



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Marius Kimbrauskas

**Objektų fizikinių poveikių modeliavimo naudojant spyruoklių metodą
tyrimas ir pritaikymas mobiliems įrenginiams**

Baigiamasis magistro projektas

Vadovas

Lekt. dr. Šarūnas Packevičius

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

TVIRTINU

Katedros vedėjas
Doc. dr. Tomas Blažauskas

**Objektų fizikinių poveikių modeliavimo naudojant spyruoklių metodą
tyrimas ir pritaikymas mobiliems įrenginiams**

Baigiamasis magistro projektas
Programų sistemų inžinerija (621E16001)

Vadovas

Lekt. dr. Šarūnas Packevičius

Recenzentas

Prof. dr. Alfonsas Misevičius

Projektą atliko

Marius Kimbirauskas

KAUNAS, 2015



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Marius Kimbirauskas

(Studento vardas, pavardė)

Programų sistemų inžinerija, 621E16001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Objektų fizikinių poveikių modeliavimo naudojant spyruoklių metodą tyrimas ir pritaikymas mobiliems įrenginiams“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 15 m. Gegužės 22 d.
Kaunas

Patvirtinu, kad mano **Mariaus Kimbirausko** baigiamasis projektas tema „Objektų fizikinių poveikių modeliavimo naudojant spyruoklių metodą tyrimas ir pritaikymas mobiliems įrenginiams“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Kimbirauskas, M. Objektų fizikinių poveikių modeliavimo naudojant spyruoklių metodą tyrimas ir pritaikymas mobiliems įrenginiams. Magistro baigiamasis projektas / vadovas lekt. dr. Šarūnas Packevičius; Kauno technologijos universitetas, Informatikos fakultetas, Programų inžinerijos katedra.

Kaunas, 2015. 48 psl.

SUMMARY

In this work it was analyzed in Master Courses studies designed software - a game in which the body deformation algorithm is implemented. In this work was implemented the spring deformation algorithm. The analysis of this work was made an attempt to evaluate the spring deformation algorithm of detail and realism. It was also an attempt to evaluate performance algorithm and was an attempt to find suggestions, which would improve performance.

This work analyzed the game for mobile devices. This game is a racing game in which deformation algorithm implemented. The game's deformation algorithm is based on the springs' algorithms which is for deformations. In this work it was analyzed and an attempted to evaluate how the algorithm realized, how fast it acts, and how visually detail and accurately perform deformation calculations.

This work focuses on a single deformation algorithm – spring algorithm.

The analytical part provides an overview of techniques that are associated with the game. In this section it also described the spring deformation algorithm and its main idea.

Design section describes the key elements that have been used and to implement the game. Also there are essential parts of the technical documentation.

The research part is an attempt to visually evaluate game performance. This part has been trying to visually evaluate the work of deformations. An attempt was made to evaluate the deformation of detail and accuracy. It was also an attempt to evaluate the operation speed.

The final section attempts to quantify the operation of the deformation speed. During evaluation, was calculated time dependence on model detail. An experiment was repeated with reduced deformation level of detail, and the results were compared with each other.

Turinys

1. Įžanga	8
1.1. Santrauka	8
1.2. Tiriamoji problema	8
1.3. Tyrimo objektas	8
1.4. Darbo tikslas	8
1.5. Darbo uždaviniai	9
2. Analitinė dalis.....	9
2.1. Užduotis.....	9
2.2. Problemos	9
2.3. Žaidimo kūrimas.....	9
2.4. 3D atvaizdavimas	10
2.5. Daugiakampių tinklelis.....	10
2.6. Spyruoklių (Spring/Mass) metodas deformacijoms apskaičiuoti	11
2.7. Optimizavimas.....	13
2.8. Bibliotekos ir programų kūrimo sąsajos	14
2.9. Žaidimo varikliai	16
3. Projektinė dalis	16
3.1. Sistemos panaudos atvejų diagrama	17
3.1. Funkciniai reikalavimai	24
3.2. Sistemos statinis vaizdas	28
3.3. Detalizuotas sistemos statinis vaizdas	29
4. Tyrimo dalis.....	33
4.1. Tikslas.....	33
4.2. Vizualus deformacijų kokybės ir tikslumo vertinimas	33
4.3. Rezultatų apibendrinimas	37
5. Eksperimentinė dalis.....	37
5.1. Tikslas.....	38
5.2. Tyrimo aprašymas	38
5.3. Eksperimento įranga.....	38
5.4. Pradinio tyrimo rezultatai	39
5.4.1. Tyrimo rezultatai atlikti mobiliajame įrenginyje.....	39
5.4.2. Tyrimo rezultatai atlikti asmeniniame kompiuteryje.....	40
5.5. Patobulintos PĮ tyrimo rezultatai	41
5.5.1. Tyrimo rezultatai atlikti mobiliajame įrenginyje.....	42
5.5.1. Tyrimo rezultatai atlikti asmeniniame kompiuteryje.....	43
5.6. Rezultatų analizė	43
6. Išvados.....	45
7. Literatūra	46
8. Terminų ir santrumpų žodynas	48

Paveikslėlių turinys

Paveikslėlis 1 Daugiakampių rūšys: A – trikampis, B – keturkampis, C - daugiakampis.....	10
Paveikslėlis 2 Trimatis objektas sudarytas iš daugiakampių[3]	11
Paveikslėlis 3 Spyruoklių metodas	13
Paveikslėlis 4 Panaudos atvejų diagrama	17
Paveikslėlis 5 Sistemos statinis vaizdas	29
Paveikslėlis 6 Žaidimo branduolys.....	29
Paveikslėlis 7 Paketas Graphic System	30
Paveikslėlis 8 Garso sistema	30
Paveikslėlis 9 Paketas Physic system	31
Paveikslėlis 10 Paketas Input System.....	31
Paveikslėlis 11 Paketas vartotojo sąsaja	32
Paveikslėlis 12 Paketas File System.....	32
Paveikslėlis 13 Paketas Data Object.....	33
Paveikslėlis 14 Deformacijų sukėlimas iš viršaus	34
Paveikslėlis 15 Deformacijų iš viršaus vizualus vertinimas	34
Paveikslėlis 16 Defromacijų sukėlimas automobilio gale	35
Paveikslėlis 17 Atliktų deformacijų automobilio gale vizualus vertinimas	35
Paveikslėlis 18 Deformacijų sukėlimas automobilio priekyje.....	36
Paveikslėlis 19 Deformuoto automobilio priekio vizualus įvertinimas.....	37

Lentelių turinys

Lentelė 1 Spyruoklių algoritmo privalumai ir trūkumai.....	13
Lentelė 2 Naujas žaidimas.....	24
Lentelė 3 Žaidimo logikos sustabdymas	24
Lentelė 4 Rezultatų atvaizdavimas	25
Lentelė 5 3D grafika.....	25
Lentelė 6 Deformacija	25
Lentelė 7 Nauji vartotojai.....	26
Lentelė 8 Vartotojo duomenų korekcija	26
Lentelė 9 Vartotojų šalinimas.....	26
Lentelė 10 Vartotojo informacijos atvaizdavimas	26
Lentelė 11 Žaidimo išsaugojimas	27
Lentelė 12 Žaidimo užkrovimas	27
Lentelė 13 Žaidimo objektas	27
Lentelė 14 Deformacijos pradžia.....	28
Lentelė 15 Žaidimo fizika	28
Lentelė 16 Mobiliojo telefono parametrai	39
Lentelė 17 Asmeninio kompiuterio parametrai	39
Lentelė 18 Rezultatai mobiliajame telefone	39
Lentelė 19 Rezultatai asmeniniame kompiuteryje.....	40
Lentelė 20 Patobulintos PĮ rezultatai asmeniniame kompiuteryje.....	42

1. ĮŽANGA

1.1. Santrauka

Šiame darbe, buvo analizuojamas magistrantūros studijų metu sukurta programinė įranga – žaidimas, kuriame realizuotas kūno deformacijų algoritmas. Šiame darbe buvo realizuotas spyruoklių deformacijų algoritmas. Analizuojant šį darbą, buvo bandoma įvertinti spyruoklių algoritmo deformacijų detalumą ir realistiškumą. Taip buvo bandoma įvertinti algoritmo veikimo spartą, ir surasti pasiūlymus, kurių pagalba, būtų galima pagerinti veikimo spartą.

Šis darbas analizavo mobiliems įrenginiams skirtą žaidimą. Šis žaidimas yra lenktynių žaidimas, kuriame realizuotas deformacijų algoritmas. Žaidimo deformacijos remiasi spyruoklių algoritmu, skirtu deformacijoms. Šiame darbe buvo analizuojama ir bandoma įvertinti, kaip realizuotas algoritmas, kaip sparčiai jis veikia ir kaip vizualiai detalai ir tiksliai atlieka deformacijos skaičiavimus.

Šis darbas koncentruojasi į vieną deformacijų algoritmą – spyruoklių algoritmą.

Analitinėje dalyje apžvelgiamos technologijos, kurios yra susijusios su sukurtu žaidimu. Taip šioje dalyje apibūdinamas spyruoklių deformacijos algoritmas.

Projektinėje dalyje aprašomi svarbiausi elementai, kurie buvo panaudoti ir realizuoti sukūrus žaidimą. Taip pat pateiktos esminės techninės dokumentacijos dalys.

Tyrimo dalyje bandoma vizualiai įvertinti žaidimo veikimą. Šioje dalyje buvo bandoma vizualiai įvertinti, kaip pateikiamos deformacijos. Buvo bandoma įvertinti deformacijų detalumą ir tikslumą. Taip buvo bandoma įvertinti veikimo spartą.

Paskutiniame skyriuje bandoma kiekybiškai įvertinti deformacijos veikimo spartą. Vertinant spartą, buvo skaičiuojama laiko priklausomybė nuo modelių detalumo. Atliktas eksperimentas buvo pakartotas sumažinus deformacijos detalumą, ir gauti rezultatai palyginti tarpusavyje.

1.2. Tiriamoji problema

Objektų fizikinių poveikių modeliavimo naudojant spyruoklių metodą tyrimas ir pritaikymas mobiliems įrenginiams.

1.3. Tyrimo objektas

Šiame darbe nagrinėjamas spyruoklių metodas, kurio pagalba atliekamas trimačių objektų transformacija. Šių transformacijų metu paimamas trimatis objektas, ir deformuojama jo forma, priklausomai nuo jį veikiančių išorinių jėgų. Šiame darbe nagrinėjamos šio algoritmo pritaikymo galimybės mobiliuosiuose įrenginiuose.

1.4. Darbo tikslas

Šio projekto tikslas sukurti lenktynių žaidimą, kuriame būtų realizuotas spyruoklių metodas, kuris yra kūnų deformacijos algoritmas, kuris leistų greitai rasti kokybiškiausią sprendimą

mechaniniuose deformacijos uždaviniuose. Taip pat ši sistema turi būti optimizuota mobiliems įrenginiams. Optimizacija apims deformacijos algoritmų bei grafikos apdorojimo sistemos pritaikymą ribotiems skaičiavimo resursams, kurių turi pakakti mobiliems įrenginiams su vidutinės spartos skaičiavimo resursais.

Sukurtas žaidimas buvo analizuojamas. Analizės metu buvo bandoma įvertinti žaidimo veikimą, deformacijų atvaizdavimą, jų spartą ir detalumą.

1.5. Darbo uždaviniai

Šiame darbe buvo suformuluotos šios užduotys:

1. Apžvelgti spyruoklių metodo algoritmą, jo veikimo principus.
2. Iširti geometrinių objektų deformacijos elgesį deformacijos proceso metu.
3. Įvertinti skaičiavimo pajėgumus, kurie reikalingi apskaičiuoti deformacijas naudojant spyruoklių metodą.
4. Įvertinti atliktų deformacijų tikslumą.
5. Pritaikyti spyruoklių metodą, mobiliems įrenginiams.

2. ANALITINĖ DALIS

2.1. Užduotis

Šio projekto tikslas sukurti lenktynių žaidimą, kuriame būtų realizuotas spyruoklių kūnų deformacijos algoritmas, kuris leistų greitai rasti kokybiškiausią sprendimą mechaniniuose deformacijos uždaviniuose. Taip pat ši sistema turi būti optimizuota mobiliems įrenginiams. Optimizacija apims deformacijos algoritmų bei grafikos apdorojimo sistemos pritaikymą ribotiems skaičiavimo resursams, kurių turi pakakti mobiliems įrenginiams su vidutinės spartos skaičiavimo resursais.

2.2. Problemos

Bandant įgyvendinti geometrinių objektų deformacijos sistema tenka susidurti su keletu projektavimo ir įgyvendinimo problemų. Daugelis problemų yra sprendžiami skaičiavimais, tačiau turint minimalius resursus susiduriame su skaičiavimo resursų stoka.

2.3. Žaidimo kūrimas

Žaidimo kūrimas skiriasi, nuo paprasto programinės įrangos kūrimo. Žaidimo kūrimo sudėtingumą didina tai, kad jam vykdyti reikalingi didesnė skaičiavimo sparta ir viskas turi vykti realiu laiku. Taip pat skaičiavimai turi būti sinchronizuoti su grafikos atvaizdavimu. Jei atsiranda uždelsimas kurioje nors dalyje, kitos sistemos dalys taip pat pradeda veikti netinkamai, dėl šių priežasčių nukenčia vartotojo gebėjimas naudotis žaidimu. Žaidimas turi būti tinkamai suprojektuotas, kad išvengtų problemų ir vartotojui būtų patogus žaisti.[14][12]

2.4. 3D atvaizdavimas

3d atvaizdavimas (angl. 3d rendering), tai procesas, kurio metu automatiškai pakeičiamas 3d vielos rėmo modelis į 2d paveikslėlį su 3d foto realistiškais efektais arba ne foto realistišką atvaizdavimą kompiuteryje.[1]

3d atvaizdavimui atlikti yra 2 būdai:

- Realus laiko atvaizdavimas – šis būdas skirtas žaidimams ir interaktyviems vaizdams. Šiuo būdu atvaizduojant, sistema suteikia interaktyvumą, sistemą galima valdyti realiu laiku.
- Ne realus laiko atvaizdavimas – šis būdas naudojamas, kai nereikalingas realaus laiko interaktyvumas. Šis būdas paprastai naudojamas animacijose. Jo metu išgaunamas detalesnis ir labiau foto realistiškas vaizdas. Priešingai nuo realaus laiko vaizdų atvaizdavimo, kur visi kadrai atvaizduojami realiu laiku, ne realaus laiko vaizdų generavime, vieno kadro generavimui gali būti skirtos iki kelių dešimčių valandų.

