

ŠIAULIŲ UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS KATEDRA

**Vytautas Kirstukas**

Informatikos specialybės IV kurso dieninio skyriaus studentas

**KOSMINIO STRATEGINIO ŽAIDIMO KŪRIMAS.  
VARTOTOJO SAŠAJOS PROJEKTAVIMAS IR  
REALIZAVIMAS**

**THE DEVELOPMENT OF SPACE STRATEGY GAME.  
DESIGN AND DEVELOPMENT OF USER INTERFACE**

BAKALAURO DARBAS

Darbo vadovas:

lekt. G. Lūža

Recenzentas:

lekt. dr. G. Felinskas

Šiauliai, 2012

*„Tvirtinu, jog darbe pateikta medžiaga nėra plagijuota ir paruošta naudojant literatūros sąraše pateiktus informacinius šaltinius bei savo tyrimų duomenis“*

Darbo autoriaus \_\_\_\_\_

(vardas, pavardė, parašas)

## **Darbo tikslai ir uždaviniai**

### **Tikslas**

Suprojektuoti ir realizuoti „3D Orion“ kosminio žingsninio strateginio žaidimo vartotojo sąsają prietaisams, naudojamiems „Android“ operacinei sistemai.

### **Uždaviniai**

- Išsiaiškinti „Android“ operacinės sistemos programų vartotojų sąsajų kūrimo ypatumus.
- Išsiaiškinti „3D Orion“ vartotojo sąsajos veikimo ypatumus ir poreikius.
- Suprojektuoti vartotojo sąsają „Android“ operacinei sistemai.
- Realizuoti suprojektuotą vartotojo sąsają.

Darbo vadovo \_\_\_\_\_

(vardas, pavardė, parašas)

# TURINYS

IVADAS .....	5
I. ANALITINĖ DALIS .....	6
1.1 Temos analizė.....	6
1.1.1 Operacinių sistemų skirtumai.....	6
1.1.2 „Android“ operacinės sistemos API lygiai .....	7
1.1.3 Mobilųjų prietaisų programų langai „Android“ operacinėje sistemoje .....	7
1.1.4 Statinis langų ir jo elementų išdėstymas „Android“ prietaisų ekranuose ...	8
1.1.5 Dinaminis elementų įkėlimas į vartotojo sąsajos langus.....	9
1.2 Darbinės srities analizė .....	11
1.2.1 „3D Orion“ žaidimo vartotojo sąsajos apimtis.....	11
1.2.2 „3D Orion“ žaidimo vartotojo sąsajos langai.....	11
1.3 Darbinės srities modelis .....	13
II. PROJEKTINĖ DALIS .....	14
2.1 Įrankių ir priemonių pasirinkimas .....	14
2.1.1 Trumpai apie „Android SDK“ .....	15
2.2 Pradinis projektas .....	15
2.3 Darbų planas.....	18
III. REALIZACINĖ DALIS .....	19
3.1 Darbų atlikimo atskaita .....	19
3.2 Galutinis projektas .....	19
3.2.1 Klasių diagrama .....	19
3.2.2 Veiksmų diagrama .....	20
3.2.3 Būsenų diagrama.....	21
3.2.4 Vartotojo sąsajos langai .....	21
3.3 Problemų sąrašas .....	25
3.4 Darbo rezultatų analizė .....	26
3.4.1 Pradinis ir galutinis projektai .....	26
3.5 Rekomendacijos .....	27
IŠVADOS .....	28
LITERATŪROS IR INFORMACINIŲ ŠALTINIŲ SĄRAŠAS .....	29
ANOTACIJA .....	31
SUMMARY .....	31
PRIEDAI.....	32

## IVADAS

Sparčiai tobulėjant mobiliosioms technologijoms, asmeninių kompiuterių programos, taip pat ir žaidimai persikelia į mobiliuosius prietaisus. Ne paslaptis, kad šiuolaikiniuose išmaniuosiuose telefonuose ar planšetiniuose kompiuteriuose galėtų sklandžiai veikti asmeniniams kompiuteriams 2000 – 2005 metais kurti žaidimai, jei jie būtų perkurti mobilių prietaisų operacinėms sistemoms. Vienas iš pavyzdžių, žaidimas „Grand Theft Auto III“, 10 metų sukakties proga, 2011 metais perkurtas „iOS“ operacinę sistemą naudojantiems mobiliesiems prietaisams. Ne išimtis ir „3D Orion“ kosminis strateginis žaidimas, kurį taip pat galima būtų kurti mobiliesiems prietaisams, nors pradinėje stadijoje šis žaidimas buvo orientuotas į asmeninių kompiuterių platformas.

„3D Orion“ vartotojo sąsają planuojama kurti „Android“ operacinei sistemai. Šiuo metu ši operacinė sistema yra dominuojanti tarp mobiliųjų operacinių sistemų.

„3D Orion“ žaidimo vartotojo sąsaja susideda iš didelio kiekio langų, todėl šiame darbe siekiama apsiriboti tik dalimi jų, išskirti svarbiausius. Siekiama, kad projektuojami vartotojo sąsajos langai turėtų funkcionalumą: būtų atvaizduojami „Android“ virtualiame įrenginyje bei pasiekiami ir iškviečiami vienu ar kitu keliu. Šiame darbe nesiekama sukurti ypatingo vartotojo sąsajos stiliaus, taip pat nesiekama sukurti vartotojo sąsajos sąveikaujančios su kitais „3D Orion“ žaidimo programiniais moduliais.

# I. ANALITINĖ DALIS

## 1.1 Temos analizė

„3D Orion“ – tai mokomasis projektas, dvimatės vartotojo sąsajos kūrimas „3D Orion“ žaidimui. Siekiama studentams suteikti praktinių užduočių, neatitrūkstant nuo tikrovės ir įgyvendinant ne pavienes užduotis, o kuriant projektus, besitęsiančius kelis studijų semestrus. Kiekvienas projektas skaidomas į mažesnes projektines dalis. Šiuo atveju „3D Orion“ žaidimo kūrimo projektinės dalys yra: žaidimo variklis (angl. *Game Engine*), grafinis variklis (angl. *Graphics Engine*), grafinė vartotojo sąsaja (angl. *User Interface*), dirbtinio intelekto variklis (angl. *AI Engine*). Visos dalys yra pakankamai didelės darbų atlikimo trukmės atžvilgiu, todėl jos skaidomos į mažesnes dalis. [6]

Grafinė vartotojo sąsaja reikalinga išreikšti žaidimo idėją, atvaizduoti duomenis, sąveikauti su vartotoju.

„3D Orion“ vartotojo sąsaja kuriama lygiagrečiai su kitomis žaidimo projektinėmis dalimis. Siekiama nustatyti, ar galima sukurti pilnavertišką vartotojo sąsają, neturint kitų projektinių dalių. Kitaip sakant, sukurti nepriklausomą vartotojo sąsają.

Darbe siekiama suprojektuoti ir realizuoti „3D Orion“ žaidimo vartotojo sąsają „Android“ operacinę sistemą naudojantiems mobiliesiems prietaisams. Bandoma atsakyti į klausimą, ar galima struktūriškai didelį žaidimą perteikti į mažus ekranus turinčius mobiliuosius prietaisus.

Darbe norima išgauti vartotojo sąsajos funkcionalumą, o ne ypatingą grafinį įvaizdį.

### 1.1.1 Operacinių sistemų skirtumai

Asmeninių kompiuterių operacinės sistemos, tokios kaip „Windows NT“, gali vykdyti neribotą kiek programų vienu metu, tuo tarpu mobilių prietaisų operacinės sistemos – tik keletą programų, iš kurių tik viena yra aktyvi, matoma ekrane. Taip yra dėl šių prietaisų didelių skirtumų. Mobiliuosiuose prietaisuose gerokai mažesni atminties resursai, lyginant su šiuolaikiniais asmeniniais kompiuteriais. Todėl grįžtame į XX a., kai asmeninių kompiuterių programų efektyvus atminties naudojimas būdavo privalumas. Taip yra dabar su mobiliųjų telefonų programomis.

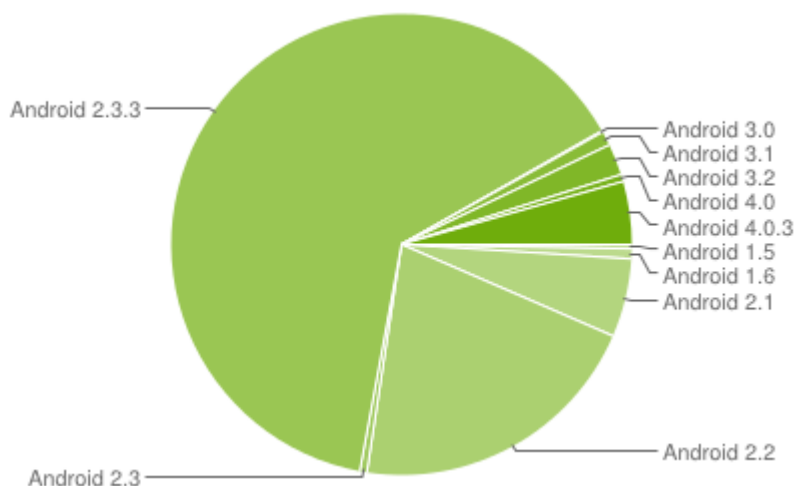
„Android“ operacinė sistema prižiūri joje veikiančias programas. Vartotojas palikdamas programos langą, jos neišjungia. Pagal nutylėjimą programa išlieka atmintyje. Todėl trūkstant atminties resursų, operacinė sistema pašalina seniausią naudotas programas, jų vartotojo sąsajos langus.

