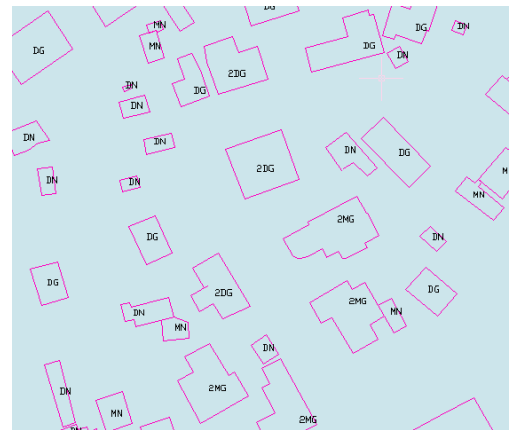
22 pav. Reljefas



23 pav. Gatvės

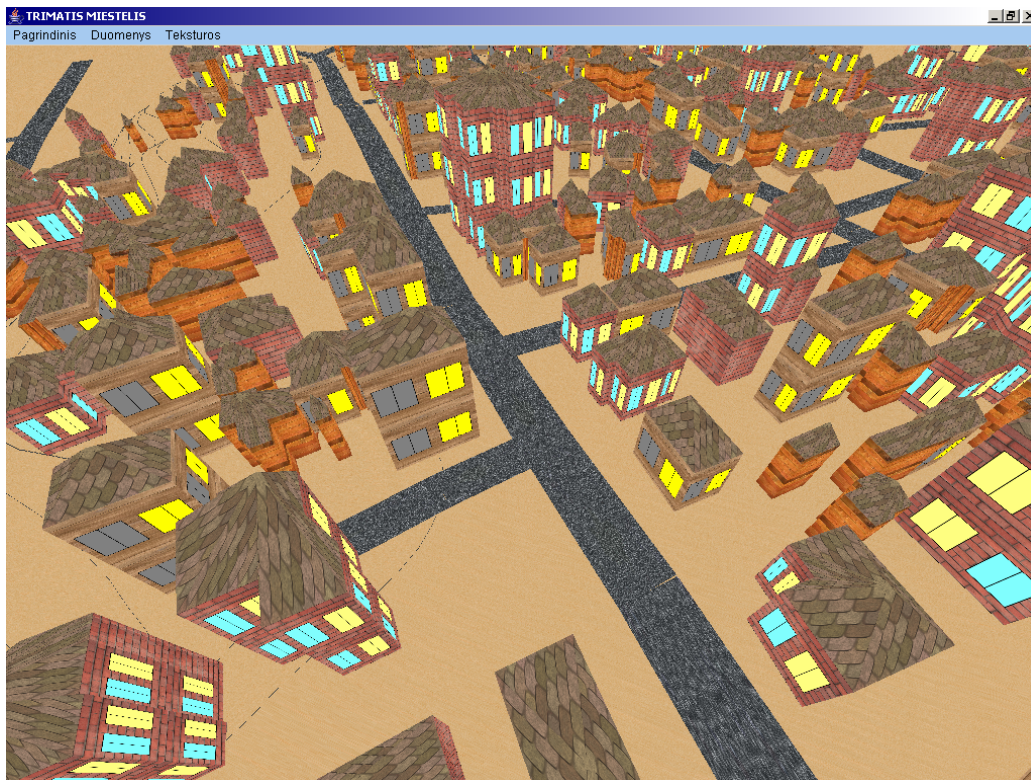


24 pav. Pastatai iš toli



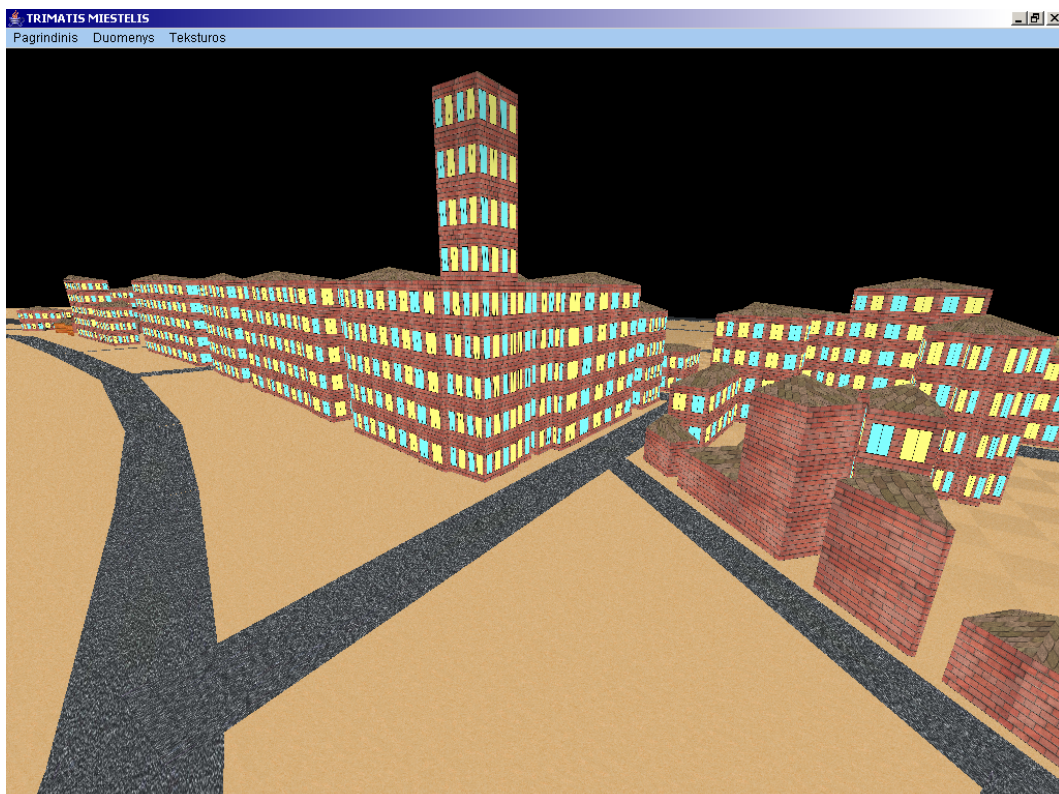
25 pav. Pastatai iš arti

Rezultatas: (vaizdas priartintas)



**26 pav.** Erdvinis miesto vaizdas iš arti

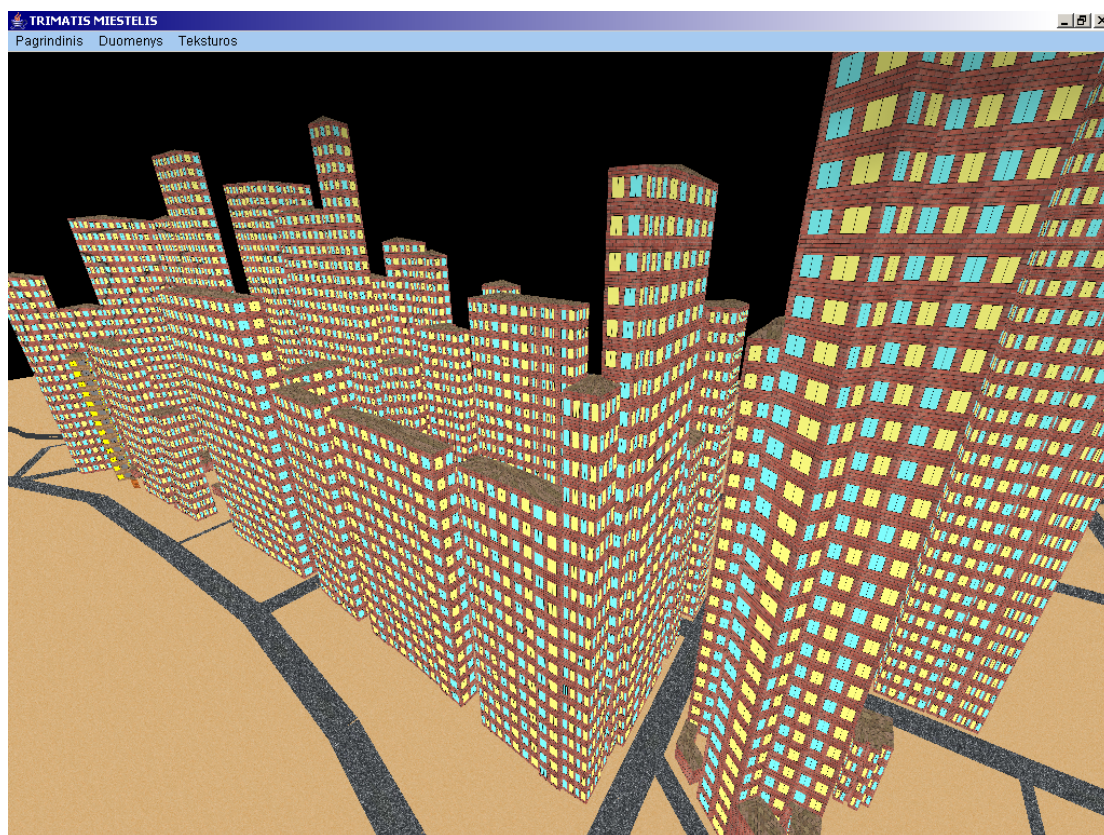
Taip pat pažvelgsime kaip vartotojas gali pakeisti miesto vaizdą. Paimame Kauno miesto dalį ir ją sugeneruojame. Gautas vaizdas (žvelgiant nuo Karaliaus Mindaugo pr. ir S. Daukanto gatvių sankirstos):





**27 pav.** Kauno miesto dalis prieš duomenų keitimą

O dabar pažvelkime kaip galėtų atrodyti ši miesto dalis ateityje:



**28 pav.** Kauno miesto dalis po duomenų keitimo

Taip pat ši sistema ateityje gali būti lengvai išplečiama. Yra numatytas naujų funkcijų diegimas bei jau esamų tobulinimas. Sistema bus tobulinama atsižvelgiant į šiuos pagrindinius bruožus: kokybę ir patikimumą, patogumą.