2.5. Daugiakampių tinklelis

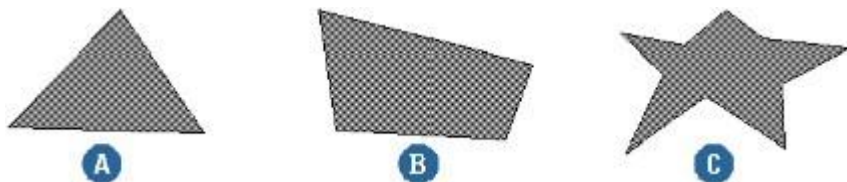
Daugiakampių tinklelis (angl. Polygon mesh) – viršūnių, kraštinių ir paviršių rinkinys, kuris nusako 3d kompiuterio objekto formą. [2]

3D modeliuose naudojami daugiakampiai – dviejų matavimų, uždaros formos, kurių kraštinės sudaro tiesios linijos. Kraštinių susikirtimo taškai laikomi viršūnėmis. Patys paprasčiausi daugiakampiai yra trikampės figūros.

Daugiakampiai klasifikuojami, pagal viršūnių arba kraštinių kiekį. Modeliavime daugiausiai naudojami trikampės ir keturkampės figūros.

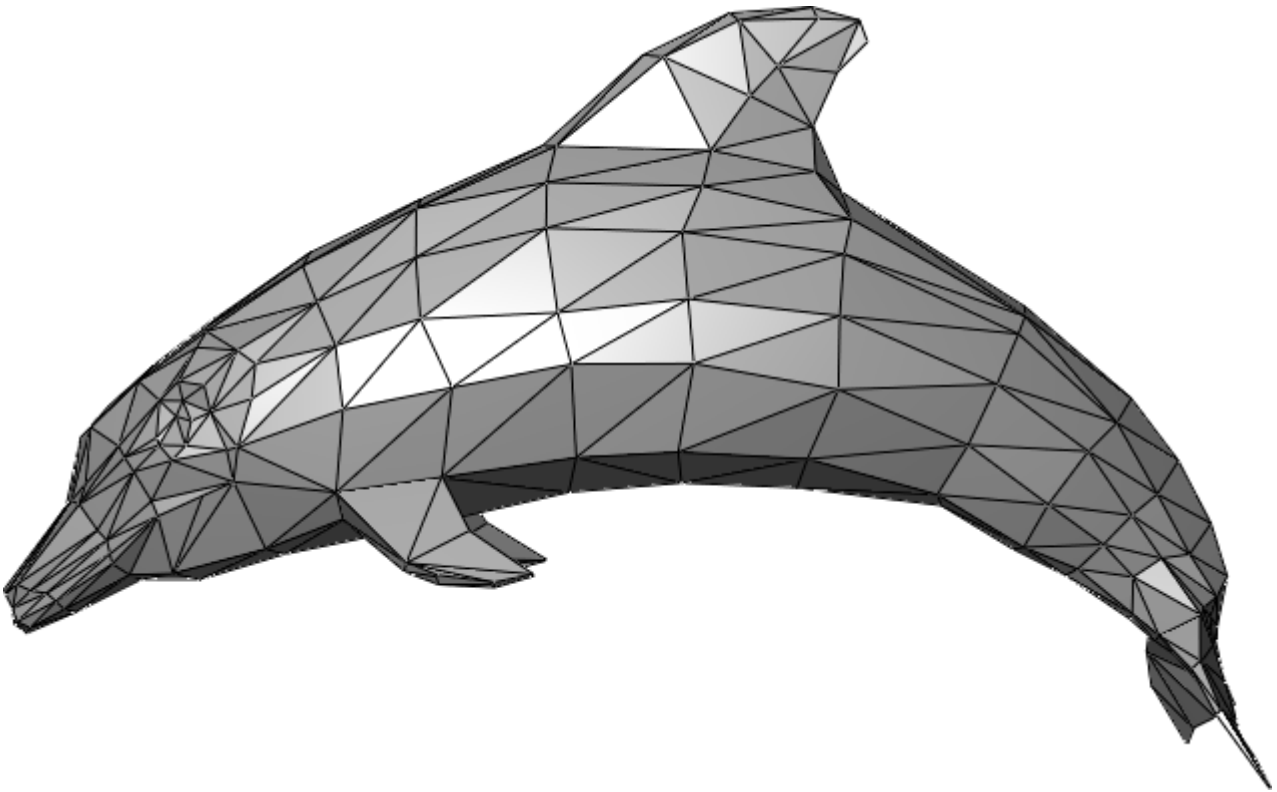
Trikampių figūrų esminė savybė yra tai, kad šios figūros yra visada plokščios, tuo tarpu keturkampės figūras lengviau skaidyti. Trikampių figūrų naudojimas, taip pateikia geresnius rezultatus, 3D modelių atvaizdavime.

Kai kuri žaidimų varikliai reikalauja, kad naudojamos 3D modeliuose naudojami daugiakampiai būtų trikampiai arba keturkampiai. Modeliai kuris susmulkinti į daugiau vienodų daugiakampių, paprastai duoda geresnius rezultatus, modifikuojant daugiakampių tinklelį.



Paveikslėlis 1 Daugiakampių rūšys: A – trikampis, B – keturkampis, C - daugiakampis

Daugiakampių tinklelis, tai 3D objektas, kuris sudarytas vieno ar kelių daugiakampių, kurie turi bendras briaunas, taip sudarydami trimatį objektą.



Paveikslėlis 2 Trimatis objektas sudarytas iš daugiakampių[3]

2.6. Spyruoklių (Spring/Mass) metodas deformacijoms apskaičiuoti

Tai objektų deformacijai nusakyti skirtas algoritmas.

Šiame modelyje yra laikoma, kad objektas yra sudarytas iš atskirų masės dalelių. Masės paprastai susiejamos su objekto viršūnėmis. Jos yra sujungtos spyruoklėmis su šalia esančiomis masėmis. Spyruoklėmis paprastai laikomos briaunos, jungiančios atskiras viršūnes. Spyruoklių modelyje geometrija yra diskretizuota į mases, kurios tarpusavyje yra sujungtos lygtimi, kuri vadinama Lagranžo judėjimo lygtimi (angl. Lagrange equation of motion).

$$M_i \ddot{x} + \delta_i P x + K_i x_i = F_{extern}; (1)$$

Viena iš pagrindinių problemų, su kuria susiduria Spyruoklių metodas yra reikiamų standumo parametrų radimas tam tikriems audiniams ar medžiagoms. Taip pat šiame metode diskretiacija nepilnai atvaizduoja visą objektą. Norint apeiti šį trūkumą, padidinamas mazgų jungiamumas panaudojant atstatomąją jėgą ir pridodant slopinimą jėgos išskaidymui.

Spyruoklių metodas simuliuoja audinių fizikas dideliu dažniu, dėl šios priežasties šis metodas tinkamas realaus laiko simuliacijoms. Šiame modelyje, audinių trikampės tinklelių viršūnės yra laikomos atskiros taškų masės, kurios patalpintos viršūnių vietose ir sujungtos tarpusavyje linijinėmis spyruoklėmis. Panaudojant jėgą kuriai nors masei, judins tam tikrą mazgą ir taip pat perduos veikiančią jėgą šalia esantiems mazgams. Šio modelio formulė yra ši:

$$m_i a_i^t = \sum_{j \in N(i)} k_{ij} d_{ij}^t (l_{ij}^t - l_{ij}^0) - \gamma_i v_i^t + f_{exti}^t; (2)$$

Čia:

a_i^t - žymi dalelių pagreitį laike t ;

$N(i)$ - i kaimynų sąrašas;

k_{ij} - spyruoklės jungiančios i ir j standumas;

d_{ij}^t – vienetinis vektorius tarp i ir j dalelių;

l_{ij}^t - spyruoklės ilgis tarp i ir j , laike t ;

l_{ij}^0 - pradinis spyruoklės ilgis tarp i ir j ;

γ_i - slopinimo koeficientas;

v_i^t - i dalelės greitis laike t ;

f_{exti}^t - išorinė jėga;

$\sum_{j \in N(i)} k_{ij} d_{ij}^t (l_{ij}^t - l_{ij}^0)$ – apibrėžia spyruoklės jėgą pritaikytą mazgui i , tokia pati jėga pritaikyta j mazgui, tačiau priešingos krypties.

Kiekviename atvaizdavimo kadre gaunamas šios diferencialinės lygties rezultatas ir naujos dalelių pozicijos turėtų būti perduotos grafikos variklui. Rezultatas šiai diferencialinei lygčiai gaunamas Verlet integracijos būdu. Verlet integracijoje sekančio žingsnio dalelių pozicijos gaunamos panaudojus prieš tai buvusių dalelių poziciją ir pagreitį. Panaudojus Verlet integraciją formulė pakeičiama į:

$$x(t+h) = x(t) + (x(t) - x(t-h)) * damping + a * h * h; (3)$$

Čia:

$x(t+h)$ - norimos viršūnės pozicija;

$x(t-h)$ - nusako prieš tai buvusią poziciją;

damping - koeficientas, nusakantis energijos išsklaidymą, kurs atsiranda dėl trinties ar kitų jėgų. Jis taip pat naudojamas suteikti stabilumo bei realistiškumo.

Šiame metode pasirinkta Verlet integracija norint padaryti informacijos saugojimą efektyvesnį, kadangi jame išsaugomos tik praeitų dalelių pozicijos.[16][18][19][20]

Spyruoklių metodo pavaizdavimas:



Paveikslėlis 3 Spyruoklių metodas

Algoritmo privalumai ir trūkumai[16]:

Lentelė 1 Spyruoklių algoritmo privalumai ir trūkumai

Privalumai	Trūkumai
Greitai ir lengvai įgyvendinamas	Silpnas saugant tinklelio tūrį
Nereikalingas pradinių sąlygų nustatymas	Stabilumas priklauso nuo laiko žingsnio
Tinklelio topologijos (mesh topology) pokyčiai lengvai sutvarkomi	Sunku integruoti audinio parametrus į sistemą.
Tinkamas lygiagrečiams skaičiavimams	Sudėtinga įtraukti apribojimus į sistemą.
	Sudėtinga patikrinti deformacijas

2.7. Optimizavimas

Sistema yra skirta išmaniesiems telefonams, kurie turi kur kas mažesnius skaičiavimo resursus nei šiuolaikiniai kompiuteriai. Siekiant išlaikyti stabilų sistemos veikimą, būtina sumažinti reikalingų skaičiavimo resursus optimizacijos būdu.[25]

Spartą nusakyti galima išskirti kelis parametrus:

- Aritmetinių funkcijų vykdymas;
- Atminties naudojimas;
- Paleidimo laikas;
- Programos dydis;

Optimizuoti sistemą galima šiais būdais[4]

- Algoritmo optimizavimas – tai alternatyvių algoritmų ieškojimas arba algoritmo tikslumo mažinimas;
- Programos kodo optimizavimas – tai atsisakymas nereikalingų ciklų, nereikalingų funkcijų pašalinimas. Šio būdo pagalba galima sumažinti atminties sunaudojimą, taip pat optimizuojant kodą sumažinama tikimybė, kad bus atminties nutekėjimas.
- Duomenų tipų optimizavimas – šis būdas siūlo naudoti tik reikiamų dydžių duomenų tipus, kadangi skirtingi duomenų tipai naudoja skirtingus resursus. Jei kuriant sistemą reikalingi

duomenų tipai dirbti su mažais duomenimis, o naudojami duomenų tipai, kurie skirti apibrėžti didelių duomenų tipus, nereikalingai eikvojami resursai.

Taip pat kuriant žaidimą, be programinio kodo, reikia optimizuoti ir kitus žaidimo resursus[5]. Mobilijų įrenginių žaidimuose, didelę įtaką spartai turi modelių detalumas. Taip pat įtaką spartai turi modelių kiekis žaidimo scenose. Kuriant žaidimus, modeliai turėtų būti optimizuoti. Pagrindinis modelių optimizavimo būdas yra kurti paprastesnius modelius. Taip pat reikia kurti kiek įmanoma paprastesnes žaidimų scenas. Detalūs modeliai ir detalios žaidimų scenos reikalauja spartaus pagrindinio procesoriaus ir grafinio procesoriaus. Mobiliuosiuose įrenginiuose tiek pagrindinis procesorius tiek grafinis procesorius yra lėtesni, nei kompiuteriuose, todėl kuriant žaidimus mobiliems įrenginiams reikia skirti daugiau dėmesio detalumo optimizavimui. Taip pat žaidimų detalumas gali turėti žaidimo dydžiui, kas gali kelti problemų norint jį parsisiųsti mobiliuoju ryšiu. Žaidimų optimizavimas susideda iš šių dalių:

- Atskirų žaidimų modelio detalumo kontroliavimas – kuriant modelius žaidimui, modelio kūrėjai turėtų kontroliuoti modelio poligonų kiekį;
- Žaidimo scenų optimizavimas – optimizuojant žaidimo scenas, reikia valdyti modelių kiekį, kuris atvaizduojamas scenose. Optimizuojant taip pat turėti būti įvertintas objektų patalpinimo tankumas.
- Tekstūrų naudojimas – naudojamos tekstūros gali naudoti apie 33% visos žaidimo naudojamos apimties. Naudojant tekstūras rekomenduojama naudoti Mip Map technologiją. Šis technologija gali sumažinti tekstūrų sunaudojamą atminties kiekį. Jo metu nereikia apdoroti tekstūrų realiu laiku, jos iš anksto paruošiamos. Naudojant Mip Map technologiją išlanksto paruošiami keli tekstūros variantai, skirti naudoti skirtingoms sąlygomis, vietoje to, kad būtų naudojamo įrenginio resursai, kad tekstūros būtų apdorotos realiu laiku.
- LOD panaudojimas – tai panaudojimas technologijos, kuri leidžia keisti objektų detalumą realiu laiku. Jį naudojant galima sumažinti objektų detalumą nepajaučiant vizualaus skirtumo, pvz. atvaizduojant objektą iš didelio atstumo. Naudojant šia technologiją, kaip ir tekstūrų atveju naudojant Mip Map, reikia pasiruošti kelis objektų variantus, kurie anuojami pagal skirtingus poreikius.