Kai asmeniniuose kompiuteriuose vienu metu galime matyti ir naudotis keliais programos(-ų) langais, tai mobiliuosiuose įrenginiuose apsiribojama su vienu langu. Pagalbiniai langai (angl. *dialog*) taip pat egzistuoja tarp mobilių prietaisų programų, tačiau ir tie patys, užima visą mobiliojo prietaiso ekraną, t. y. pagrindinis langas matomas, tačiau jis yra užtamsintas ir neaktyvus iki tol, kol bus iškvieistas pagalbinis langas.

### 1.1.2 „Android“ operacinės sistemos API lygiai

API (*application programming interface*) lygis – tai sveiko skaičiaus reikšmė, kuri identifikuoja „Android“ operacinės sistemos versiją (angl. *framework*). Šiuo metu išleista 15 API lygių. Kuo aukštesnis API, tuo platesnė biblioteka, tačiau su kai kuriais lygiais keičiasi ir vartotojo sąsajos elementų išvaizda (mygtukai, tekstų laukai ir t. t.). Kiekvienas lygis apima žemesnius lygius, t. y., tarkim 15 lygis, turi 1-14 lygių bibliotekas. [1]

„Android“ kūrėjų 2012 gegužės 1d. duomenimis, tarp šiuo metu „Android“ operacinę sistemą naudojančių prietaisų visame pasaulyje labiausiai paplitusi „Android“ 2.3.3 versija (10 API) (žr. 1-ąją pav.). [2]



1 pav. „Android“ operacinės sistemos versijos paplitimas 2012-05-01

### 1.1.3 Mobilųjų prietaisų programų langai „Android“ operacinėje sistemoje

Kiekvienai programai mobiliajame prietaise „Android“ operacinė sistema išskiria tik jai priklausančią steką (angl. *task*), kuriame talpinami iškviešti tos programos langai. Pereinant iš vieno programos lango į kitą, pagal nutylėjimą, buvęs langas nukeliamas į programos langų steką, tačiau įmanoma keisti šią nustatytą tvarką. Kiekvienas „Android“ operacinės sistemos prietaisas turi mygtuką „Atgal“, kurį nuspaudus, grįžtama atgal, į paskiausiai į steką nukeltą langą, sunaikinant, ištrinant iš atminties dabartinį langą. Langai gali būti sukuriami statiškai arba dinamiškai. [3]

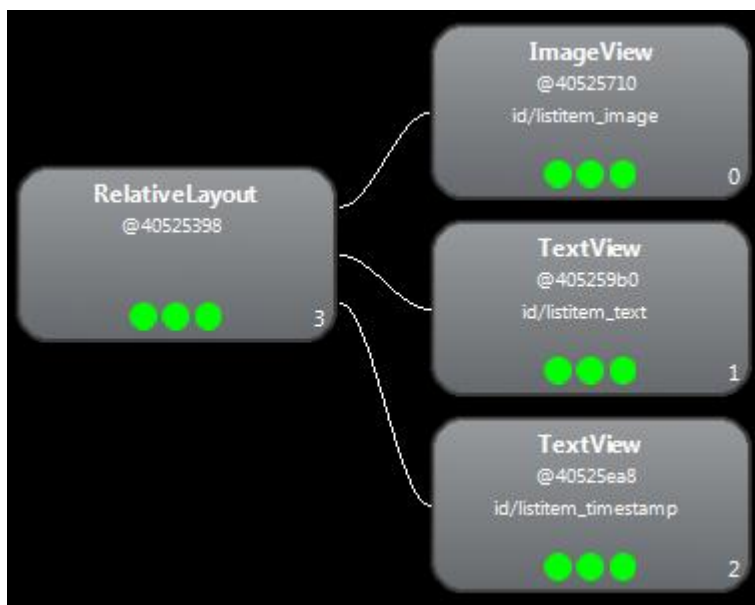
### 1.1.4 Statinis langų ir jo elementų išdėstymas „Android“ prietaisų ekranuose

Langas, jo elementų išdėstymo planas (angl. *layout*) ir patys elementai aprašomi xml duomenų faile. Kiekvienas langas turi savo šakninį elementą - elementų išdėstymo planą, kuriame gali būti talpinami kiti elementų išdėstymo planai arba elementai. Komponuojant elementus į visumą, susidaro elementų hierarchija, kuri gali stipriai nulemti sklandų programos veikimą.

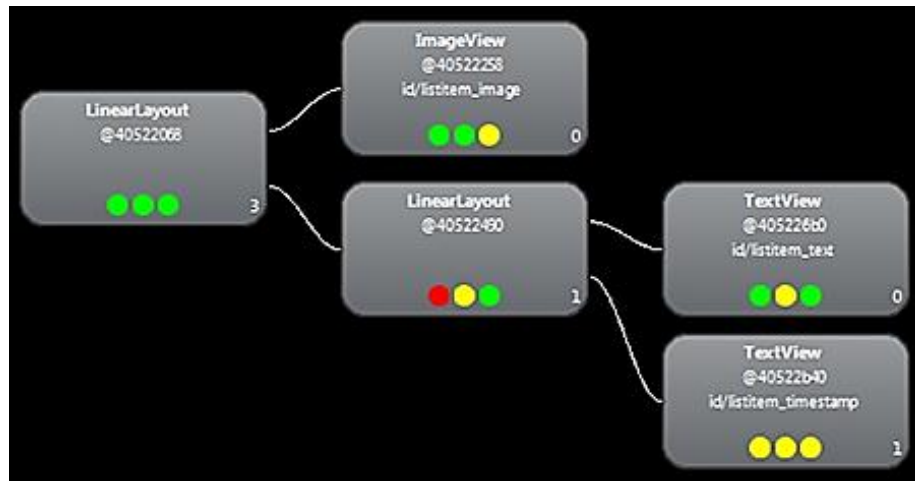
„Prastai“ sukurtas langas gali labai ženkliai padidinti prietaiso darbinės atminties naudojimą. Tai lems lėtą vartotojo sąsają. Panašiai kaip sudėtingą elementų hierarchiją turintis internetinis puslapis kraunasi lėtai, taip ir su „Android“ programų vartotojo sąsajomis.

Neteisingas manymas, kad naudojant įprastus išdėstymo planus, galima sukurti labai efektyvią vartotojo sąsają. Kiekvienas ekrano įtaisas (angl. *widget*) ar išdėstymo planas (angl. *layout*) prieš atsiradamas prietaiso ekrane atlieka tokius veiksmus: išskiria vietą savo elementams, juos talpina į tą vietą ir tada atvaizduoja (angl. *draw*) į ekraną. Talpinant išdėstymo planus vieną į kitą galima sukurti „gilią“ hierarchiją. Kiekviename hierarchijos lygyje vartotojo sąsajai teks atlikti prieš tai išvardintus veiksmus. Kuo „gilesnė“ hierarchija, tuo labiau ilgėja vartotojo sąsajos atvaizdavo laikas. Ypač negerai, kai persipinantys išdėstymo planai atvaizduojami dinamiškai daug kartų per labai trumpą laiko tarpą. Didelė tikimybė, kad vartotojo sąsaja užstrigs arba nebereaguos į vartotojo veiksmus.

Kiekvieno lango elementų hierarchija turėtų būti kiek įmanoma „platesnė“ (žr. 2-ąjį pav.), o ne „gilesnė“ (žr 3-įjį pav).



2 pav. Vartotojo sąsajos lango "platesnė" hierarchija.