## 6. IŠVADOS

Šiame darbe buvo išanalizuoti magistrinio darbo „Erdvinio miesto generavimas pagal topografinius duomenis“ ypatumai, t.y.:

- atlikta darbo sisteminė analizė;
- parengtas programinės įrangos projektas;
- pateikta programinės įrangos architektūra;
- sukurta programinė įranga;
- sudaryta vartotojo dokumentacija.

Projekto kūrimo prioritetai: paprastumas ir kokybė. Programos paprastumą atspindi patogi ir lengvai įsisavinama vartotojo sąsaja. Pačią kokybę įvertins vartotojas pritaikęs šią programinę įrangą praktikoje, tačiau vis tik programos tobulinimui ribų tikrai nėra. Planuojama:

- didinti programinės įrangos universalumą bei suteikti vartotojui didesnę laisvę.
- realizuoti papildomas funkcijas, kurios leistų realiu laiku koreguoti, modifikuoti erdvinį vaizdą.
- išleisti programos versijas, kurios veiktų platesniame operacinių sistemų spektre.
- spartinti erdvinių vaizdų generavimo procesus.
- įvesti pradinių duomenų redagavimo galimybę.

Taigi sukurta programinė įranga siekia palengvinti tiek architektų, tiek geodezinikų darbą bei sumažinti vartotojo rūpesčius atliekant į trimačių vaizdų generavimą ir analizę.

## 7. LITERATŪRA

1. Autodesk - AutoCAD – Features [interaktyvus]. [žiūrėta 2006-02-15], prieiga per internetą: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5138869>.
2. Autodesk - Autodesk MAP 3D – Features [interaktyvus]. [žiūrėta 2006-02-18], prieiga per internetą: <http://www.pugh.co.uk/Products/autodesk/map3d-2007.htm>
3. Autodesk - Autodesk VIZ – Features [interaktyvus]. [žiūrėta 2006-02-18], prieiga per internetą: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5235467>
4. Autodesk – 3DS Max – Features [interaktyvus]. [žiūrėta 2006-02-24], prieiga per internetą: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5659451>
5. Graphisoft - ArchiCAD – Features [interaktyvus]. [žiūrėta 2006-02-23], prieiga per internetą: <http://www.graphisoft.com/products/archicad/ac10/demo/>
6. mb Software AG – ArCon – Features [interaktyvus]. [žiūrėta 2006-02-24], prieiga per internetą: [http://www.arcon-software.com/ArCon-International/arcon\\_standard.htm](http://www.arcon-software.com/ArCon-International/arcon_standard.htm)
7. SGI - OpenGL – Overview [interaktyvus]. [žiūrėta 2006-03-01], prieiga per internetą: <http://www.opengl.org/about/overview/>
8. Software Forge– LinuxCad – Features [interaktyvus]. [žiūrėta 2006-03-01], prieiga per internetą: <http://www.linuxcad.com/>

## **8. TERMINŲ IR SANTRUMPŲ ŽODYNAS**

1. JVM – Javos virtuali mašina, sluoksnis tarp programos ir operacinės sistemos (Java Virtual Machine).
2. Modelis(sistema) – erdvinis miesto vaizdas.
3. Objektas – miesto vaizdo dalis (pastatai, reljefas ir kt.).
4. Specifikacija - sistemos funkcionalumo aprašymas formaliais metodais.
5. Primityvas – taškas, linija ar daugiakampis.
6. OpenGL(Open Graphics Library) – grafinių komandų biblioteka
7. CAD(Computer Aided Design) – kompiuterių panaudojimas projektavimui.

## 9. SANTRAUKA ANGLŲ KALBA

### **Three-Dimensional City Generation Using Topographical Data**

#### Summary

Recently three-dimensional graphics is being applied in many fields. It is used in geodesy, topography and architecture as well. There are a lot of programs for the work with 2D and 3D graphics in the world. Most of them are related by CAD name. It could be mentioned such production of graphic software „giants“ like Autodesk (AutoCAD, Map 3D ) or Graphisoft (ArchiCAD). The aim of our work is the program which main function is the generation of three-dimensional city. Obviously this could be done using programs mentioned above, but the assimilation of such program needs much time and money. Therefore, if we need only to generate 3D city, our program is suitable for this.

Our program quickly retrieves topographic data of the city and generates its 3D view. The program is not overloaded with unnecessary functions. It makes the program easy to understand and assimilate for the user who has minimal knowledge in design and modelling.