2.8. Bibliotekos ir programų kūrimo sąsajos

Programų kūrimo sąsajos (angl. Application Programming Interface (API)) – tai gairių, protokolų rinkinys, kuris gali būti panaudotas programų kūrimui. Jie naudojami pasiekti ir panaudoti paslaugas ar resursus, kuriuos suteikia biblioteka ar kita programa, kurioje realizuotos API pateikiamos funkcijos. API tarnauja, kaip sąsaja, tarp skirtingų sistemos komponentų.

API galim išskirti į kelias pagrindines kategorijas:

- Operacinės sistemos API;
- Programavimo kalbų API;
- Programų servisų API;
- Infrastruktūros paslaugų API;
- WEB paslaugų API;

OpenGL ES – tai integruotoms sistemoms (embedded systems) skirta, žemo lygio biblioteka, kuri naudojama supaprastinti su grafika susijusį programavimą. Ją prižiūri Khronos Group.Inc. Ši biblioteka apima 3d ir 2d grafikos apdorojimą. OpenGL ES yra atviro kodo biblioteka. OpenGL ES biblioteka yra plačiausiai naudojama biblioteka tarp mobiliųjų įrenginių. Vienintelė išimtis yra Windows Phone sistema, kuri naudoja savo DirectX sukurtą biblioteką, skirtą grafikos apdorojimui.

OpenGL ES yra pagrįsta OpenGL, todėl ją naudojantis nereikia papildomų technologijų. Ši priežastis užtikrina, kad perėjimas tarp OpenGL ES ir OpenGL skirto kompiuteriams, yra kuo papastesnis. [21][22][23][24][11]

OpenSL (Open Sound Library for Embedded Systems) – tai žemo lygio, techniškai spartinama biblioteka, skirta naudotis garsų posisteme. Ji skirta integruotoms sistemoms.[6]

Simple DirectMedia Layer (SDL) biblioteka – įrankių rinkinys, palengvinantis žemo lygio įrenginių valdymą. Ši biblioteka suteikia ne tik įvesčių funkcijas, bet ir garso bei video išvedimą. Ji taip pat gali būti panaudota nuskaitant įvairiu duomenų failus. SDL biblioteka, taip pat gali būti panaudota kuriant žaidimų variklius, kadangi ji suteikia bazines funkcija, naudojamas žaidimų kūrime.[7]

Allegro biblioteka – tai atviro kodo žaidimų programavimo biblioteka, palaikanti įvairias platformas. Ji labiausiai skirta C ir C++ programavimo kalboms. Allegro, kaip ir SDL, skirta perimti žemo lygio užduotims. Tai įtraukia langų sukūrimą, vartotojų įvesčių perėjimą, duomenų įkrovimą, paveikslėlių atvaizdavimą ir muzikos posistemės valdymą. Allegro suteikia galimybę atvaizduoti tik 2d grafiką, bet ne 3d, tačiau ši biblioteka gali būti panaudota su kitomis bibliotekomis.[8]

Simple and Fast Multimedia Library (SFML) – tai atviro kodo biblioteka, kuri skirta kurti žaidimams ir multimedijos programoms. Ši biblioteka oficialiai palaiko C ir .NET šeimos programavimo kalbas, tačiau ją galima naudoti ir su kitomis programavimo kalbomis. Šią biblioteką sudaro keli moduliai, kurie suteikia prieigą prie sistemos, vaizdo ir garso posistemės, langų atvaizdavimo ir tinklo posistemės.[9]

2.9. Žaidimo varikliai

Žaidimų varikliai – tai programinė įranga, kurios paskirtis perimti dalį žaidimų kūrimo etapų, kad žaidimo kūrėjai galėtų susitelkti į žaidimo logikos, istorijos ir išvaizdos kūrimą. Paprastai žaidimo variklis perima šias žaidimo kūrimo dalis:

- Žaidimo atvaizdavimas;
- Fizikos atvaizdavimas;
- Įvesčių apdorojimas.

Žaidimo varikliai pateikia įvairius komponentus, kurie panaudojami žaidimų kūrime. Tai apima duomenų užkrovimą, jų atvaizdavimą, animacijų pateikimą, grafikos atvaizdavimą, fiziką.

Žaidimų varikliai savo funkcija pateikia su programų kūrimo sąsajomis ([angl.](#) Application Programming Interface, API) arba pasinaudojant programinės įrangos kūrimo komplektais ([angl.](#) Software Development Kit, SDK). Paprastai API būna pateiktos, kartu su bibliotekomis. SDK paprastai pateikia bibliotekų ar API rinkinius. SDK taip pat pateikia įrankius, kurių pagalba galima kurti žaidimus.

Yra kelios žaidimų variklių rūšys, kurios gali apimti, tik dalį žaidimo kūrimo dalių, o kitas dalis gali tekti sukurti patiems kūrėjams arba naudoti kitas bibliotekas ar karkasus.

Šių variklių pavyzdys būtų grafikos varikliai. Šie varikliai apima tik grafikos atvaizdavimą, o kitas dalis turėtų realizuoti patys žaidimo kūrėjai arba pasinaudoti papildomais varikliais ar žaidimų bibliotekomis. Grafikos variklio pavyzdys yra OGRE (Object-Oriented Graphics Rendering Engine). Tai išskirtinai žaidimų grafikos variklis, kuris perima žaidimo grafikos atvaizdavimą. Naudojantis šiuo varikliu būtų pateiktos visos reikalingos bibliotekos, susijusios su grafikos išvedimu į įrenginio ekraną, tačiau kitas žaidimo dalis, tokias kaip fizika ar įvesčių nuskaitymas, reiktų realizuoti patiems kūrėjams ar naudoti kitas bibliotekas.

Fizikos variklis - variklis kuris perima žaidimų fizikos realizavimą. Šie varikliai gali būti panaudoti su kitais žaidimo varikliais, suteikiant jiems fizikos savybių. Tai gali būti objektų susidūrimų aptikimas, gravitacijos simuliacija, skysčių fizika ar objektų deformacija. Fizikos varikliai taip pat naudojami mokslinėse srityse. Jie gali būti panaudoti balistikos simuliacijai ar skysčių simuliacijai. Fizikos variklių pavyzdžiai yra šie: Bullet, Havok, PhysX.

3. PROJEKTINĖ DALIS

Šioje dalyje pateikiama magistratūros studijų metu sukurtos programinės įrangos techninės-projektinės dokumentacijos esminiai aspektai.

Projektinė dalis buvo grupinis darbas, kuris buvo atliktas dviejų studentų: Mariaus Kimbirausko iš IFM-3/2 grupės ir Marijaus Sebeckio iš IFM-3/2 grupės.

Kuriant žaidimą, buvo pasirinktas Unity žaidimo variklis. Šis variklis buvo pasirinktas, kadangi juo lengva kurti žaidimus, kurie turi veikti keliose skirtingose platformose.

3.1. Sistemos panaudos atvejų diagrama

Čia pateikti sistemos panaudos atvejai, kurie nusako pagrindines sistemos funkcijas. Žemiau pateikta panaudos atvejų diagrama. Taip pat pateiktas kiekvieno panaudos atvejo aprašymas.



Paveikslėlis 4 Panaudos atvejų diagrama

Panaudojimo atvejis 1: Žaidėjo prisijungimas

Aktorius: Žaidėjas.

Tikslas: perduoti programai vartotojo prisijungimo duomenis.

Ryšiai su kitais PA: iššaukiamas suaktyvavus kitus PA norint nustatyti ar vartotojas turi teisę į pasirinktą PA.

Prieš-sąlygos: duomenų bazė veikia ir yra pasiekama programos.

Sužadinimo sąlyga: sistemos vartotojas paleidžia PĮ vykdomąją rinkmeną.

Po-sąlyga: vartotojas autorizuotas ir įgavo teises naudotis programos paslaugomis.

Pagrindinis scenarijus:

1. Sistema paprašo įvesti vartotojo prisijungimo duomenis.
2. Patikrinama, ar toks vartotojas egzistuoja duomenų bazėje ir suteikiamos atitinkamos teisės.

Alternatyvūs scenarijai:

- Vartotojas įvedė neteisingus prisijungimo duomenis.

- Vartotojas nusprendė baigti darbą dar neprisijungus prie programos.

Panaudojimo atvejis 2: Rezultatų matymas

Aktorius: Žaidėjas.

Tikslas: Atvaizduoti vartotojui daugiausiai surinkusių taškų vartotojus.

Ryšiai su kitais PA: Vartotojas turi būti sėkmingai identifikuotas žaidėjo prisijungimo PA

Prieš-sąlygos: duomenų bazė veikia ir yra pasiekama programos.

Sužadinimo sąlyga: sistemos vartotojas meniu punkte pasirenka punktą, nusakantys rodyti daugiausiai surinkusių taškų žaidėjus.

Po-sąlyga: vartotojas mato daugiausiai surinkusių taškų vartotojų sąrašą su atitinkama informacija.

Pagrindinis scenarijus:

1. Identifikuotas vartotojas meniu sąrašė paspaudžia rodyti rezultatus
2. Vartotojui pateikiamas daugiausiai surinkusių taškų vartotojų sąrašas ir jų surinktais taškais.

Alternatyvūs scenarijai:

- Vartotojui atvaizduojamas tuščias rezultatų sąrašas
- Vartotojui atvaizduojama programos apribotas kiekis rezultatų kiekis

Panaudojimo atvejis 3: Meniu atvaizdavimas

Aktorius: Žaidėjas.

Tikslas: Identifikuotas žaidėjas inicializuoja kitus PA pasirinkdamas atitinkamą meniu punktą.

Ryšiai su kitais PA: Vartotojas turi būti sėkmingai identifikuotas žaidėjo prisijungimo PA. Vartotojas gali inicializuoti „Rezultatų matymo“, „Naujo žaidimo pasirinkimas“, „Išsaugoto žaidimo pasirinkimo“ PA.

Prieš-sąlygos: Vartotojas sėkmingai identifikuotas „Žaidėjo prisijungimo“ PA.

Sužadinimo sąlyga: Žaidėjo prisijungimo PA sėkmingo prisijungimo metu sužadina Meniu atvaizdavimo PA.

Po-sąlyga: Vartotojas mato meniu, kuris inicializuoja atitinkamus Panaudos Atvejus.

Pagrindinis scenarijus:

1. Vartotojas sėkmingai identifikuoja save programoje.
2. Žaidėjo prisijungimo PA inicializuoja Meniu atvaizdavimo PA

Alternatyvūs scenarijai:

- Sukuriamas naujas žaidėjas programoje
- Naujas žaidėjas identifikuojamas programoje

Panaudojimo atvejis 4: Naujo žaidimo pasirinkimas

Aktorius: Žaidėjas.

Tikslas: Inicializuoti duomenis, numatytus pagal nutylėjimą, reikalingus kitiems PA.

Ryšiai su kitais PA: Naujo žaidimo pasirinkimo PA praplečia „Meniu atvaizdavimas“ PA. „Naujo žaidimo pasirinkimas“ PA inicializuoja „Žaidimo logikos vykdymo“ PA.

Prieš-sąlygos: Žaidėjas „Meniu atvaizdavimas“ PA gali pasirinkti meniu punktą, inicializuojantis šį PA.

Sužadinimo sąlyga: . Žaidėjas „Meniu atvaizdavimas“ PA pasirenka meniu punktą inicializuojantis „Naujo žaidimo pasirinkimas“ PA.

Po-sąlyga: Šis PA perduoda pradinis duomenis reikalingus inicializuoti „Žaidimo logikos vykdymas“ PA.

Pagrindinis scenarijus:

1. Žaidėjas meniu elementų sąrašė pasirenka punktą nurodantis naujo žaidimo pasirinkimą

Alternatyvūs scenarijai:

1. Žaidėjas nurodo specifinius parametrus reikalingus ir tik tuomet inicializuoja naują žaidimą.

Panaudojimo atvejis 5: Išsaugoto žaidimo pasirinkimas

Aktorius: Žaidėjas.

Tikslas: Inicializuoti duomenis, išsaugoti žaidimo metu, reikalingus kitiems PA.

Ryšiai su kitais PA: Išsaugoto žaidimo pasirinkimo PA praplečia „Meniu atvaizdavimas“ PA. „Išsaugoto žaidimo pasirinkimas“ PA inicializuoja „Žaidimo logikos vykdymo“ PA.

Prieš-sąlygos: Žaidėjas „Meniu atvaizdavimas“ PA gali pasirinkti meniu punktą, inicializuojantis šį PA.

Sužadinimo sąlyga: . Žaidėjas „Meniu atvaizdavimas“ PA pasirenka meniu punktą inicializuojantis „Išsaugoto žaidimo pasirinkimas“ PA.

Po-sąlyga: Šis PA perduoda pradinis duomenis reikalingus inicializuoti „Žaidimo logikos vykdymas“ PA.

Pagrindinis scenarijus:

1. Žaidėjas meniu elementų sąrašė pasirenka punktą nurodantis Išsaugoto žaidimo pasirinkimą.
2. Vartotojas pasirenka išsaugoto žaidimo objektą.

Alternatyvūs scenarijai:

2. Žaidėjas.

Panaudojimo atvejis 6: Žaidimo logikos vykdymas

Aktorius: Mobilusis įrenginys.

Tikslas: Kontroluoti viso žaidimo logiką bei sąryšį su kitais PA.

Ryšiai su kitais PA: „Žaidimo valdymas“ PA gali stabdyti šį PA o vartotojas nurodyti kokius veiksmus reikia atlikti programai. „Automobilio valdymas“ PA atsakingas už žaidėjo įvesties duomenų pateikimą šiam PA. „Išsaugoto žaidimo pasirinkimas“ arba „Naujo žaidimo pasirinkimas“ PA perduoda duomenis reikalingus startuoti šiam PA. „Naujo žaidimo pradėjimas“ arba „Išsaugoto žaidimo užkrovimas“ PA perduoda duomenis reikalingus objektų atvaizdavimui. „Žaidimo atvaizdavimas“ PA atsakingas už objektų atvaizdavimą vykdant Žaidimo logikos vykdymo PA. Deformacijos apskaičiavimo PA atsakingas už Žaidimo logikos vykdyme įvykusių objektų deformacijas. Garso posistemės PA atsakinga už garsų skleidimą iš programos.