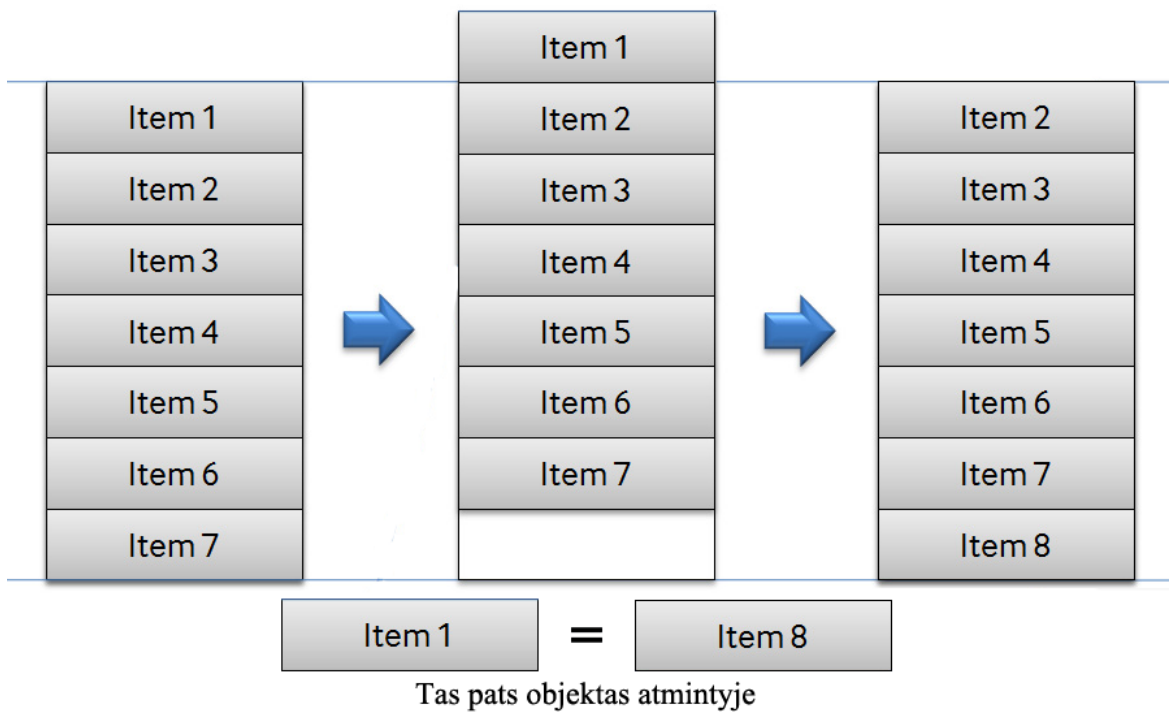
3 pav. Vartotojo sąsajos lango "gilesnė" hierarchija.

### 1.1.5 Dinaminis elementų įkėlimas į vartotojo sąsajos langus

Dažnai dinamiškai atvaizduojami elementai gali tiesiog „suvalgyti“ prietaiso darbinę atmintį, jeigu jie kiekvieną kartą nuskaitomi iš xml failo.

„Android“ programų vartotojo sąsajose dažnai sutinkamas ListView komponentas. Tai „Android SDK“ bibliotekos siūlomas komponentas, atliekantis elementų sąrašo funkciją ir galintis talpinti didelį kiekį elementų. „Android“ kūrėjai pasirūpino, kad šie sąrašai kiek įmanoma efektyviau (atvaizduotų) savo vidinius elementus, tačiau dalis efektyvumo priklauso ir nuo programuotojo.

ListView sąrašo ypatybė – atvaizduoti tik tiek įrašų, kiek yra matoma prietaiso ekrane. Visiems kitiems įrašams „Android“ operacinė sistema vietos atmintyje neišskiria. Slenkant šiuo sąrašu žemyn, pirmasis įrašas pradingsta, tuo metu atsiranda naujas. Naujasis – tai tas pats pradingęs pirmasis įrašas, tik su pakeistomis reikšmėmis, sudarantis įspūdį, kad tai naujas objektas atmintyje (žr. 4-ąjį pav.).



4 pav. ListView komponento veikimas

Programuotojo įdirbis: sukurti vidinę statinę klasę ir aprašyti joje ListView eilutės elementus. Pirmą kartą inicijuojant ListView eilutę, jos elementai priskiriami sukurtos statinės klasės objekto kintamiesiems. Vykstant pakartotiniam sąrašo eilutės atvaizdavimui ar jos elementų reikšmių keitimui, nebereikia kreiptis į jos aprašymą xml faile.

## 1.2 Darbinės srities analizė

### 1.2.1 „3D Orion“ žaidimo vartotojo sąsajos apimtis

„3D Orion“ žaidimo vartotojo sąsają galima išskirti į dvi dalis:

- vartotojo sąsaja prieš žaidimo eigą;
- vartotojo sąsaja žaidimo eigoje;

Šiame darbe apsiribota projektuoti ir realizuoti tą vartotojo sąsają, kuri reikalinga prieš pradėdant žaidimo sceną. Ši dalis yra svarbesnė pradėdant žaidimą, kuriant jo pasaulį, vystant „3D Orion“ projektą.

Pasirinktoje dalyje vartotojo sąsaja turėtų suteikti galimybę susikurti savo vartotoją, redaguoti žaidimo pasaulį ir jo objektus, keisti įvairius žaidimo nustatymus. Pagrindinis privalumas yra kurti savo žaidimo objektus ir juos naudoti žaidime, taigi šiuo atveju reikalingi žaidimų objektų redaktoriai – kiekvienam iš jų išvaizda, t. y. vartotojo sąsaja.

### 1.2.2 „3D Orion“ žaidimo vartotojo sąsajos langai

Projektuojant žaidimą buvo sumanyta, kad didžiausi žaidimo pasaulio objektai – galaktikų spiečiai, susidarantys iš galaktikų, turinčių žvaigždžių sistemas. Žvaigždžių sistemas sudaro planetų sistemos, kurios sudarytos iš planetų ir jos palydovų. Planetos ar jų palydovai skaidomi į regionus, kuriuose įkurtos vienos ar kitos civilizacijos kolonijos.

Dėl labai didelio žaidimo objektų kiekio ir ribotos mobiliųjų prietaisų atminties, kuriant vartotojo sąsają mobiliems įrenginiams, apsiribota žaidimo pasauliu laikyti vieną žvaigždžių sistemą su joje esančiais objektais. Įvertinant „3D Orion“ žaidimo kūrimo stadiją, šiuo metu vartotojo sąsajoje norima apjungti pagrindinius ir didžiausius žaidimo objektus (žr. 5 pav.).

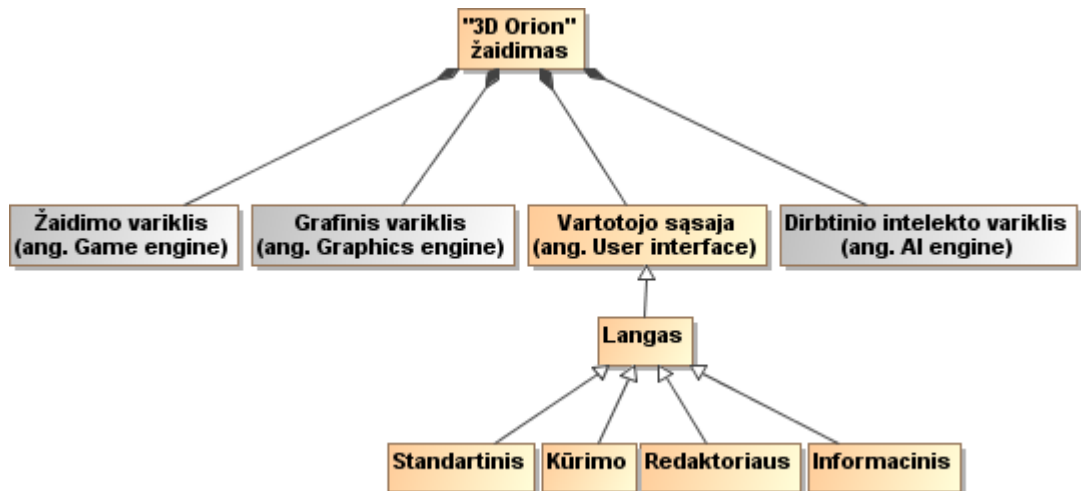
Išanalizavus langų poreikį, juos galima suskirstyti į grupes:

- Standartiniai – sutinkami dažniausiai kiekviename kompiuteriniame žaidime, tokie kaip pagrindinis meniu ar žaidimo parametrų keitimas.
- Kūrimo – norint pradėti žaidimą, užtenka pasirinkti jau sukurtus objektus.
- Redaktoriai – jau minėti langai, kurie leidžia kurti, keisti ir trinti žaidimo objektus, taip pat jų parametrus.
- Informaciniai – langai, kurie pateikia išsamią informaciją apie vieną ar kitą objektą.



### 1.3 Darbinės srities modelis

„3D Orion“ žaidimo modelis, programiniu atžvilgiu, susidaro iš kelių pagrindinių dalių, įskaitant ir vartotojo sąsają (žr. 6-ąjį pav.).



6 pav. Darbinės srities modelis

## II. PROJEK TINĖ DALIS

### 2.1 Įrankių ir priemonių pasirinkimas

Vartotojo sąsajai suprojektuoti pasirinkta „Magic Draw UML“ priemonė. Ši priemonė pasirinkta dėl šių privalumų:

- Pritaikyta JAVA kalbos sintaksei.
- Turima patirtis dirbant su šia priemone.
- Studijų metu mokoma ir rekomenduojama dirbti su šia priemone.

Taip pat pasirinkimą lėmė, kad ši priemonė visiškai patenkina poreikius. Be to, kitos UML diagramų braižymo priemonės, pagal savo siūlomas galimybes, yra panašios į šią.

Suprojektuotai vartotojo sąsajai pasirinktas „Android SDK” įrankis ir „eclipse“ programavimo aplinka. „Android SDK“ – įrankių paketas, leidžiantis kurti „Android“ operacinei sistemai skirtas programas.