Prieš-sąlygos: Inicializuoti duomenys reikalingi žaidimo logikos vykdymui.

Sužadinimo sąlyga: Iškviečiamas metodas nurodantis pradėti žaidimo logikos vykdymo ciklą.

Po-sąlyga: Sėkmingai komunikacija su kitais įtakojančiais PA

Pagrindinis scenarijus:

1. Iškviečiamas metodas nurodantis žaidimo logikos ciklo inicializavimą
2. Surenkami duomenys iš kitų susietų PA

Alternatyvūs scenarijai:

3. Nenumatytu duomenų paketo rinkiniui inicializavimo atveju žaidimo logikos vykdymas yra stabdomas.

Panaudojimo atvejis 7: Naujo žaidimo pradėjimas

Aktorius: Mobilusis įrenginys.

Tikslas: Inicializuoti duomenis, numatytus pagal nutylėjimą, reikalingus žaidimo logikos vykdymui bei žaidimo objektų užkrovimui.

Ryšiai su kitais PA: „Žaidimo logikos vykdymo“ PA suteikti pradinis duomenis reikalingus sėkmingam žaidimo logikos vykdymui. „Žaidimo objektų užkrovimas“ atsakingas už objektų duomenų saugojimą ir pasiekiamumą.

Prieš-sąlygos: Duomenų bazėje patiekta objektų pradiniai duomenys ir kita reikalinga informacija.

Sužadinimo sąlyga: Iškviečiamas metodas nurodantis grąžinti objektų duomenų paketus.

Po-sąlyga: Grąžinami ,numatytieji pagal nutylėjimą, žaidimo objektų duomenys.

Pagrindinis scenarijus:

1. Iškviečiamas metodas objektų duomenų grąžinimui
2. Nuskaitomi numatytieji objektų duomenys iš duomenų bazės
3. Grąžinami objektų duomenys

Alternatyvūs scenarijai:

1. Iškviečiamas metodas objektų duomenų grąžinimui
2. Duomenų bazėje nerandama duomenų aprašančių objektą

3. Gražinama klaidos rezultatas

Panaudojimo atvejis 8: Išsaugoto žaidimo užkrovimas

Aktorius: Mobilusis įrenginys.

Tikslas: Inicializuoti duomenis, išsaugotus žaidimo saugojimo metu, reikalingus žaidimo logikos vykdymui bei žaidimo objektų užkrovimui.

Ryšiai su kitais PA: „Žaidimo logikos vykdymo“ PA suteikti pradinius duomenis reikalingus sėkmingam žaidimo logikos vykdymui. „Žaidimo objektų užkrovimas“ atsakingas už objektų duomenų saugojimą ir pasiekiamumą.

Prieš-sąlygos: Duomenų bazėje patiekta išsaugotų objektų duomenys ir kita reikalinga informacija.

Sužadinimo sąlyga: Iškviečiamas metodas nurodantis grąžinti objektų duomenų paketus.

Po-sąlyga: Gražinami žaidimo objektų duomenys.

Pagrindinis scenarijus:

1. Iškviečiamas metodas objektų duomenų gražinimui
2. Nuskaitomi numatytieji objektų duomenys iš duomenų bazės
3. Gražinami objektų duomenys

Alternatyvūs scenarijai:

1. Iškviečiamas metodas objektų duomenų gražinimui
2. Duomenų bazėje nerandama duomenų aprašančių objektą
3. Gražinama klaidos rezultatas

Panaudojimo atvejis 9: Žaidimo objektų užkrovimas

Aktorius: Mobilusis įrenginys.

Tikslas: Saugoti žaidimo objektų duomenis.

Ryšiai su kitais PA: „Naujo žaidimo pradėjimas“ ir „Išsaugoto žaidimo užkrovimas“ PA nurodo specifinius parametrus reikalingus duomenų mainams su kitais PA.

Prieš-sąlygos: Nurodyti parametrai nusakantys su kokiais objektais reikia duomenų mainų.

Sužadinimo sąlyga: Iškviečiamas metodas nurodantis grąžinti objektų duomenų paketus.

Po-sąlyga: Gražinami žaidimo objektų duomenys.

Pagrindinis scenarijus:

1. Iškviečiamas metodas objektų duomenų gražinimui
2. Nuskaitomi numatytieji objektų duomenys iš duomenų bazės
3. Gražinami objektų duomenys

Alternatyvūs scenarijai:

4. Iškviečiamas metodas objektų duomenų gražinimui

5. Duomenų bazėje nerandama duomenų aprašančių objektą
6. Grąžinama klaidos rezultatas

Panaudojimo atvejis 10: Žaidimo atvaizdavimas

Aktorius: Mobilusis įrenginys.

Tikslas: Žaidimo objektų atvaizdavimą 3D grafikoje mobiliajame įrenginyje.

Ryšiai su kitais PA: „Žaidimo logikos vykdymas“ PA nurodo kuriuos objektus atvaizduoti mobiliajame įrenginyje.

Prieš-sąlygos: Priskiriamos objektų reikšmės reikalingos duomenų atvaizdavimui.

Sužadinimo sąlyga: Iškviečiamas metodas inicializuojantis objektų atvaizdavimą.

Po-sąlyga: Žaidimo objektai atvaizduojami 3D grafikoje mobiliajame įrenginyje.

Pagrindinis scenarijus:

1. Priskiriamos žaidimo objektų duomenys.
2. Atvaizduojami objektai 3D grafikoje.
3. Tęsimas ciklas vėl nuo pirmo punkto.

Alternatyvūs scenarijai:

1. Klaidingai inicializuoti žaidimo objektų duomenys.
2. Neatvaizduojamas objektas 3D grafikoje.
3. Tęsimas ciklas iškviečiantis kitų objektų užkrovimą.

Panaudojimo atvejis 11: Deformacijos apskaičiavimas

Aktorius: Mobilusis įrenginys.

Tikslas: Apskaičiuoti žaidimo objektų geometrinės korekcijas.

Ryšiai su kitais PA: „Žaidimo logikos vykdymas“ PA nurodo kuriuos ir kaip objektus reikia deformuoti.

Prieš-sąlygos: Priskiriamos objekto reikšmės reikalingos geometrinėms korekcijoms apskaičiuoti.

Sužadinimo sąlyga: Iškviečiamas metodas inicializuojantis objektų deformacijos skaičiavimą.

Po-sąlyga: Išsaugomi objektai su geometrinėmis deformacijomis.

Pagrindinis scenarijus:

1. Priskiriamos žaidimo objekto duomenys.
2. Vykdomi geometrinio objekto deformacijos.
3. Išsaugomi geometrinio objekto deformacijos.

Alternatyvūs scenarijai:

1. Priskiriamos žaidimo objekto duomenys.
2. Geometrinio objekto deformacijos apskaičiavimo metu įvyksta klaida.

3. Gražinama deformacijos apskaičiavimo klaida.

Panaudojimo atvejis 12: Garso posistemė

Aktorius: Mobilusis įrenginys.

Tikslas: Skleisti žaidimo garsus.

Ryšiai su kitais PA: „Žaidimo logikos vykdymas“ PA nurodo parametrus apie žaidimo objektų statusus.

Prieš-sąlygos: Įtraukti visi garso takeliai reikalingi žaidimo eigoje.

Sužadinimo sąlyga: Iškviečiamas metodas inicializuojantis garsų posistemę.

Po-sąlyga: Mobilusis įrenginys garsiakalbių pagalba, skleidžia garsus įtakotus žaidimo logikos.

Pagrindinis scenarijus:

1. Inicializuojama „Garso posistemės“ PA
2. Žaidimo logikos vykdymo PA nurodo parametrus įtakojančius Garso posistemės PA

Alternatyvūs scenarijai:

1. Inicializuojama „Garso posistemės“ PA
2. Žaidimo logika nurodo nenumatytus parametrų rinkinius įtakojančius Garso posistemės PA
3. Garso posistemė laukia naujų Žaidimo logikos vykdymo PA parametrų

Panaudojimo atvejis 13: Žaidimo valdymas

Aktorius: Žaidėjas.

Tikslas: Nurodyti mobiliajam įrenginiui vartotojo pasirinktis.

Ryšiai su kitais PA: „Žaidimo logikos vykdymas“ priima vartotojo nurodymus.

Prieš-sąlygos: Programa reaguoja į vartotojo paspaudimus.

Sužadinimo sąlyga: Žaidėjas paspaudžia ant numatytų komponentų.

Po-sąlyga: Komponento paspaudimas inicializuoja priskirto metodo iškvietimą.

Pagrindinis scenarijus:

1. Žaidėjas paspaudžia ant komponento
2. Komponentas iškviečia jam priskirtą metodą

Alternatyvūs scenarijai:

1. Žaidėjas paspaudžia ant komponento
2. Komponentui nėra priskirtas metodas
3. Komponento paspaudimas įvykdomas be metodo iškvietimo

Panaudojimo atvejis 14: Automobilio valdymas

Aktorius: Mobilusis įrenginys.

Tikslas: Nuskaityti gireskopo ar akselerometro duomenis.

Ryšiai su kitais PA: „Žaidimo logikos vykdymas“ priima judėjimo kryptį iš šio PA.

Prieš-sąlygos: Aptinkama ir nuskaitoma mobiliojo įrenginio akselerometras ir gireskopas.

Sužadinimo sąlyga: Programiškai inicializuojamas kartu su žaidimo logikos vykdymo ciklo pradžia.

Po-sąlyga: Automobilio valdymo PA perduoda įrenginio padėties parametrus žaidimo logikos vykdymo ciklui.

Pagrindinis scenarijus:

1. Inicializuojamas Automobilio valdymo PA kartu su Žaidimo logikos vykdymo PA
2. Gaunami mobiliojo įrenginio padėties parametrai iš akselerometro ir gireskopo
3. Perduodami komponentų duomenys žaidimo logikos vykdymo PA.

Alternatyvūs scenarijai:

1. Inicializuojamas Automobilio valdymo PA kartu su Žaidimo logikos vykdymo PA
2. Nesėkmingas inicializavimas grąžina klaidos kodą.

Žaidimo logikos vykdymas stabdomas.

3.1. Funkciniai reikalavimai

Žemiau pateikti pagrindiniai funkciniai reikalavimai, kurie skirti apibrėžti sistemos funkcionalumą.

Lentelė 2 Naujas žaidimas

Reikalavimas #:	1	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	1
Aprašymas:	Žaidimas turi leisti žaisti naują žaidimą				
Pagrindimas:	Baigus žaidimą, turi būti galimybė pradėti žaidimą pakartotinai				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Menu pasirinktyje yra galimybė pradėti naują žaidimą, kuris inicializuoja žaidimą su pradiniais numatytais duomenimis				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	5		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

Lentelė 3 Žaidimo logikos sustabdymas

Reikalavimas #:	2	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	2
Aprašymas:	Žaidimas gali būti sustabdomas žaidimo eigoje.				
Pagrindimas:	Žaidimo eigoje žaidimas gali būti pertraukiamas, o po pertraukties toliau vykdyti žaidimo logiką.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Po žaidimo pertraukties, žaidimą atkuriamas ir vykdomas tolimesnė žaidimo eiga sustabdyta iki žaidimo pertraukties.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	3		
Priklausomybės	Nėra.	Konfliktai:	Nėra		

Papildoma medžiaga:	
Istorija:	Užregistruotas 2014 balandžio 1d.

Lentelė 4 Rezultatų atvaizdavimas

Reikalavimas #:	3	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	3
Aprašymas:	Žaidimas turi atvaizduoti geriausius rezultatus.				
Pagrindimas:	Vartotojai nori įvertinti savo surinktus žaidimo taškus su kitais vartotojais.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Atvaizduojamas didėjimo tvarka dešimt daugiausiai taškų surinkusių žaidėjų.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	4		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

Lentelė 5 3D grafika

Reikalavimas #:	4	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	4
Aprašymas:	Žaidimas grafika turi būti atvaizduojama 3D grafika.				
Pagrindimas:	Žaidimo atvaizdavimo esmė yra 3D grafika, atvaizduojanti žaidimo geometrinių objektų deformacijas ir geometrinių objektų sąveiką tarpusavyje.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Naudojama 3D grafika geometrinių objektų atvaizdavimui.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	5		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

Lentelė 6 Deformacija

Reikalavimas #:	5	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	5
Aprašymas:	Žaidimas turi panaudoti objektų deformaciją.				
Pagrindimas:	Žaidimo geometriniai objektai sąveikaudami turi keisti savo geometrines formas, kuo panašiau realistiniams fizinių kūnų deformacijoms.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Geometriniai objektai keičia savo geometrines formas tam tikrose objekto srityse, po tam tikrų sąveikų su kitais geometriniais objektais.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	4		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

Lentelė 7 Nauji vartotojai

Reikalavimas #:	6	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	6
Aprašymas:	Žaidimas turi leisti kurti naujus vartotojus				
Pagrindimas:	Daugi vartotojų programoje reikalingas vartotojų identifikavimas ir individualių vartotojų informacijai saugoti.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Menu pasirinkime yra naujo vartotojo kūrimo galimybė.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	4		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

Lentelė 8 Vartotojo duomenų korekcija

Reikalavimas #:	7	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	7
Aprašymas:	Žaidimas turi leisti koreguoti vartotojo informaciją				
Pagrindimas:	Vartotojas turi galimybę keisti savo paskyros informaciją įtakojančia žaidimo eigą.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Vartotojas turi galimybę pakeisti reikšmes, kurios buvo priskirtos pagal nutylėjimą.				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	5		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

Lentelė 9 Vartotojų šalinimas

Reikalavimas #:	8	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	8
Aprašymas:	Žaidimas turi leisti pašalinti vartotojus.				
Pagrindimas:	Vartotojas turi galimybę pašalinti savo paskyrą.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Menu pasirinkimas leidžiantis pašalinti prisijungusio vartotojo paskyrą pakartotinai identifikavus vartotoją.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	4		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

Lentelė 10 Vartotojo informacijos atvaizdavimas

Reikalavimas #:	9	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	9
Aprašymas:	Žaidimas turi atvaizduoti vartotojo informaciją				

Pagrindimas:	Vartotojo informacijai koreguoti reikalinga grafinė sąsaja, palengvinanti duomenų korekciją.		
Šaltinis:	Vadovas		
Tikimo kriterijus:	Vartotojo duomenų informacija, įtakojanti žaidimo eigą ar logiką, patogumo kriterijais atvaizduojama pasitelkiant grafine sąsaja.		
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	5
Priklausomybės	Nėra.	Konfliktai:	Nėra
Papildoma medžiaga:			
Istorija:	Užregistruotas 2014 balandžio 1d.		

Lentelė 11 Žaidimo išsaugojimas

Reikalavimas #:	10	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	10
Aprašymas:	Žaidimas turi leisti išsaugoti žaidimą				
Pagrindimas:	Neišgyvendinus pilnai numatyto žaidimo logikos pabaigos, vartotojas gali išsaugoti žaidimą tarpinėje žaidimo logikos padėtyje, neįgyvendinus žaidimo logikos pabaigos reikalavimų.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Žaidimo eigoje sustabdžius žaidimo logiką, meniu pasirinktyje yra punktas leidžiantis išsaugoti žaidimo padėtį.				
Užsakovo tenkinimas:	1	Užsakovo netenkinimas:	4		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

Lentelė 12 Žaidimo užkrovimas

Reikalavimas #:	11	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	11
Aprašymas:	Žaidimas turi leisti įkrauti išsaugota žaidimą				
Pagrindimas:	Vartotojas turi galimybę atkurti žaidimo logiką iš išsaugotos žaidimo logikos tarpinės padėties.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Meniu punktas leidžiantis įkrauti žaidimo tarpinius duomenis iš išsaugotų tarpinių duomenų paketų.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	2		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

Lentelė 13 Žaidimo objektas

Reikalavimas #:	12	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	12
Aprašymas:	Žaidimas turi leisti pasirinkti sąveikausiančios geometrinį objektą(automobilį)				
Pagrindimas:	Žaidimo įvairovei suteikti reikalinga geometrinių objektų įvairovė, kurios tarpusavyje skiriasi geometrinėmis formomis ir deformacijos				

	savybėmis.		
Šaltinis:	Vadovas		
Tikimo kriterijus:	Meniu pasirinktyje galima pasirinkti žaidimo geometrinius objektą su specifiniais geometrinėmis formomis ir savybėmis.		
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	4
Priklausomybės	Nėra.	Konfliktai:	Nėra
Papildoma medžiaga:			
Istorija:	Užregistruotas 2014 balandžio 1d.		

Lentelė 14 Deformacijos pradžia

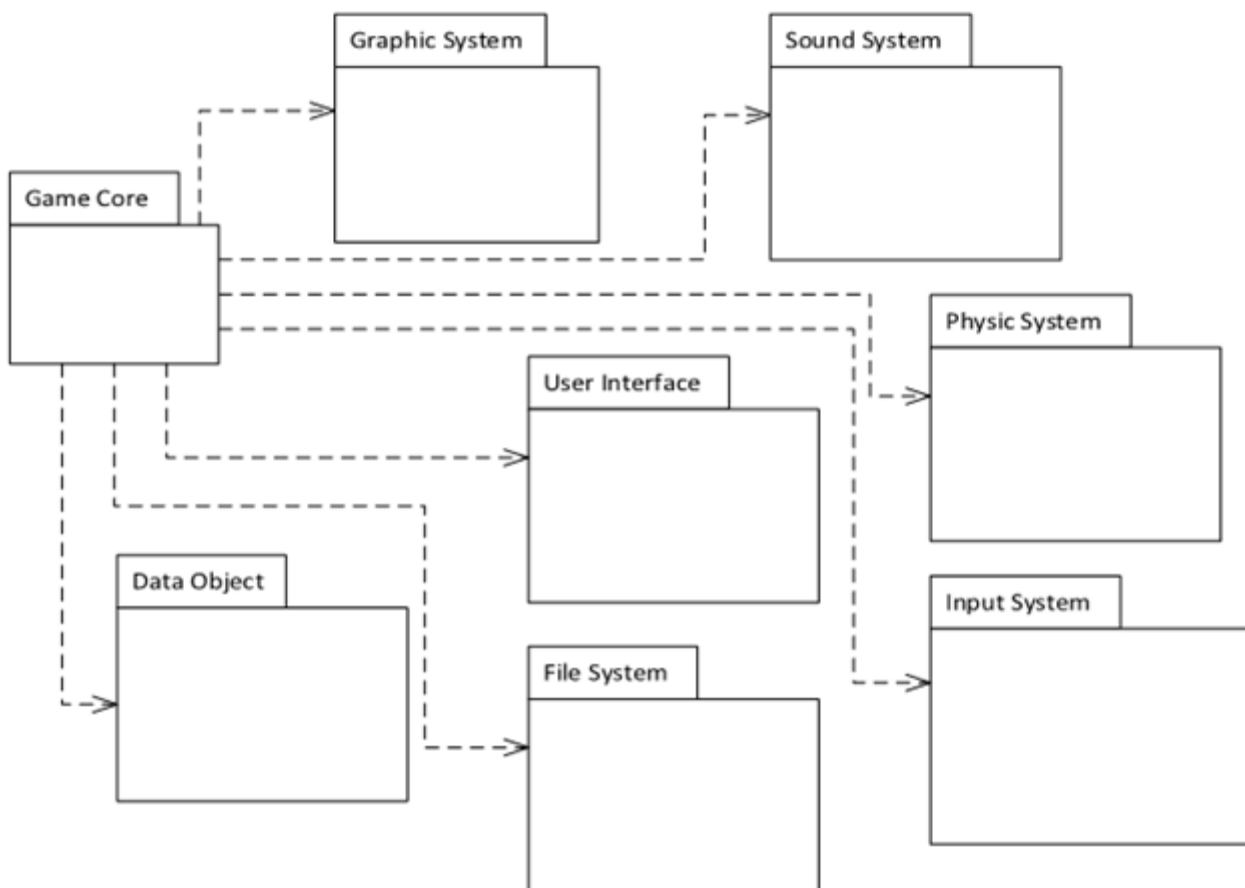
Reikalavimas #:	13	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	13
Aprašymas:	Žaidimo eigoje geometriniai objektai po susidūrimo turi būti deformuoti				
Pagrindimas:	Žaidimo realistiškumui įgyventi reikalinga geometrinių objektų deformacija. Objektų deformacija parodys, kaip kūnai sąveikauja tarpusavyje ir kaip įtakoja deformacija galutiniam geometrinių objektų atvaizdavimui.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Po geometrinių objektų sąveikos tarpusavyje, įvyksta atitinkamoje srityje geometrinių objektų deformacija įtakojanti objekto vizualizacijai.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	3		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

Lentelė 15 Žaidimo fizika

Reikalavimas #:	14	Reikalavimo tipas:	9a	Įvykis/panaudojimo atvejis #:	14
Aprašymas:	Žaidimo eigoje geometriniai objektus sąlygoja fiziniai veiksniai				
Pagrindimas:	Dėl sąveikos tarp geometrinių objektų, deformacijos įvyksta tik atitinkamose geometrinio objekto srityse.				
Šaltinis:	Vadovas				
Tikimo kriterijus:	Po geometrinių sąveikos tarpusavyje, deformacijos įvyksta tik tam tikrose objekto srityje.				
Užsakovo tenkinimas:	2	Užsakovo netenkinimas:	4		
Priklausomybės	Nėra.	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	Užregistruotas 2014 balandžio 1d.				

3.2. Sistemos statinis vaizdas

Žemiau pateiktas sistemos statinis vaizdas. Jame pavaizduota, kaip sistema sugrupuota į dalis. Kiekviena sistemos dalis sugrupuota pagal savo veikimo sritį. Sistema sugrupuota pagal sritis, kad vėliau esant reikalui būtų galima ją tobulinti, įtraukiant naujas funkcijas į jau esamas grupes arba sudarant naują funkcijų grupę.

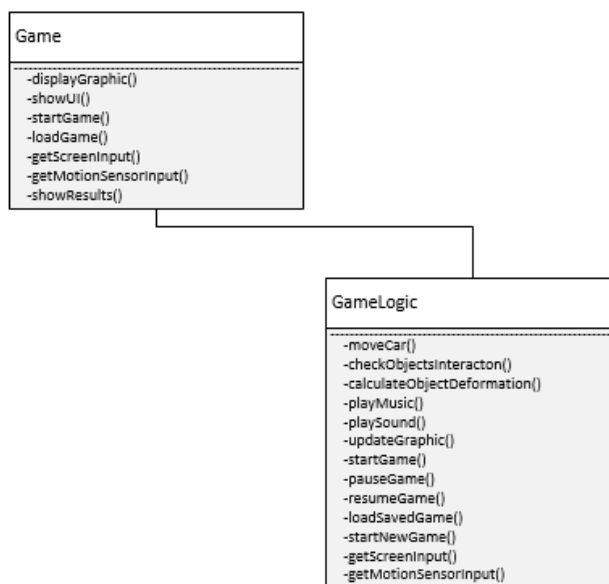


Paveikslēlis 5 Sistemos statinis vaizdas

3.3. Detalzuotas sistemos statinis vaizdas

Paketas Game Core

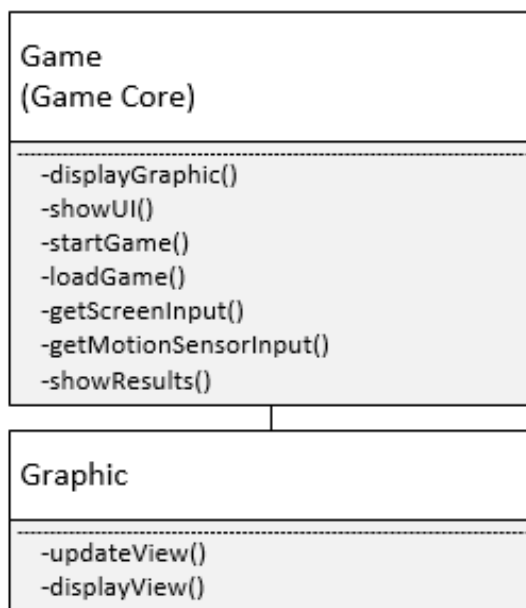
Šis paketas sudaro žaidimo branduolį. Visas žaidimo funkcijas ir kitus komponentus iškviečia šis paketas.



Paveikslēlis 6 Žaidimo branduolys

Paketas Graphic System

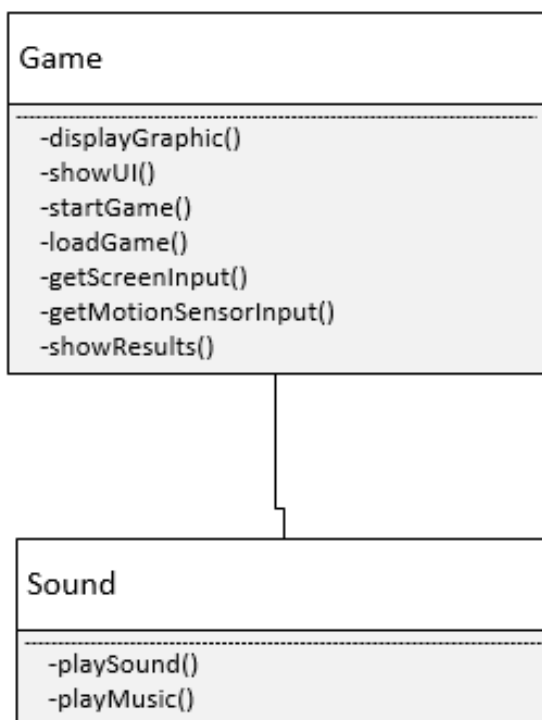
Šis paketas atsakingas už žaidimo grafikos atvaizdavimą.



Paveikslėlis 7 Paketas Graphic System

Sound System

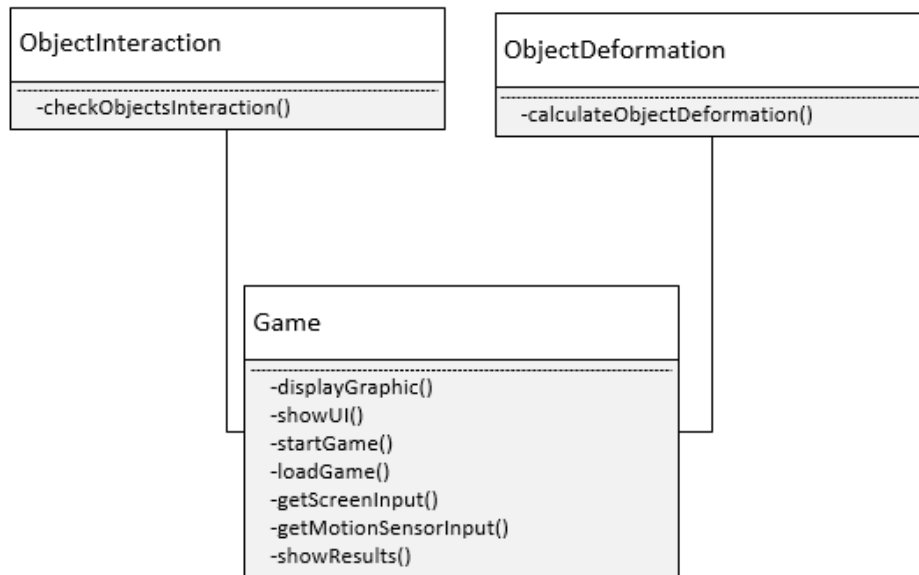
Šis paketas sudaro žaidimo garso sistemą, kuri atsakinga už žaidimo garsų atkūrimą.