#### **„Android SDK“ įrankis pasirinktas dėl šių privalumų:**

- nemokamas ir nemokama dokumentacija;
- labai išsami dokumentacija;
- daug išsamios ir bet kokio lygio programuotojui – tiek pradedančiajam, tiek profesionalui – tinkančios informacijos apie „Android“ platformą ir jos programų kūrimą.

#### **„eclipse“ programavimo aplinka pasirinkta dėl šių kriterijų:**

- suderinama su „Android SDK“;
- nemokama;
- patogiai naudotis;
- galimybė individualizuoti aplinką asmeniniam patogumui;
- patirtis realizuojant studijų metu užduotas praktines užduotis įvairiomis kalbomis, įskaitant ir užduotis atliktas sukūriant „Android“ programas;
- „Google“ ir didžioji dalis „Android“ mokomųjų knygų programų pavydžių yra sukurta su „eclipse“.

Taip pat yra ir kitų platformų su savais įrankiais ir ypatumais, leidžiančiais kurti programas ne vien „Android“ operacinei sistemai, tačiau ir kitoms mobiliųjų prietaisų operacinėms sistemoms. Vienas iš tokių įrankių paketų yra „Titanium SDK“, kurio pagrindinis įrankis Java-script kalba. „Titanium SDK“ leidžia kurti programas „Android“, „iOS“ operacinėms sistemoms, taip pat HTML5 internetines svetaines. Tačiau „Titanium SDK“

su savo programavimo aplinka labiau skirtas įmonėms, užsiėmusiomis mobilių programų kūrimu.

### 2.1.1 Trumpai apie „Android SDK“

„Google“ kūrėjai sukūrė galingą ir nemokamą įrankių paketą – „Android SDK“, kurį sudaro:

- įrankiai, leidžiantys mobilių programų kūrėjams kurti jas „Android“ platformai;
- kiti įvairūs įrankiai, padedantys programuotojui diagnozuoti problemas ir jas ištaisyti. Vienas iš jų, „Hierarchy Viewer“, leidžia sugeneruoti lango elementų hierarchiją (žr. 2-ąjį ir 3-įjį pav.) ir matyti papildomą informaciją;
- atviro kodo mokomieji projektai;
- virtualūs prietaisai (angl. *emulator*), leidžiantys išbandyti sukurtas programas;
- bibliotekos, reikalingos „Android“ programų kūrimui;

Programos rašomos JAVA kalba ir vykdomos „Android“ prietaiso „Dalvik“ virtualiojoje mašinoje, kuri veikia Linux branduolyje. „Android“ kūrėjai „Android SDK“ jungia kartu su nemokama programavimo aplinka „eclipse“.

## 2.2 Pradinis projektas

Šiame darbe orientuojamasi suprojektuoti ir realizuoti šiuos pagrindinius langus:

- Standartiniai:
  - Pagrindinio meniu (pradinis žaidimo startavimo) langas.
  - Žaidimo išsaugojimo (angl. *save game*) ir įkėlimo (angl. *load game*) langas.
  - Žaidimo meniu (žaidimą sustabdžius jo eigoje) langas.
  - Žaidėjo meniu langas.
  - „Apie“ langas.
  - Pagalbos langas.
- Kūrimo langai:
  - Norimos žvaigždžių sistemos (kurioje vyktų visas žaidimas) pasirinkimo.
  - Norimos civilizacijos (už kurią žaistų vartotojas) pasirinkimo.
  - Norimos planetų sistemos ir vietos joje (kur būtų įkurdinta civilizacijos pradinė kolonija) pasirinkimo.

- Redaktorių langai:
  - Žvaigždžių sistemos.
  - Planetų sistemos.
  - Planetų sistemos objektų (planetų ir asteroidų žiedų).
  - Rasių.
  - Civilizacijų.
  - Kolonijų.
  - Kompleksų.
  - Įmonės.
  - Regionų.
  - Laivyno.
  - Projekto.
- Informaciniai langai:
  - Planetų sistemos.
  - Planetų sistemos objektų (planetų ir asteroidų žiedų).
  - Civilizacijos.
  - Komplexo.
  - Įmonės.
  - Sluoksniu (angl.. *dwelling*).
  - Regiono.
  - Rasės.
  - Laivyno.
  - Laivo.
  - Pasaulio civilizacijos.

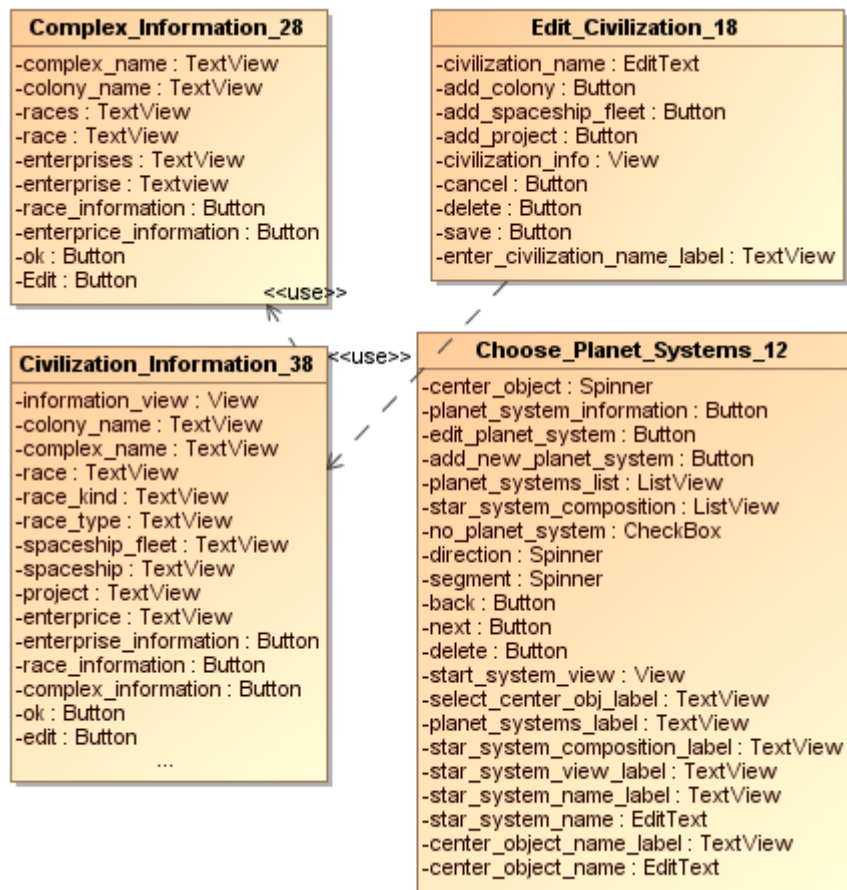
Visi langai yra susieti tarpusavyje, t. y. yra pasiekiami vienu ar kitu keliu. Į kai kuriuos langus galima patekti iš kelių skirtingų langų. Visi langai orientuoti į gulsčią (angl. *landscape*) mobiliojo prietaiso padėtį. Pakeičiant vartotojo sąsajos orientaciją, langai išlieka gulsčioje orientacijoje.

Remiantis temos analize:

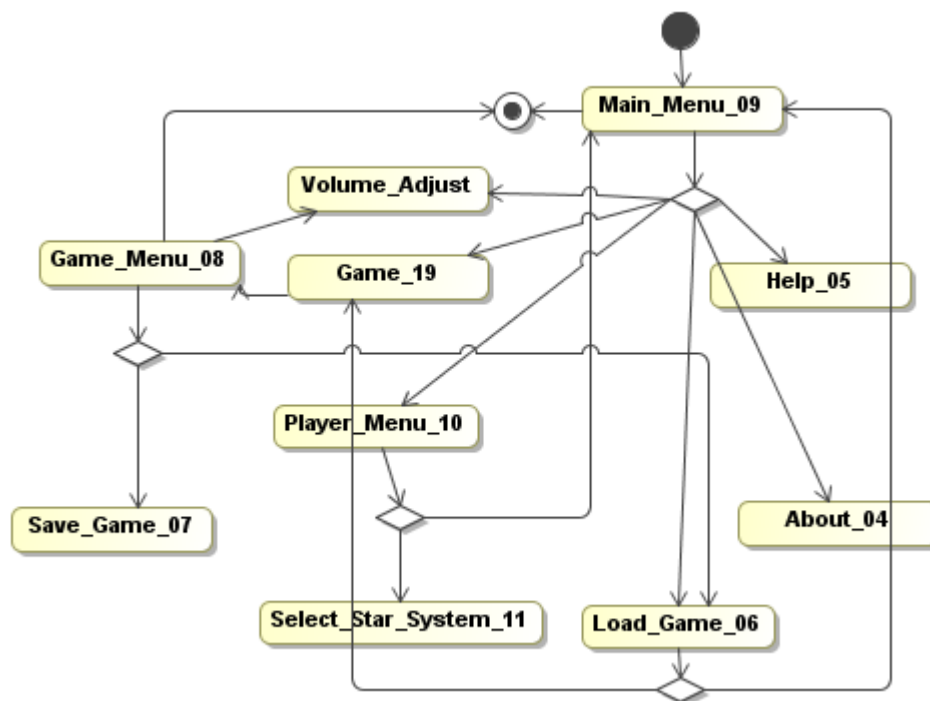
- vartotojo sąsaja turėtų būti suderinama su „Android“ operacinės sistemos 2.3.3 versija, kadangi šių ir vėlesnes versijas naudojančių prietaisų pasaulyje daugiau nei 70%;
- statiskai aprašytų langų elementų išdėstymo hierarchija turėtų būti kuo „platesnė“;
- ListView komponento naudojimas turėtų būti efektyvus;

Aprašymo darbo priede pateikiamas pilnas pradinio projekto UML:

- Klasių diagrama (žr. 7-ąją pav.), kur klasę atitinka vartotojo sąsajos langas. Preliminarus langų elementų sąrašas surašytas vietoje klasės argumentų.
- Būsenų diagrama (žr. 8-ąją pav.), kur būseną atitinka vartotojo sąsajos langas. Ryšiai tarp būsenų – variantai, į kokius langus galima patekti.