Paveikslėlis 8 Garso sistema

Physic System

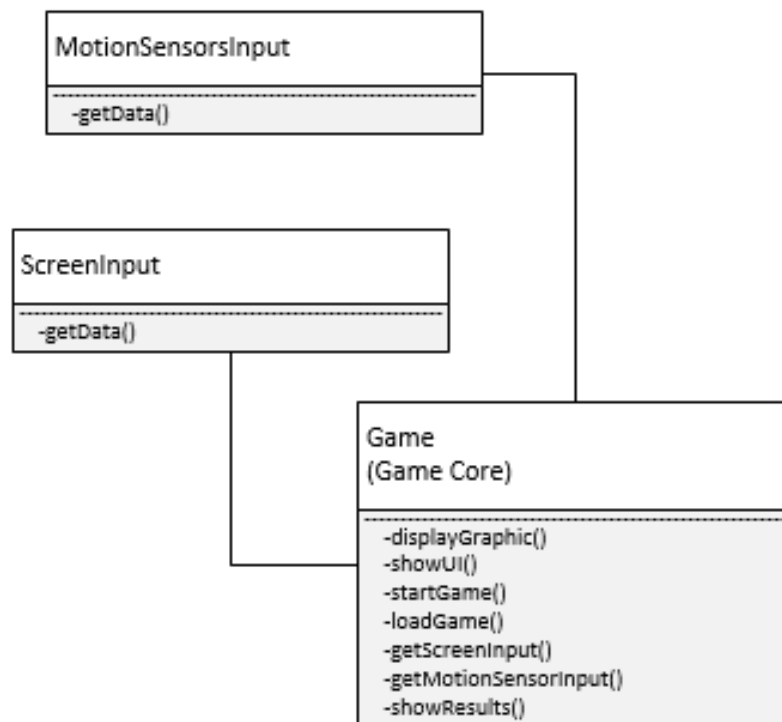
Šis paketas atsakingas už žaidimo fiziką. Jis skirtas objektų deformacijai ir sąveikai tarp objektų apskaičiuoti.



Paveikslėlis 9 Paketas Physic system

Inptut System

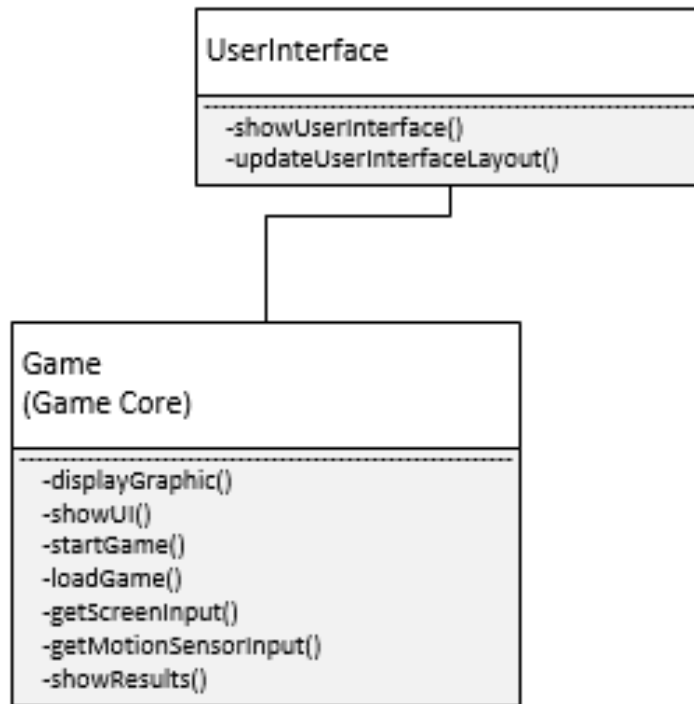
Šis paketas atsakingas už duomenų gavimo iš mobiliojo įrenginio įvedamų duomenų (lietumui jautraus ekrano, akselerometro ir giroskopo).



Paveikslėlis 10 Paketas Input System

User Interface

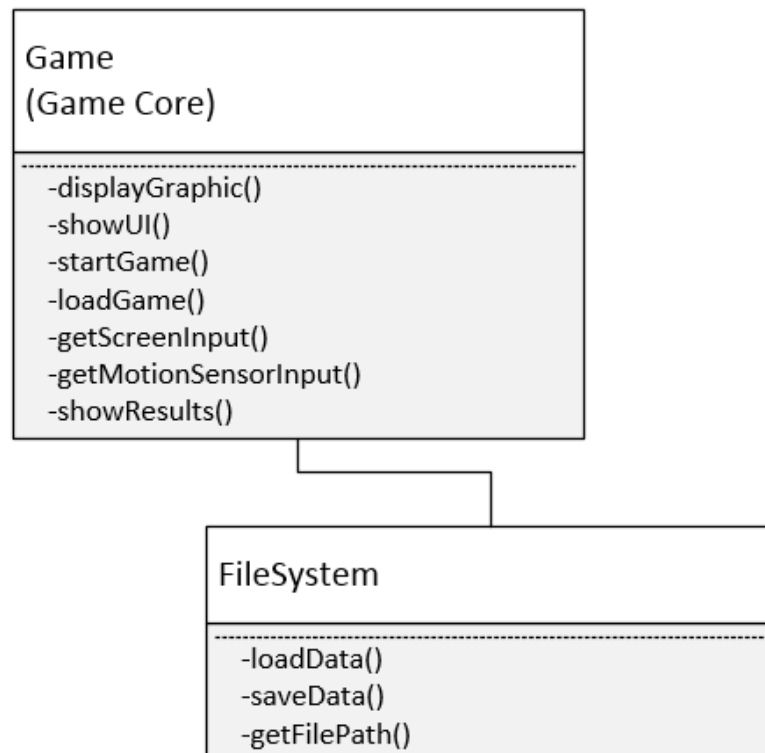
Šis paketas atsakingas už vartotojo sąsajo atvaizdavimą.



Paveikslėlis 11 Paketas vartotojo sąsaja

File System

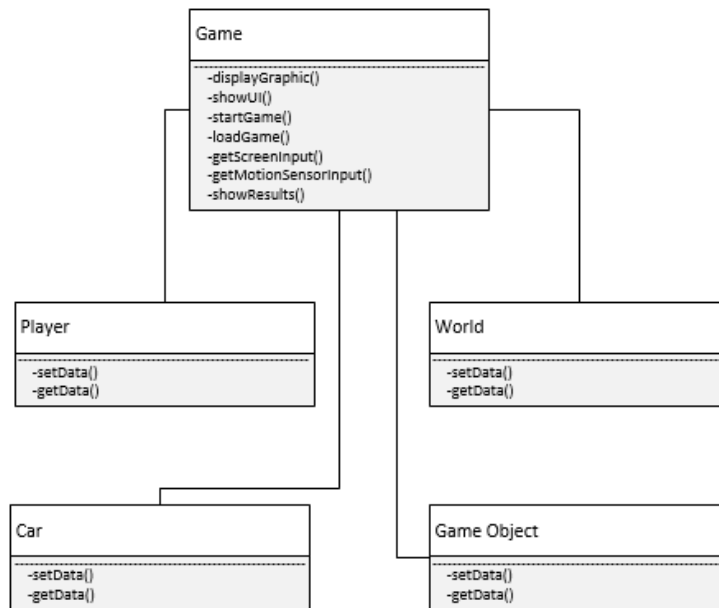
Šis paketas skirtas duomenų nuskaitymui ir įrašymui iš mobiliojo įrenginio vidinės atminties.



Paveikslėlis 12 Paketas File System

Data Object

Šiame pakete bus saugomi užkrauti žaidimo duomenys ir įvairūs žaidimo objektai.



Paveikslėlis 13 Paketas Data Object

4. TYRIMO DALIS

Tyrimo dalyje nagrinėjama sukurtos programinės įrangos kokybė.

4.1. Tikslas

Tyrimo dalies tikslas išanalizuoti sukurtą programinę įrangą. Analizavimo metu bus įvertinta programos atitikimas užsibrėžtiems tikslams. Taip pat bus įvertinama ar programa pilnai atitiko tai, ko iš jos buvo siekiama, ar programa veikė tinkamai.

4.2. Vizualus deformacijų kokybės ir tikslumo vertinimas

Tiriant programinę įrangą, bandoma atsižvelgti į sukurtą žaidimo kokybę. Pagrindinis vertinimo kriterijus vertinant sukurtą žaidimą yra įvertinti pateikiamų deformacijų tikslumą.

Įvertinti, kaip realistiškai algoritmas deformuoja objektus yra sudėtinga. Todėl visas deformacijų tikslumas ir realistiškumas vertinimas buvo atliekamas vizualiai.

Vertinimo metu, buvo apdaužomas automobilis, ir po apdaužymo buvo vertinama, kaip vizualiai skiriasi automobilio kūbulas prieš smūgį ir po jo.

Pirmam bandymui buvo numestas kietas kubo formos kūnas ant automobilio. Numestas objektas, atsitrenkęs į automobilio stogo šoną sukėlė deformacijas. Tai pateikta žemiau pateiktame paveikslėlyje.



Paveikslėlis 14 Deformacijų sukėlimas iš viršaus

Kai žaidimas apskaičiuoja ir vizualiai atlieka deformacijas, jos bandomos įvertinti vizualiai. Iš žemiau pateikto paveikslėlio galima matyti, kad deformacijos kryptis ir vieta apytiksliai atitinka numatomą deformaciją.

Taip pat galima pastebėti, kad automobilis laikomas vienalyčiu kūnu. Automobilio stiklas, kuris yra iš kitos medžiagos, nei automobilio kėbulas, tačiau kaip galima vizualiai įvertinti iš pateikto paveikslėlio žemiau, jis linko taip pat, kaip plieninis kėbulas



Paveikslėlis 15 Deformacijų iš viršaus vizualus vertinimas

Sekančiame bandyme, buvo bandoma deformuoti automobilį gavus smūgį į automobilio galą. Sudarant smūgį į automobilio galą, buvo bandoma imituoti smūgį galu į nejudančio objekto kampą. Smūgio vizualus pavaizdavimas pateiktas žemiau esančiame paveikslėlyje.



Paveikslėlis 16 Defromacijų sukėlimas automobilio gale

Po smūgio galima vizualiai galima įvertinti padarytą žalą automobilio kėbului, bei įvertinti deformacijos tikslumą. Iš žemiau pateikto paveikslėlio galima matyti, kad deformacijos vieta atitiko vietą, kuri buvo paveikta išorinės jėgos, kurią sukėlė smūgis į automobilio galą. Taip pat matoma, kad deformacijos forma taip pat apytiksliai atitinka formą, į kurią automobilis atsitrenkė.



Paveikslėlis 17 Atliktų deformacijų automobilio gale vizualus vertinimas

Sekančiame bandyme, buvo bandoma deformuoti automobilio priekį, kadangi vertinant vizualiai, automobilio priekis yra sudėtingesnė struktūra.

Atliktas smūgis, automobiliui važiuojant į priekį. Smūgis buvo atliktas kampu. Smūgio vizualus atvaizdas pateiktas žemiau esančiame paveikslėlyje.



Paveikslėlis 18 Deformacijų sukėlimas automobilio priekyje

Po atlikto smūgio ir atliktos deformacijos, buvo vertinamas deformacijos tikslumas. Kaip ir prieš tai buvusiais deformacijos bandymais, smūgio vieta ir kryptis apytiksliai atitinka smūgio vietą.

Tačiau šiuo atveju deformacijos detalumas vizualiai atrodo mažiau tikslus. Nors ir automobilio priekinis bamperis deformuotas, gan realistiškai, tačiau kitų dalių deformacija atrodo mažiau detali ir realistiška. Automobilio žibintas atrodo nepažeistas, tik vizualiai atliko poslinkio smūgio vektorius kryptimi. Taip pat galima pastebėti, kad aplink esančios detalės deformuotos nerealistiškai. Variklio gaubti kampas, kuris buvo paveiktas išorinės jėgos, vizualiai deformuotas nerealistiškai, dalis jo išnyko.

Atliktos deformacijos pateiktos žemiau esančiame paveikslėlyje.



Paveikslėlis 19 Deformuoto automobilio priekio vizualus įvertinimas

Atlikus vizualaus deformacijos vertinimo bandymus, taip pat buvo vizualiai įvertinta deformacijų sparta. Vertinant spartą buvo pastebėta, kad deformacijų skaičiavimas turi pastebimą žaidimo veikimo sulėtėjimą. Automobilį paveikiant išorine jėga, buvo jaučiamas 1-2 sekundžių sulėtėjimas, kuris jaučiamas ir trukdo žaisti.

4.3. Rezultatų apibendrinimas

Atlikus vizualinį deformacijų įvertinimą, galima apibendrinti šiuos dalykus:

4. Deformacijų vieta atitinka vietą, kuri buvo paveikta išorinės jėgos;
5. Deformacijos kryptis atitinka smūgio krypties vektorių, kurs gaunamas paveikus objektą išorine jėga;

Taip pat buvo pastebėti trūkumai, atliekant deformacijas:

- Visas automobilio kūbulas laikomas vieno medžiagos;
- Smulkios detalės deformuojamos nerealistiškai;
- Detalios automobilio vietos deformuojamos nerealistiškai;
- Jaučiamas sulėtėjimas atliekant deformacijas.

Peržvelgus rezultatus, nuspręsta, kad šiuo metu didinti detalumą sudėtinga, kadangi žaidžiant žaidimą jaučiamas sulėtėjimas. Dėl šios priežasties, buvo nuspręsta pabandyti didinti deformacijų spartą, tikslumo sąskaita.

5. EKSPERIMENTINĖ DALIS

Ekspirimentinėje dalyje, bus įvertinimas programos veikimas.

5.1. Tikslas

Eksperimentinės tyrimo tikslas, kiekybiškai įvertinti sukurtą programinę įrangą. Tyrimo metu, bus skaičiuojama iš anksto apibrėžto proceso trukmė. Procesas bus kartojamas kelis kartus, su pakeistais parametrais.

5.2. Tyrimo aprašymas

Atliekant tyrimą, sudaromos sąlygos, kurių metu galima kontroliuoti vykdomą procesą. Tyrimo metu, naudojamas spyruoklių deformacijos algoritmas, kuris realizuotas programinės įrangos kūrimo metu.

Atliekamo proceso metu, bus kartojamas deformacijos procesas. Jo metu, atliekamas iš anksto apibrėžtas deformacijos procesas su nurodomomis sąlygomis. Atliekant eksperimentą, skaičiuojama deformacijos proceso trukmė.

Tyrimo metu sudaromas sintetinis testas, kurio metu ignoruojamos įvairios sąlygos, kurias gali turėti įtakos atliekamam deformacijos procesui ar trukmei, kuri skaičiuojama deformacijos metu.