7 pav. Pradinio projekto langų, jų elementų diagramos dalis



8-asis pav. Perėjimai į vartotojo sąsajos langus diagramos dalis

## 2.3 Darbų planas

		Vasaris		Kovas				Balandis				Gegužė				Birželis				
		Savaitės																		
Nr.	Val. sk.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	10																			
2	50																			
3	20																			
4	50																			
5	150																			
6	10																			
7	30																			
<b>Iš viso:</b>	320																			
<b>Lentelėje numerius atitinkantys darbai:</b>																				
1	Konsultacijos su darbo vadovu																			
2	„Android“ operacinės sistemos programų vartotojų sąsajų analizė																			
3	„3D Orion“ žaidimo vartotojo sąsajos analizė																			
4	Vartotojo sąsajos projektavimas																			
5	Vartotojo sąsajos realizavimas																			
6	Vartotojo sąsajos testavimas																			
7	Darbo aprašymo rašymas																			

1 lentelė. Darbų planas

### III. REALIZACINĖ DALIS

#### 3.1 Darbų atlikimo atskaita

		Vasaris		Kovas				Balandis					Gegužė				Birželis			
		Savaitės																		
Nr.	Val. sk.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	6																			
2	50																			
3	20																			
4	50																			
5	130																			
6	14																			
7	50																			
<b>Iš viso:</b>	320																			
<b>Lentelėje numerius atitinkantys darbai:</b>																				
1	Konsultacijos su darbo vadovu																			
2	„Android“ operacinės sistemos programų vartotojų sąsajų analizė																			
3	„3D Orion“ žaidimo vartotojo sąsajos analizė																			
4	Vartotojo sąsajos projektavimas																			
5	Vartotojo sąsajos realizavimas																			
6	Vartotojo sąsajos testavimas																			
7	Darbo aprašymo rašymas																			

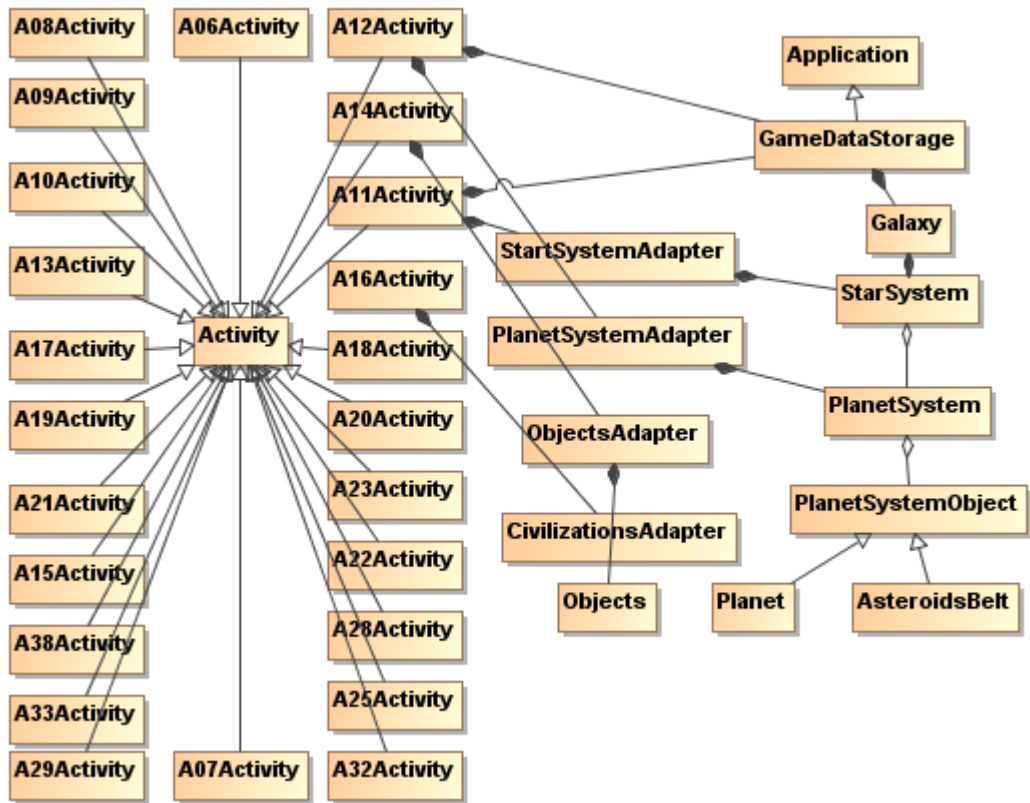
2 lentelė. Darbų atlikimo atskaita

#### 3.2 Galutinis projektas

##### 3.2.1 Klasių diagrama

Sukurta klasių diagrama pateikiama darbo prieduose. Dalis diagramos pavaizduota 9-ame paveikslėlyje. Klasės:

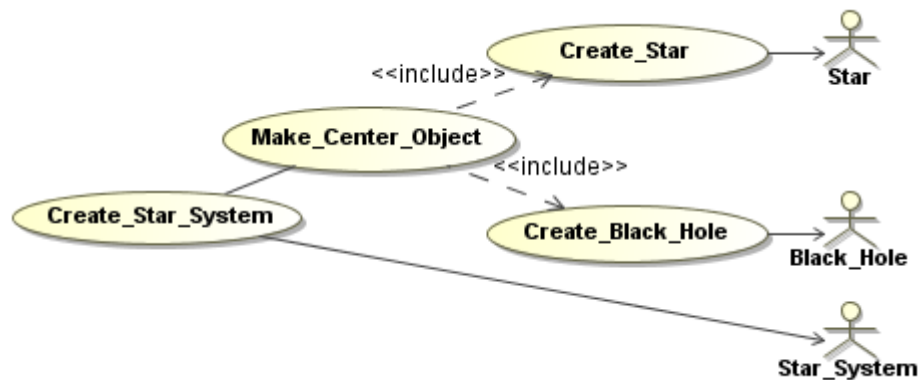
- Sunumeruotos Activity – vartotojo sąsajos langai. Numeracija atitinka langų sąrašo, pateikto žemiau 3.1.4 skyrelyje, numerius.
- Objektų ir jų adapterių (angl. *adapter*).
- Modifikuotų lango elementų, praplečiant esamų langų elementų klases.



9 pav. Galutinio projekto klasių diagramos dalis

### 3.2.2 Veiksmų diagrama

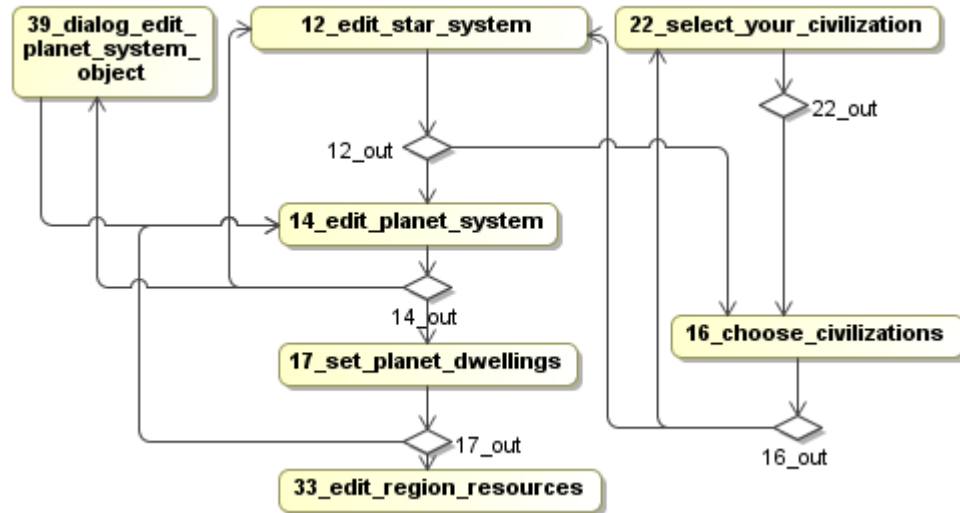
Sukurta veiksmų diagrama pateikiama darbo prieduose. Dalis diagramos pavaizduota 10-ame paveikslėlyje. Diagrama rodo, ką, naudojantis sukurta vartotojo sąsaja, žaidėjas gebėtų atlikti.