Atliekant tyrimą, paimamas iš anksto apibrėžtas objektas. Vėliau, objektas deformuojamas. Atliekant tyrimą, objektas deformuojamas tris kartus. Atlikus 3 bandymus, sudaroma trijų deformacijų trukmių lentelė, kurios pagalba sudaromas vidutinis laikas, reikalingas atlikti nurodytai deformacijos operacijai.

Atlikus tris deformacijas, modifikuojamas tiriamas objektas. Modifikuojant objektą, keičiamas 3d modelio sudėtingumas. Keičiant objekto sudėtingumą, keičiamas 3d objekto viršūnių kiekis. Pakeitus viršūnių kiekį, kartojamas deformacijos procesas.

Atlikus šiuos procesus, modifikuojamas deformacijos algoritmas. Jis pakeičiamas, taip kad būtų supaprastintas smūgio taškų aptikimas. Paprastai smūgio metu gali būti fiksuojami kelio smūgio taškai, jų kiekis priklauso nuo analizuojamo objekto sudėtingumo. Smūgio taškų kiekis nusako, kiek kartų reiks kartoti deformacijos algoritmą, vienai vietai, kuri buvo paveikta išorinės jėgos. Pakeistas metodas fiksuos tik vieną smūgio tašką, taip siekiant pamatuoti, kiek pasikeis deformacijos sparta.

Atlikus tyrimus, analizuojami gauti duomenys. Visi duomenys įtraukiami į lentelę. Įtraukus duomenis į lentelę, palyginamas vidutinis laikas, reikalingas atlikti deformacijos procesui. Sudarius laikų lentelę, taip pat sudaromas grafikas, kuris skirtas vizualiai įvertinti modelio sudėtingumo ir deformacijos atlikimo trukmės santykį ir pokytį, gaunamą keičiant deformuojamo objekto sudėtingumą.

5.3. Eksperimento įranga

Deformacijų eksperimento tyrimas buvo atliekamas dvejose sistemose. Vienodas algoritmas buvo paleidžiamas mobiliajame telefone ir asmeniniame kompiuteryje.

Mobiliojo telefono parametrai:

Lentelė 16 Mobiliojo telefono parametrai

Mobiliojo telefono modelis	LG Optimus L5 II E460
Operacinė sistema	Android 4.1 JellyBean
Procesorius	Mediatek MT6575, 1GHz
Vaizdo procesorius	PowerVR SGX531
Operatyvioji atmintis	512MB
Ekranas dydis	4 coliai
Ekranas raiška	480 x 800

Asmeninio kompiuterio parametrai:

Lentelė 17 Asmeninio kompiuterio parametrai

Kompiuterio modelis	Acer Aspire V3-571G
Operacinė sistema	Windows 8.1 Professional
Procesorius	Intel i3-3110M, 2,4GHz
Vaizdo procesorius	NVidia Geforce 630m
Operatyvioji atmintis	8096MB
Ekranas dydis	15,6 coliai
Ekranas raiška	1366x768

5.4. Pradinio tyrimo rezultatai

Pradinis tyrimas atliekamas su nepakeistu, programoje realizuotu algoritmu.

Tyrimo metu gauti rezultatai surašyti į lentelę. Joje pateiktas nagrinėjamo objekto sudėtingumas, kuris išreikštas viršūnėmis, kurios nusako objekto detalumą. Taip pat lentelėje įtraukti kiekvieno deformacijos proceso laikai, kurie buvo gauti atliekant deformacijas. Iš gautų deformacijų išvedamas vidutinis laikas kuris, kuris vėliau naudojamas vertinant rezultatus. Taip pat sudaromas grafikas, kuris vizualiai pateikia deformacijos priklausomybę nuo nagrinėjamo modelio viršūnių kiekio.

5.4.1. Tyrimo rezultatai atlikti mobiliajame įrenginyje

Gautų rezultatų lentelė atlikus tyrimą mobiliajame įrenginyje:

Lentelė 18 Rezultatai mobiliajame telefone

Viršūnių kiekis	Bandymas 1	Bandymas 2	Bandymas 3	Vidurkis
38	14	12	14	13,33333
138	45	41	44	43,33333
302	159	157	156	157,3333
530	478	482	476	478,6667
822	1268	939	891	1032,667
1178	1213	1192	1289	1231,333
1598	1887	1944	1830	1887
2082	2817	2493	2529	2613
2630	3609	3585	3553	3582,333
3242	5047	4976	4590	4871
3918	5798	6213	5852	5954,333
4658	7236	7297	7611	7381,333
5462	9067	9284	9013	9121,333
6330	11512	12047	11199	11586

Žemiau pateiktas grafikas, kuris nurodo deformacijos laiko reikalingos atlikti deformaciją ir nagrinėjamo modelio viršūnių kiekio priklausomybę.



5.4.2. Tyrimo rezultatai atlikti asmeniniame kompiuteryje

Gautų rezultatų lentelė atlikus tyrimą asmeniniame kompiuteryje:

Lentelė 19 Rezultatai asmeniniame kompiuteryje

Viršūnių kiekis	Bandymas 1	Bandymas 2	Bandymas 3	Vidurkis
38	5	5	5	5
138	18	15	9	14
302	43	41	28	37,33333
530	105	76	76	85,66667
822	120	127	120	122,33333
1178	190	186	183	186,33333
1598	263	267	264	264,66667
2082	376	371	373	373,33333
2630	501	567	494	520,66667
3242	634	633	634	633,66667
3918	808	806	807	807
4658	1004	998	1020	1007,333
5462	1221	1226	1224	1223,667
6330	1489	1486	1483	1486

Žemiau pateiktas grafikas, kuris nurodo deformacijos laiko reikalingos atlikti deformaciją ir nagrinėjamo modelio viršūnių kiekio priklausomybę.



5.5. Patobulintos PĮ tyrimo rezultatai

Šiame etape modifikuojamas algoritmas, kad būtų fiksuojamas tik vienas smūgio taškas, taip bandant supaprastinti algoritmą ir paspartinti skaičiavimus.

Tyrimo metu gauti rezultatai surašyti į lentelę. Joje pateiktas nagrinėjamo objekto sudėtingumas, kuris išreikštas viršūnėmis, kurios nusako objekto detalumą. Taip pat lentelėje įtraukti kiekvieno deformacijos proceso laikai, kurie buvo gauti atliekant deformacijas. Iš gautų deformacijų išvedamas vidutinis laikas kuris, kuris vėliau naudojamas vertinant rezultatus. Taip pat sudaromas grafikas, kuris vizualiai pateikia deformacijos priklausomybę nuo nagrinėjamo modelio viršūnių kiekio.

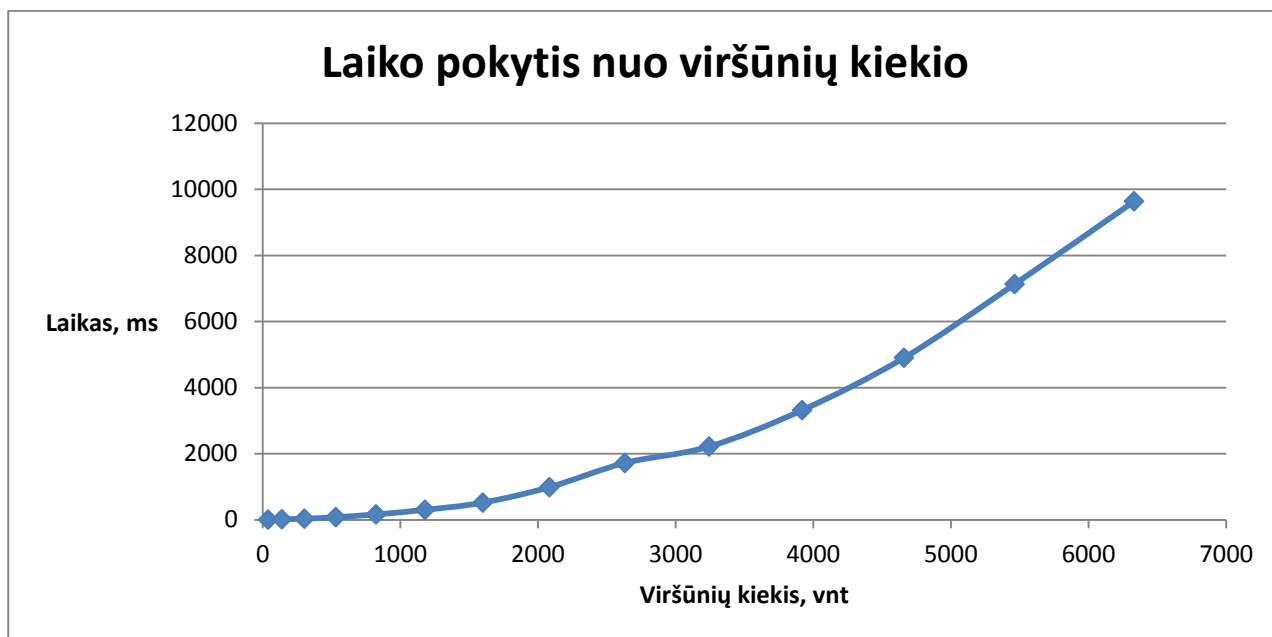
5.5.1. Tyrimo rezultatai atlikti mobiliajame įrenginyje

Rezultatai gauti atlikus modifikuoto metodo tyrimą:

Lentelė 20 Patobulintos PĮ rezultatai asmeniniame kompiuteryje

Viršūnių kiekis	Bandymas 1	Bandymas 2	Bandymas 3	Vidurkis
38	2	3	2	2,333333
138	17	9	15	13,66667
302	27	30	34	30,33333
530	58	87	86	77
822	111	195	179	161,6667
1178	136	402	362	300
1598	201	678	672	517
2082	390	1146	1420	985,3333
2630	836	2279	2038	1717,667
3242	657	3021	2965	2214,333
3918	947	4557	4443	3315,667
4658	1227	6729	6744	4900
5462	1481	9625	10280	7128,667
6330	2241	13316	13350	9635,667

Diagrama, gauta iš rezultatų atlikus modifikuoto metodo tyrimą mobiliajame įrenginyje.



5.5.1. Tyrimo rezultatai atlikti asmeniniame kompiuteryje

Rezultatai gauti atlikus modifikuoto metodo tyrimą kompiuteryje:

Viršūnių kiekis	Bandymas 1	Bandymas 2	Bandymas 3	Vidurkis
38	0	0	0	0
138	3	3	3	3
302	7	9	5	7
530	16	14	10	13,33333
822	29	22	22	24,33333
1178	36	41	41	39,33333
1598	45	73	72	63,33333
2082	45	119	119	94,33333
2630	65	183	182	143,3333
3242	122	275	275	224
3918	107	417	397	307
4658	147	556	558	420,3333
5462	180	760	788	576
6330	246	1024	1024	764,6667

Diagrama gauta iš rezultatų, kurie gauti atlikus modifikuoto metodo tyrimą kompiuteryje:

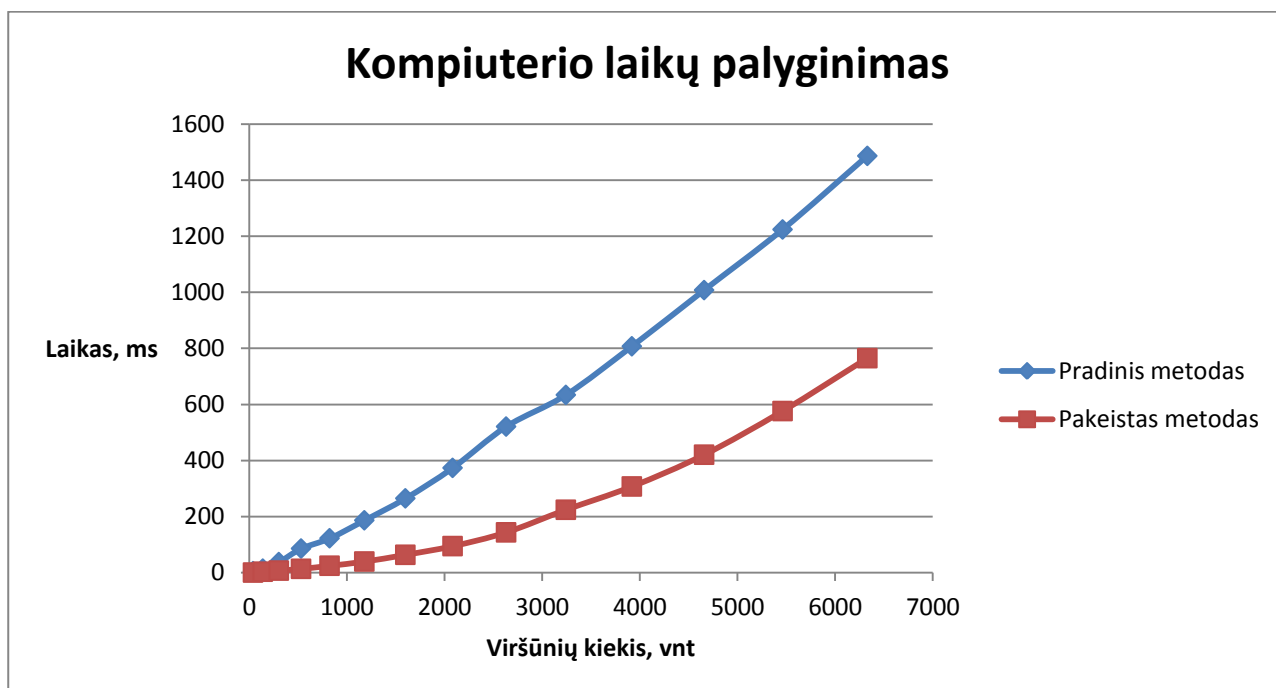
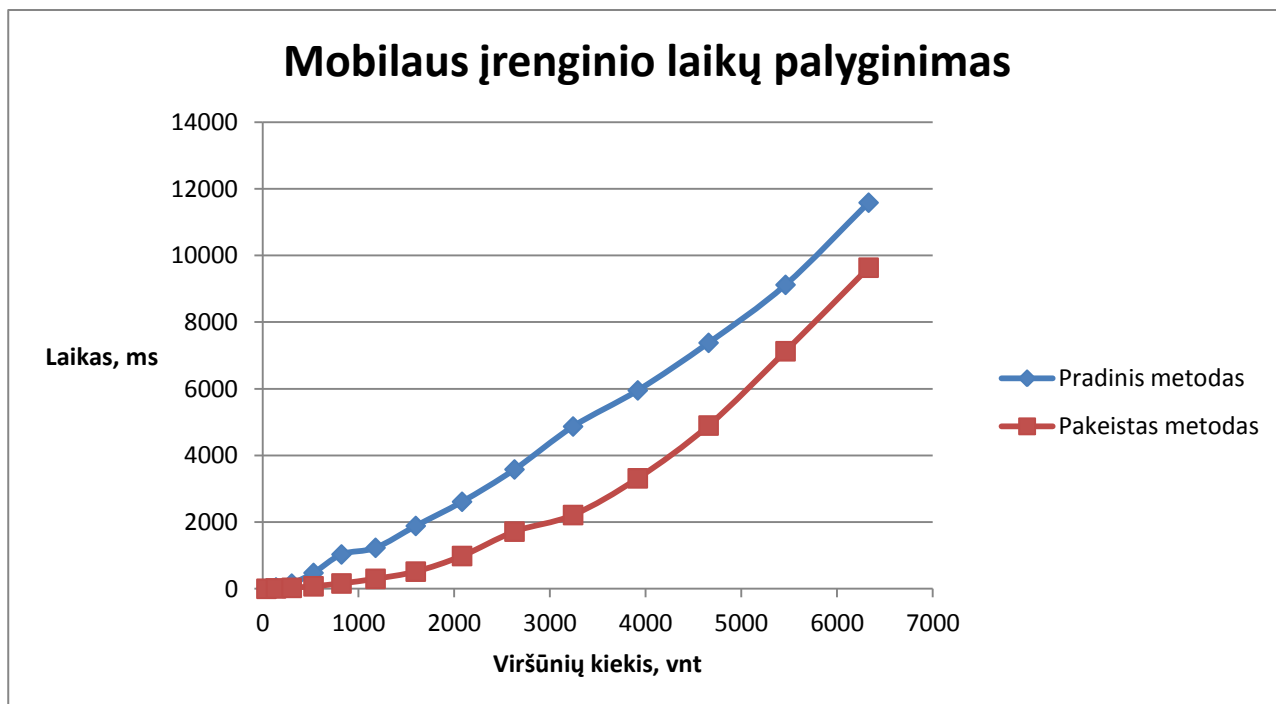