10 pav. Galutinio projekto veiksmų diagramos dalis

### 3.2.3 Būsenų diagrama

Sukurta būsenų diagrama pateikiama darbo prieduose. Dalis diagramos pavaizduota 11-ame paveikslėlyje, kur būsenos – vartotojo sąsajos langai. Būsenų ryšiai nurodo galimus perėjimus iš vieno lango į kitą.



11 pav. Galutinio projekto būsenų diagramos dalis

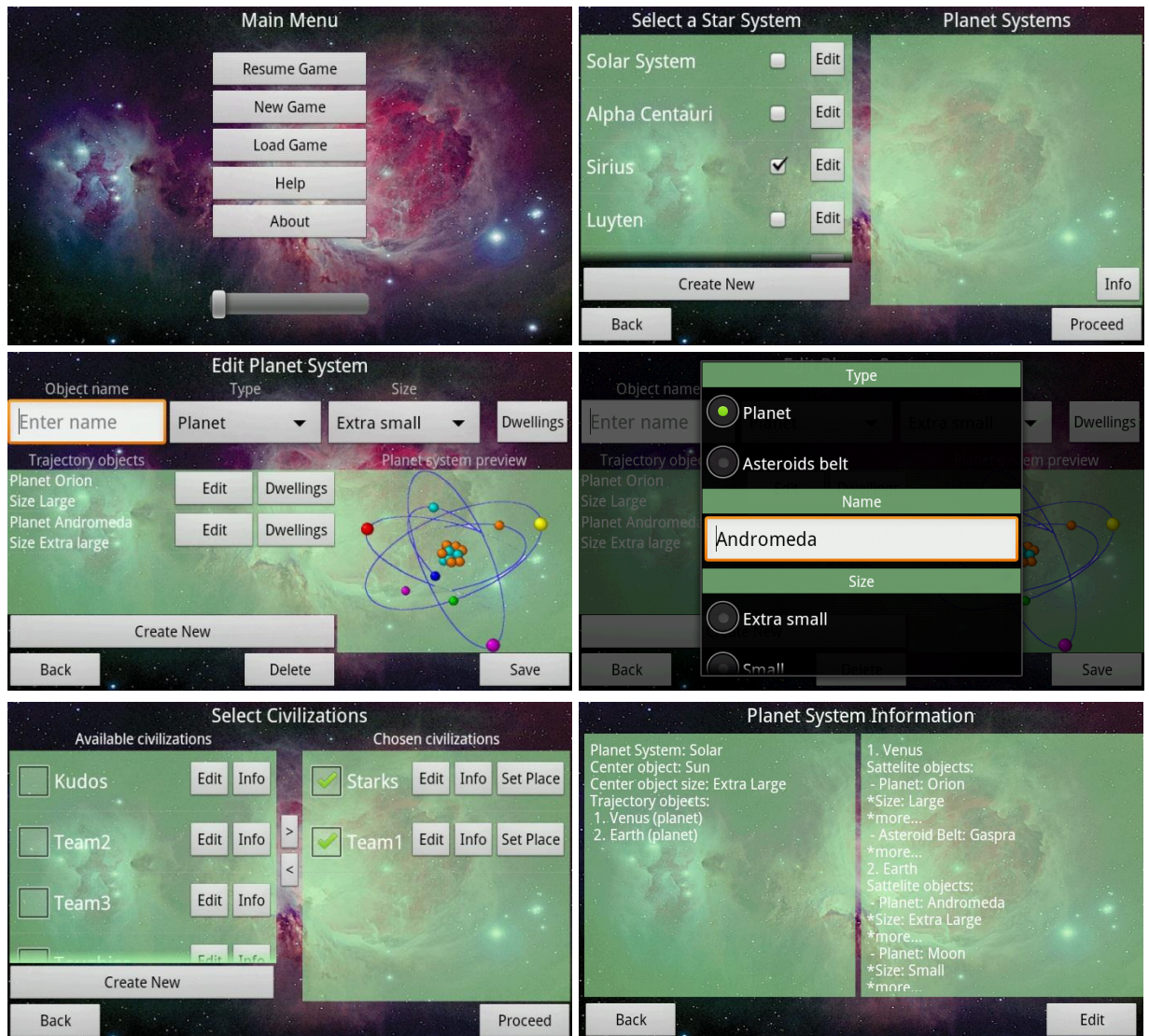
### 3.2.4 Vartotojo sąsajos langai

Suprojektuoti ir realizuoti šie langai:

- **a06\_load\_game.** Suteikia galimybę įkelti išsaugotą žaidimo sesiją.
- **a07\_save\_game.** Suteikia galimybę išsaugoti žaidimo sesiją.
- **a08\_game\_menu.** Žaidimo sesijos metu iškviečiamas pagalbinis meniu.
- **a09\_main\_menu.** Žaidimo inicializacijos pirmasis langas.
- **a10\_player\_menu.** Leidžia susikurti vartotoją ir pasirinkti žaidimo sunkumo lygį.
- **a11\_select\_star\_system.** Leidžia pasirinkti jau sukurtą žvaigždžių sistemą, kurioje vyktų žaidimo scena. Leidžia sukurti naują ar redaguoti esamą žvaigždžių sistemą. Taip pat rodo pažymėtos žvaigždžių sistemos planetų sistemų pagrindinę informaciją bei leidžia pažiūrėti išsamesnę informaciją pereinant į planetų sistemos informacinį langą.
- **a12\_edit\_star\_system.** Galima keisti žvaigždžių sistemos parametrus bei sandarą.
- **a13\_planet\_system\_infomation.** Rodo pilną informaciją apie planetų sistemą.
- **a14\_edit\_planet\_system.** Galima keisti planetų sistemos parametrus bei sandarą.

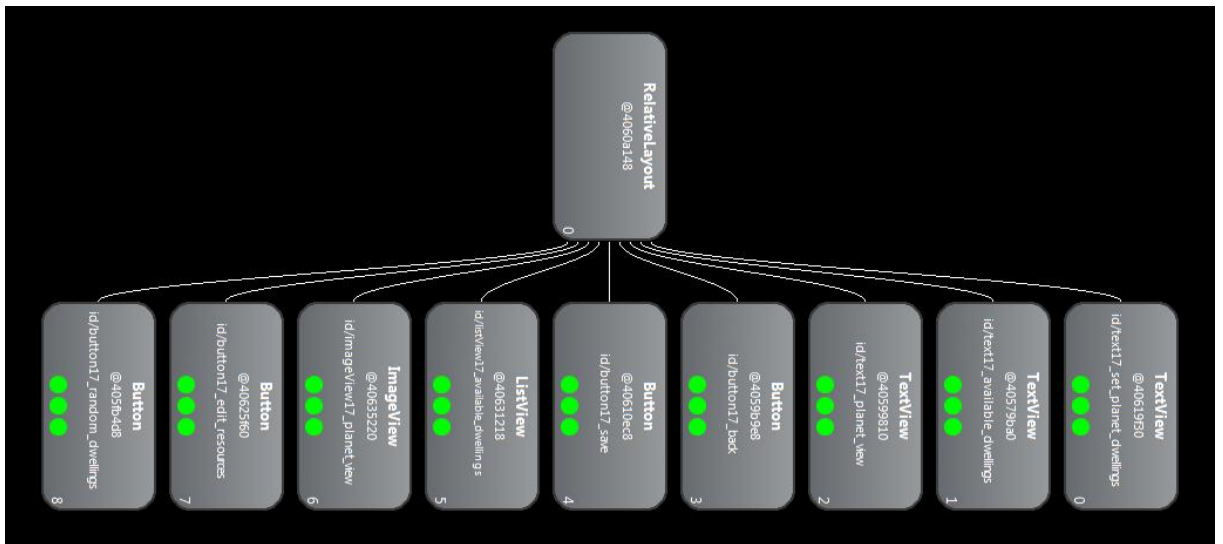
- **a15\_planet\_system\_object\_information.** Rodo pilną informaciją apie planetų sistemos objektą.
- **a16\_choose\_civilizations.** Leidžia priskirti civilizacijas, redaguojamai žvaigždžių sistemai ir iškviesti civilizacijos vietos langą. Galimybė sukurti naują ar redaguoti esamą civilizaciją. Galima iškviesti pasirinktos civilizacijos informacinį langą.
- **a17\_set\_planet\_dwelling.** Galimybė nustatyti planetos sandarą iš sluoksnių (angl. *dwelling*s).
- **a18\_edit\_civilization.** Galimybė keisti civilizacijos parametrus, pridėti koloniją ir kitus objektus. Rodoma pilna informacija apie civilizaciją.
- **a19\_game\_screen.** Žaidimo scenos langas.
- **a20\_choose\_place\_for\_colony.** Priskiriama civilizacijos pradinė kolonija, pasirinktos planetų sistemos objektui.
- **a21\_edit\_race.** Galimybė sukurti, keisti, pasirinkti rasės tipą, rūšį ir pačią rasę. Taip pat priskirti rasei, jos tipui ir rūšiai, įvairius gebėjimus (angl. *skills*).
- **a22\_select\_your\_civilization.** Leidžia pasirinkti žaidėjo atstovaujamą civilizaciją iš pateikto žvaigždžių sistemos civilizacijų sąrašo. Galimybė keisti sąrašą.
- **a23\_choose\_region.** Priskiriami civilizacijos kolonijos kompleksams ar laivynams planetų sistemos objekto regionai.
- **a25\_edit\_colony.** Galimybė keisti kolonijos parametrus ir pridėti, kurti, keisti kompleksus, žiūrėti jų informaciją.
- **a28\_complex\_information.** Rodo pilną informaciją apie civilizacijos kolonijos kompleksą.
- **a29\_edit\_complex.** Galimybė keisti komplekso parametrus ir pridėti, kurti, keisti įmones ir gyvenančią rasę, žiūrėti jų informaciją.
- **a32\_race\_information.** Rodo pilną informaciją apie civilizacijos kolonijos kompleksą valdančią rasę.
- **a33\_edit\_region\_resources.** Galimybė nustatyti planetos sistemos objekto sluoksnio regiono pradinį resursų kiekį.
- **a38\_civilization\_information.** Rodo pilną informaciją apie civilizaciją.
- **a39\_dialog\_edit\_planet\_system\_object.** Pagalbinis (angl. *dialog*) langas, leidžiantis keisti, sukurti planetų sistemos objektą ir jo parametrus.