5.6. Rezultatų analizė

Atlikus eksperimentą, galima teigti, kad yra tiesinė priklausomybė, tarp deformacijos proceso trukmės ir naudojamo modelio sudėtingumo. Naudojant paprastesnius modelius, deformacijos procesai vyksta sparčiau. Didėjant modelio detalumui, deformacijos tampa lėtomis, kas žaidime tampa nepriimtina. Mažinant deformuojamo objekto detalumą, deformacija vyksta sparčiau, tačiau vizualinis objekto atvaizdas tampa vizualiai prastesnis.

Atlikus tyrimą ir palyginus modifikuotą metodą ir nepakeistą metodą, galima teigti, kad smūgio taškų kiekis turi įtaką deformacijų atlikimo spartai. Sumažinus deformacijos taškų aptikimą, deformacijos procesas vyksta sparčiau, tačiau dėl to nukenčia deformacijos detalumas.

Apibendrinus rezultatus, galima teigti, kad nors ir buvo sumažintas taškų kiekis, kuris gaunamas paveikus kūną išorine jėga, iki minimumo, taip sumažinant deformacijos apskaičiavimui skirtų iteracijų kiekį, tačiau su detalesniais modeliais sparta vis tiek nepriimtina. Šiuo atveju, reiktų pabandyti perprojektuoti deformacijų algoritmą taip, kad būtų galima panaudoti lygiagrečius skaičiavimus. Spyruoklių metodo vienas iš privalumų, kad jis tinkamas naudoti lygiagrečiams procesams, todėl teoriškai pasvarsčius galima teigti, kas sparta turėtų padidėti. Taip pat lygiagretūs skaičiavimui turėtų padidinti spartą, kadangi dauguma įrenginių turi kelis skaičiavimo branduolius, kas turėtų padidinti lygiagrečių procesų išnaudojimą.



6. IŠVADOS

Atlikus šį darbą buvo sukurtas žaidimas, kuriame realizuotas deformacijų atvaizdavimas. Apskaičiuoti deformacijoms, naudojamas spyruoklių metodas.

Sukurtas žaidimas skirtas, mobiliesiems įrenginiams, kurie turi ribotus resursus, todėl atliekant tyrimą, buvo atsižvelgiama į deformacijos algoritmo spartą.

Vizualiai įvertinus deformacijų algoritmą, galima teigti, kad deformacijos atitinka išorinės jėgos vektoriau kryptį, taip pat deformacijos tinkamai atvaizduoja smūgio vietą, tačiau deformacijų detalumas ir kokybė nėra aukšta.

Atlikus eksperimentinę dalį buvo paskaičiuota, kad didelę įtaką deformacijų spartai turi deformuojamo modelio detalumas. Peržiūrėjus rezultatus, gaunama tiesinė laiko priklausomybė nuo modelyje naudojamų viršūnių kiekio. Sumažinus taškų, kuriuos paveikia išorinė jėga aptikimą, daugeliu atvejų deformacijos skaičiavimai paspartėja. Esant nedetaliems 3D modeliams, šis paspartėjimas beveik nesijaučia, tačiau sudėtingėjant modeliui, skirtumas didėja.

Apibendrinus darbą, galima teigti:

1. Projektuojant žaidimą, buvo pasirinktas Unity žaidimo variklis, nes jis supaprastina žaidimo kūrimą skirtingoms platformoms;
2. Deformacijos apskaičiuoti buvo pasirinktas spyruoklių deformacijos algoritmas;
3. Deformacijos modeliams neturintiems daug smulkių detalių atrodo pakankamai realistiškai, tačiau modeliuose kurie turi daug skirtingų detalių, deformacijos realistiškumas mažėja;
4. Deformacijų sparta priklauso nuo modelio detalumo, modeliai turintys daug viršūnių užima daugiau laiko deformacijų apskaičiavimams;
5. Sumažinus išorinės jėgos paveiktų taškų kiekį, deformacijos apskaičiavimai paspartėja, tačiau detalumas stipriai sumažėja.

7. LITERATŪRA

[1] What is Rendering?

Prieiga internete: (<http://3d.about.com/od/3d-101-The-Basics/a/Rendering-Finalizing-The-3d-Image.htm>);

[2] About Polygon Meshes

Prieiga internete: (http://softimage.wiki.softimage.com/xsidocs/poly_basic_PolygonMeshes.htm)

[3] An example of a polygon mesh.

Chrschn

http://commons.wikimedia.org/wiki/File:Dolphin_triangle_mesh.png

[4] Weishan Zhang; Dong Han; Kunz, T.; Hansen, K.M., "Mobile Game Development: Object-Orientation or Not," *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International* , vol.1, no., pp.601,608, 24-27 July 2007

doi: 10.1109/COMPSAC.2007.151

prieiga

internetė: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4291059&isnumber=4290963>

[5] Jie, Jiang; Yang, Kuang; Haihui, Shen, "Research on the 3D Game Scene Optimization of Mobile Phone Based on the Unity 3D Engine," *Computational and Information Sciences (ICCIS), 2011 International Conference on* , vol., no., pp.875,877, 21-23 Oct. 2011

doi: 10.1109/ICCIS.2011.317

Prieiga

internetė: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6086340&isnumber=6086119>

[6] The Standard for Embedded Audio Acceleration

Prieiga internete: <https://www.khronos.org/opensles/>

[7] About SDL

Prieiga internete: <http://www.libsdl.org/>

[8] A game programming library

Prieiga internete: <http://alleg.sourceforge.net/readme.html>

[9] Simple and Fast Multimedia Library

Prieiga internete: <http://www.sfml-dev.org/>

[10] 3D Rendering

Prieiga internete: <http://www.3drender.com/glossary/3drendering.htm>

[11] Xueyan Hu; Zhonglin Xia; Zhiyong Yuan, "Study on Image Deformation Simulation Based on ARM Linux and OpenGL ES," *Intelligence Science and Information Engineering (ISIE), 2011 International Conference on* , vol., no., pp.303,306, 20-21 Aug. 2011

doi: 10.1109/ISIE.2011.40 [Žiūrėta 2013 12 04], prieiga

internetė <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5997441&isnumber=5997361>>

[12] Ng Moon Hui; Liu Ban Chieng; Wen Yin Ting; Mohamed, H.H.; Rafie Hj Mohd Arshad, M., "Cross-platform mobile applications for android and iOS," *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP* , vol., no., pp.1,4, 23-25 April 2013

doi: 10.1109/WMNC.2013.6548969, prieiga

internete <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6548969&isnumber=6548948>>

[13] Fahy, R.; Krewer, L., "Using open source libraries in cross platform games development," Games Innovation Conference (IGIC), 2012 IEEE International , vol., no., pp.1,5, 7-9 Sept. 2012

[14] Sarinho, V.T.; Apolinário, A.L., "A Generative Programming Approach for Game Development," Games and Digital Entertainment (SBGAMES), 2009 VIII Brazilian Symposium on , vol., no., pp.83,92, 8-10 Oct. 2009

doi: 10.1109/SBGAMES.2009.18 [Žiūrēta 2013 12 04], prieiga internete <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5479106&isnumber=5479088>>

[15] Davis J; Tierney, A.; Chang, E., "A user adaptable user interface model to support ubiquitous user access to EIS style applications," Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International , vol.1, no., pp.351,358 Vol. 2, 26-28 July 2005

doi: 10.1109/COMPSAC.2005.37 [Žiūrēta 2013 12 04], prieiga internete <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1510046&isnumber=32335>>

[16] Halic, T.; Kockara, S.; Bayrak, C.; Rowe, R.; Chen, B., "Soft Tissue Deformation and Optimized Data Structures for Mass Spring Methods," Bioinformatics and BioEngineering, 2009. BIBE '09. Ninth IEEE International Conference on , vol., no., pp.45,52, 22-24 June 2009 [Žiūrēta 2013 11 26], prieiga internete <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5211316&isnumber=5211104>>

[17] U. Meier, O. López, C. Monserrat, M.C. Juan, M. Alcañiz, Real-time deformable models for surgery simulation: a survey, Computer Methods and Programs in Biomedicine, Volume 77, Issue 3, March 2005, Pages 183-197, ISSN 0169-2607, <http://dx.doi.org/10.1016/j.cmpb.2004.11.002> [Žiūrēta 2013 12 03]. Prieiga internete: <<http://www.sciencedirect.com/science/article/pii/S0169260704002093>>

[18] Nedel, L.P.; Thalmann, D., "Real time muscle deformations using mass-spring systems," Computer Graphics International, 1998. Proceedings , vol., no., pp.156,165, 22-26 Jun 1998

doi: 10.1109/CGI.1998.694263 [Žiūrēta 2013 11 26], prieiga internete <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=694263&isnumber=15133>>

[19] Min Hong; Jung, S.; Min-Hyung Choi; Welch, S.W.J., "Fast Volume Preservation for a Mass-Spring System," Computer Graphics and Applications, IEEE , vol.26, no.5, pp.83,91, Sept.-Oct. 2006 doi: 10.1109/MCG.2006.104 [Žiūrēta 2013 11 26], prieiga internete <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1683697&isnumber=35448>>

[20] Jian-Kun Shen; Matuszewski, B.J.; Shark, L.; Moore, C.J., "Deformable Registration using Spring Mass System with Cross-section Correction," Medical Information Visualisation - BioMedical Visualisation, 2007. MediVis 2007. International Conference on , vol., no., pp.9,14, 4-6 July 2007

doi: 10.1109/MEDIVIS.2007.10 [Žiūrēta 2013 11 26], prieiga internete <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4272104&isnumber=4272095>>

[21] Frisken-Gibson, S.F., "Using linked volumes to model object collisions, deformation, cutting, carving, and joining," Visualization and Computer Graphics, IEEE Transactions on , vol.5, no.4, pp.333,348, Oct-Dec 1999

doi: 10.1109/2945.817350 [Žiūrėta 2013 12 03], prieiga internete <
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=817350&isnumber=17715>>

[22] Pulli, K.; Aarnio, T.; Roimela, K.; Vaarala, J., "Designing graphics programming interfaces for mobile devices," Computer Graphics and Applications, IEEE , vol.25, no.6, pp.66,75, Nov.-Dec. 2005

doi: 10.1109/MCG.2005.129 [Žiūrėta 2013 11 26], prieiga internete
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1528436&isnumber=32639>>

[23] The Standard for Embedded Accelerated 3D Graphics [Žiūrėta 2013 12 01], prieiga internete
<<http://www.khronos.org/opengles/>>

[24] Moller, R., "State-of-the-Art 3D Graphics for Embedded Systems," Devices, Circuits and Systems, Proceedings of the 6th International Caribbean Conference on , vol., no., pp.339,343, 26-28 April 2006

doi: 10.1109/ICCDCS.2006.250884 [Žiūrėta 2013 12 01], prieiga internete
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4016913&isnumber=4016847>>

[25] Inhye Park; Hyungkeun Lee; Hyukjoon Lee, "Software optimization for embedded communication system," Information Networking (ICOIN), 2013 International Conference on , vol., no., pp.676,679, 28-30 Jan. 2013

doi: 10.1109/ICOIN.2013.6496708 [Žiūrėta 2013 12 04], prieiga internete
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6496708&isnumber=6496336>>

8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

3D grafika – grafika atvaizduojama 3 dimensijose.

2D – dviejų dimensijų objektas

OpenGL ES – grafikos biblioteka Android operacinėms sistemoms.

Android – tai operacinė sistema skirta mobiliesiems telefonams. Android yra atvirojo kodo operacinė sistema naudojanti Linux branduolį.

OGRE – (Object-Oriented Graphics Rendering Engine) tai realaus laiko, 3D grafikos atvaizdavimo variklis

MIP maps – iš anksto apskaičiuota, optimizuota tekstūrų seka, kuri pateikiama su pagrindine tekstūra

SDL – tai biblioteka, suteikianti priėjimą prie žemo lygio įrangos.

LOD – (Level of detail) tai technologija, kurios pagalba toli esantys 3d objektai praranda detalumą, kad sumažėtų apkrova kompiuteriui.