Keletas realizuotų vartotojo sąsajos langų pavyzdžių:

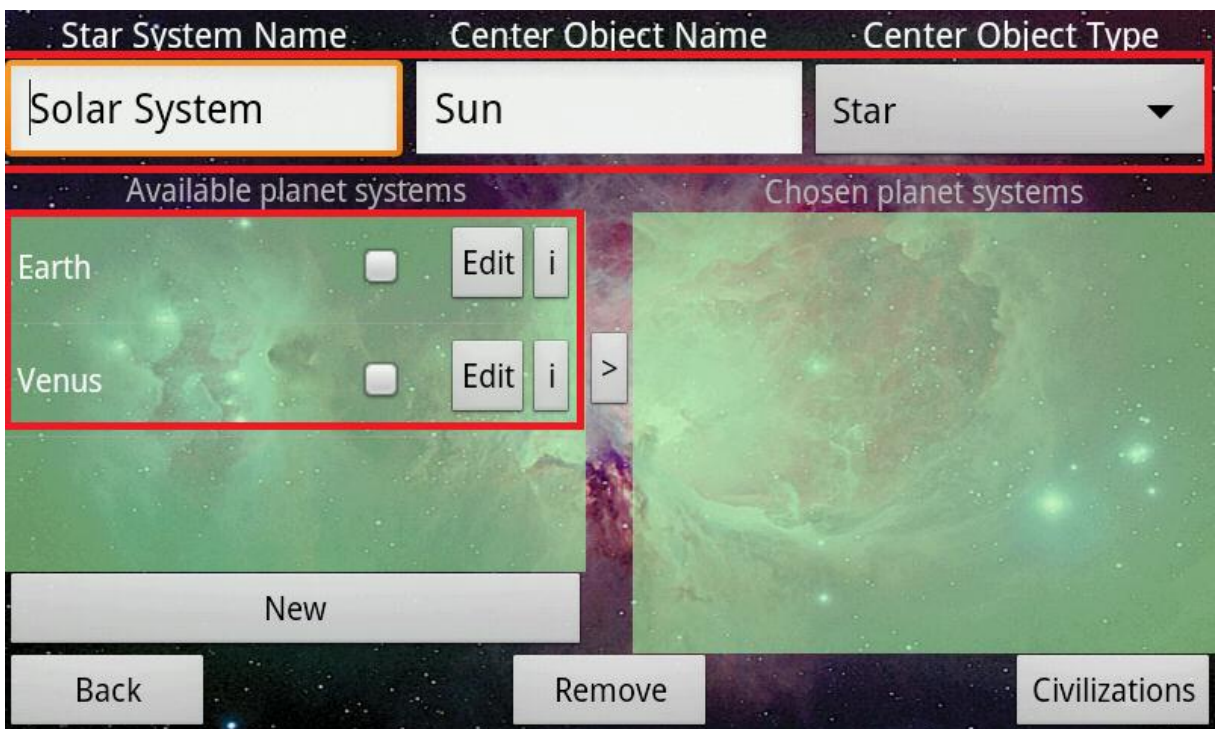


Langų ypatybės:

- Visi langai orientuoti gulsčiai „Android“ prietaiso padėčiai.
- Langų elementų išsidėstymo hierarchijos dydį stengtasi kaip įmanoma sumažinti (žr. 12 pav).
- Kai kuriems langams sukurti vietiniai pavyzdiniai duomenys (žr. 13 pav. Apibraukti duomenys).
- Visiems vartotojo sąsajos ListView komponentams, kuriems sukurti vietiniai duomenys, įgyvendintas efektyvus, temos analizėje aprašytas, jų elementų reikšmių keitimas.
- Visiems langų elementams suteikti unikalūs identifikaciniai pavadinimai, pagal struktūrą: <tipas><lango\_nr>\_<elemento\_pavadinimas>.



12 pav. a17\_set\_planet\_dwelling langos hierarchija



13 pav. a12\_edit\_star\_system langas ir jo vietiniai duomenys

### 3.3 Problemų sąrašas

**Navigacija tarp programos langų steko.** Uždarant (užbaigiant) vartotojo sąsajos langą, grįžtama į prieš tai buvusį langą, t. y. esantį steko pradžioje. Kai kuriais atvejais, reikia grįžti į langą, esantį steko antroje ar tolimesnėje pozicijoje. „Android SDK“ biblioteka nesuteikia galimybių peršokti į norimą programos langą esantį steke. Sprendimas - gražinti rezultatą į kviečiantįjį langą iš užbaigto lango. Pagal rezultatą kviečiantysis langas gali apdoroti gražinamą rezultatą ir, tenkinant reikšmės sąlygą, užsidaryti. Apibendrinant, šiuo būdų grįžtama iš eilės per steko langus, tačiau kai kuriomis sąlygomis vienas ar kitas langas užbaigiamas ir neparodomas, prieš jo parodymo iniciavimą.

**Mygtuko „atgal“ paspaudimas žaidimo scenos metu.** Nuspaudus mygtuką „atgal“, esamas langas susinaikintų ir būtų iškviečiamas prieš tai buvęs. Žaidžiant žaidimą, toks mygtuko veiksmas tiesiog sunaikintų visą žaidimą ir jį tektų pradėti žaisti iš naujo. Sprendimas - perrašyti (angl. *override*) „atgal“ mygtuko elgesį žaidimo scenos lango metu, kad jo nuspaudimas iškvieštų pagalbinių meniu (a08\_game\_menu).

**Duomenų trūkumas.** Ne visada vartotojo sąsaja susidaro tik iš statiška aprašytų elementų. „3D Orion“ vartotojo sąsajos languose naudojamuose ListView komponentuose atvaizduojama informacija yra susijusi su žaidimo pasaulio objektais, tačiau kartais reikia atvaizduoti ne tik informaciją, bet ir papildomus vartotojo sąsajos elementus. Šiuo metu „3D Orion“ žaidimo projekte nėra realizuota jokia duomenų bazė su joje esančiais žaidimo objektais, todėl atvaizduoti nėra ko. Sprendimo būdai: sukurti duomenų bazę, duomenų failą arba vietinius, programos vykdymo metu sukuriamus, duomenis. Šiame darbe buvo nuspręsta pasirinkti vietinius duomenis, kadangi tai yra sparčiau nei juos rašyti į duomenų failus, o duomenų bazė nėra šio darbo tikslas ar uždavinys. Duomenys padeda sudaryti tikresnę vartotojo sąsajos įspūdį ir leisti ją pilnai realizuoti.

## 3.4 Darbo rezultatų analizė

### 3.4.1 Pradinis ir galutinis projektai

**Pradinis projektas.** Pradiniame projekte aprašyta, kokie langai planuojami suprojektuoti ir realizuoti. Buvo bandoma aprašyti vartotojo sąsajos langų sudėtį, kokie preliminarūs elementai turėtų juos sudaryti. Taip pat, kokie keliai galimi norint pereiti iš vieno vartotojo sąsajos lango į kitą.

**Galutinis projektas.** Galutiniame projekte langų kiekis buvo sumažintas. Dėl duomenų konkretumo trūkumo ir pačių duomenų nebuvimo, smulkesnių „3D Orion“ žaidimo pasaulio objektų redaktorių ir informacinių langų projektavimas ir realizavimas neįmanomas. Atsisakyta šių langų:

- Standartinių:
  - Apie
  - Pagalbos
- Redaktorių:
  - Įmonės
  - Laivyno
  - Projekto
- Informacinių:
  - Įmonės
  - Sluoksniu (angl. *dwelling*)
  - Regiono
  - Laivyno
  - Laivo
  - Pasaulio civilizacijos

Galutiniame projekte klasių diagrama sudaryta iš langų klasių, kuriose nėra aprašyti jų elementai, kadangi statiniai elementai aprašomi projekto langų xml failuose. Skirtingai nuo pradinio projekto klasių diagramos, atsirado papildomos klasės: objektų, praplėstų lango elementų, objektų adapterių. Praplėstieji elementai panaudoti keliuose ListView komponentuose. Objektų ir jų adapterių klasės reikalingos ListView komponentams, atvaizduoti žaidimo pasaulio objektų informaciją bei papildomiems vartotojo sąsajos elementams. Taip pat objektų klasės pasitarnautų ateityje, kai duomenys būtų gauti iš duomenų bazės bei lokaliai patalpinami klasių objektuose.

Atsisakius langų, sumažėjo būsenų diagramos apimtis. Pridėta naujų ryšių, kurie pradiniam projekte nebuvo pastebėti. Keičiant langų struktūrą ir perėjimo tarp jų logiką buvo pašalinta dalis ryšių.

Sukurta veiksmų diagrama, parodanti, ką žaidėjas teoriškai galėtų atlikti su vartotojo sąsaja, kokius objektus sukurti ar pasižiūrėti jų informaciją, jei egzistuotų „3D Orion“ žaidimo pasaulio objektų duomenų bazė.

Apjungiant projektavimo ir realizacijos darbus, sukurta „MS Excel“ lentelė, kurioje pateiktas sukurtų langų sąrašas su duomenimis, kurie rodo, į kokį langą galima patekti bei pateikia variantus, kaip tą langą galima pasiekti.

### **3.5 Rekomendacijos**

Igyvendinant darbo tikslą ir užduotis, kilo nemažai idėjų, susijusių su šiuo darbu ir „3D Orion“ projektu.

„Android“ operacinė sistema naudojama įvairių ekranų dydžių ir raiškos prietaisuose. Tam, kad būtų supaprastintas programų pritaikymas šiai įvairovei, „Android“ kūrėjai ekranų dydžius suskirstė į 4 kategorijas: small, medium, large, extra-large ir ekranų raišką suskirstė į 3 kategorijas: ldpi (low), mdpi (medium), hdpi (high). Šiuo atveju „3D Orion“ vartotojo sąsają realizuoti reiktų 4 skirtingiems dydžiams, bei multimediją (paveiksliukus ir panašius dalykus) - trim skirtingom raiškom.

Šiuo metu gana sparčiai auga naujausios „Android“ operacinės sistemos (4.0.3 versijos) paplitimas, taigi įvertinant „3D Orion“ projekto vystymosi progresą, galima šį žaidimą orientuoti šiai naujausiai versijai, kadangi po kelerių metų ši operacinė sistema turėtų užimti nemažą rinkos dalį.

Vartotojo sąsają toliau vystyti rekomenduojama tik tada, kai būtų sukurta objektų duomenų bazė arba bent 95% visų žaidimo objektų apibrėžtis būtų aiški. Tai yra, kad suprojektavus ir realizavus langą, kuriame galima nustatyti objekto 10 parametrų, vėliau neatsirastų dar 20. Taip pat privalo būti sukurti vartotojo sąsajos ir jos elementų stiliai, nes jie nulemia elementų išsidėstymą ir jų kiekį lange. Vartotojo sąsajos kūrimas – lėtas procesas, todėl perdarinėti vartotojo sąsają būtų netikslinga ne tik programuotojui, bet ir kitų specialybių žmonėms.

## IŠVADOS

- „Android“ programų vartotojo sąsajos kūrimas – brangus procesas. Todėl tai turėtų būti vienas iš paskutinių etapų programų sistemų kūrime, kai visa kita jau suprojektuota ir realizuota.
- Kuriant vartotojo sąsają lygiagrečiai su pačiu projektu, vartotojo sąsajos visiškai realizuoti neįmanoma: trūksta duomenų, žaidimo variklio, grafinių elementų.
- Greitai „Android“ programų ir jų projektų peržiūrai, geriausia viską pateikti virtualioje mašinoje, kurioje iki galo įdiegtas ir sukonfigūruotas „Android SDK“ įrankių paketas.

## LITERATŪROS IR INFORMACINIŲ ŠALTINIŲ SĄRAŠAS

1. *Android API Levels* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą: [developer.android.com/guide/appendix/api-levels](http://developer.android.com/guide/appendix/api-levels)
2. *Platform Versions* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą: [developer.android.com/resources/dashboard/platform-versions](http://developer.android.com/resources/dashboard/platform-versions)
3. *Tasks and Back Stack* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą: [developer.android.com/guide/topics/fundamentals/tasks-and-back-stack](http://developer.android.com/guide/topics/fundamentals/tasks-and-back-stack)
4. *Optimizing Layout Hierarchies* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą: [developer.android.com/training/improving-layouts/optimizing-layout](http://developer.android.com/training/improving-layouts/optimizing-layout)
5. *Android developers* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą: [developer.android.com](http://developer.android.com)
6. KIRSTUKAS, Vytautas. Šiaulių universitetas. *Kosminio žingsninio strateginio žaidimo kūrimas. Efektyvių žaidimo objektų generatorių kūrimas*: Kursinis darbas. Šiauliai: 2011. 22 p.
7. *Kosminio žingsninio strateginio žaidimo kūrimas: projekto aprašymas*. Šiaulių universiteto informatikos katedra, 2010.
8. „3D Orion“. *Koncepcinis žaidimo modelis*. Šiaulių universiteto informatikos katedra, 2010.
9. „3D Orion“. *Klasių ir veiksmų diagramos*. Šiaulių universiteto informatikos katedra, 2011.
10. VOGEL, Lars. *Android ListView and ListActivity – Tutorial* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą: [vogella.com/articles/AndroidListView/](http://vogella.com/articles/AndroidListView/)
11. VOGEL, Lars. *Android Intents - Tutorial* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą: [vogella.com/articles/AndroidIntent/](http://vogella.com/articles/AndroidIntent/)
12. *Custom ListView with ability to check items* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą: [marvinlabs.com/2010/10/custom-listview-ability-check-items/](http://marvinlabs.com/2010/10/custom-listview-ability-check-items/)
13. GARGENTA, Marko. *Learning Android*. Jungtinės Amerikos valstijos, 2011, 268 p. ISBN 978-1-449-39050-1.
14. *Vartotojo sąsajos projektavimas* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą: [kopustas.elen.ktu.lt/studentai/\\_media/16.vartotojo.sasaja](http://kopustas.elen.ktu.lt/studentai/_media/16.vartotojo.sasaja)
15. *Android Tools: Leveraging the Hierarchy Viewer for UI Profiling* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą:

[mobile.tutsplus.com/tutorials/android/android-tools-leveraging-the-hierarchy-viewer-for-ui-profiling/](http://mobile.tutsplus.com/tutorials/android/android-tools-leveraging-the-hierarchy-viewer-for-ui-profiling/)

16. GARGENTA. Marko. *Android Bootcamp Series 2012* [interaktyvus]. [žiūrėta 2012 m. gegužės 2 d.]. Prieiga per internetą: [youtube.com/playlist?list=PLE08A97D36D5A255F&feature=plcp](http://youtube.com/playlist?list=PLE08A97D36D5A255F&feature=plcp)

## **ANOTACIJA**

Autorius: Vytautas Kirstukas

Tema: Kosminio strateginio žaidimo kūrimas. Vartotojo sąsajos projektavimas ir realizavimas.

Šiaulių universitetas 2012.

Darbas skirtas sukurti „3D Orion“ kosminio strateginio žaidimo vartotojo sąsają „Android“ operacinei sistemai. Darbo metu buvo apibrėžta žaidimo vartotojo sąsajos langų sritis, kuriai įvykdytas projektavimas ir realizavimas. Darbai atlikti optimizuojant vartotoją sąsają, pagal tam tikras, darbe aprašytas taisykles.

## **SUMMARY**

Author: Vytautas Kirstukas

Subject: The Development of Space Strategy Game. Design and Development of User Interface.

Šiauliai University, 2012.

The purpose of this work is to develop user interface of „3D Orion“ space strategy game for „Android“ operating system. The range of user interface windows was defined, designed and implemented during the course of this work. Also it was done in an efficient way by following some rules that were defined in this work.

## **PRIEDAI**

1. Kompaktinis diskas. Turinys:
  - 1.1. Bakalauro darbo aprašymas.
  - 1.2. „Oracle VM VirtualBox“ programos instaliacijos failas „VirtualBox-4.1.16-78094-Win.exe“.
  - 1.3. „Oracle VM VirtualBox“ virtualios mašinos atvaizdo paketas „WinXP\_Android.ova“.
  - 1.4. Naudojimosi instrukcija.
  - 1.5. Programos projektas skirtas „eclipse“ programavimo aplinkai su įdiegtu „Android SDK“ papildiniu.
  - 1.6. Programos projekto diagramos: paveikslėliai ir „MagicDraw UML“ projektai.
  - 1.7. Vartotojo sąsajos langų perėjimo lentelė.
  - 1.8. Vartotojo sąsajos langų paveikslėliai.
  - 1.9. „3D Orion“ kosminio strateginio žaidimo aprašymas, klasių ir veiksmų diagramos (žr. [7], [8], [9] nr. lit. šaltinius).