



Latvijas Universitāte  
Datorikas fakultāte

Tīmekļa vietnes manipulācija no skārienjūtīga  
ekrāna ierīces

Kvalifikācijas darbs

Autors:

Mārtiņš Bērziņš

Stud. apl. nr.:

mb10124

Vadītājs:

M.Dat, Oļegs Urziks

Rīga, 2012

## ANOTĀCIJA

Ņemot vērā lielo pieprasījumu pēc iekārtām ar skārienjūtīgiem ekrāniem. Tika izveidota sistēma, kura nodrošina izklaides iespēju tās lietotājam. Sistēma tas ir paredzēts tā ka tās lietotājs izmantojot savu „iPhone” kā pulti var spēlēt spēli „Pong”. Spēle „Pong” ir uz programmēšanas valodas Java veidota, spēle seko galda tenisa principiem. „iPhone” programmēšanai izmanto programmēšanas valodu Objective-C. Sistēmas lietošanas princips ir lietotājs savā tīmekļa pārlūkprogrammā atver vietni, kur parāda spēli „Pong”, tad lietotājs savā „iPhone” ieslēdz vadības pults lietotni un kontrolē spēlē notiekošo.

Tīmekļa vietnes manipulācija no skārienjūtīga ekrāna ierīces sistēma ir veidota kā izpētes sistēma ar domu, vai šāda veida sistēmu attīstot tālāk to varētu pārvērst par ienesīgu uzņēmējdarbības iespēju.

Atslēgas vārdi: Java, Objective C, iPhone,Pong.

## ABSTRACT

Considering huge demand for touchscreen devices, this system was made for entertainment purposes for its user. Usage of system is, user uses his „iPhone” as remote for game „Pong”. Game „Pong” is made in programming language Java, and it follows principles of table tennis. Programming language for „iPhone” is Objective-C.

Principle how to uses this system is like this, user in his web browser opens site location address which shoves „Pong” game, then in his „iPhone” launches application for remote and controls the actions in game.

Manipulation of web content with touch screen device system is developed with thought , if system like this after improvements could be made in profitable business possibility.

Key words: Java, Objective C, iPhone,Pong.

## Satura rādītājs

1.	Programmatūras Prasību Specifikācija.....	7
1.1	Ievads .....	7
1.1.1	Darbības sfēra .....	7
1.1.2	Definīcijas.....	7
1.1.3	Saistība ar citiem dokumentiem .....	7
1.1.4	Dokumenta pārskats .....	7
1.2	Vispārējais apraksts .....	7
1.2.1	Produkta perspektīva .....	8
1.2.2	Biznesa prasības .....	8
1.2.3	Produkta funkcijas .....	8
1.2.4	Lietotāja raksturiezīmes.....	9
1.2.5	Vispārējie ierobežojumi.....	9
1.2.6	Pieņēmumi un atkarības .....	9
1.3	Funkcionālās prasības .....	9
1.3.1	„iPhone” lietotnes „Controller” funkcijas. ....	9
1.3.2	Sistēmas servera funkcijas:.....	11
1.3.3	Spēles „Pong” funkcijas .....	12
1.4	Arējās saskarnes prasības.....	17
1.4.1	Sakaru saskarne .....	17
1.5	Nefunkcionālās prasības .....	17
1.5.1	Ātrdarbība.....	17
1.5.2	Datu drošība.....	17
2.	Programmatūras projektējuma apraksts .....	18
2.1	Ievads .....	18
2.2	Nolūks.....	18
2.3	Darbības sfēra .....	18
2.4	Izmantotie jēdzieni un saīsinājumi.....	18
2.5	Saistība ar citiem dokumentiem.....	18
2.6	Dekompozīcijas apraksts .....	19
2.6.1	Moduļu dekompozīcija.....	19
2.7	Moduļu atkarības apraksts. ....	24
2.8	Moduļu detalizēts apraksts.....	26
3.	Testēšanas dokumentācija.....	32

3.1	Ievads .....	32
3.2	Manuālie testi.....	32
4.	Lietotāja dokumentācija .....	35
4.1	Ievads .....	35
4.1.1	Auditorijas raksturojums .....	35
4.1.2	Lietojamība.....	35
4.1.3	Dokumenta nolūks.....	35
4.1.4	Saistītie dokumenti .....	35
4.1.5	Dokumenta lietošanas apraksts.....	35
4.1.6	Pieņemtie apzīmējumi un vienošanās.....	35
4.1.7	Problēmu ziņošana.....	35
4.2	Sistēmas lietošana .....	36
4.2.1	Ievads.....	36
4.2.2	Darbības sākums.....	36
4.2.3	Darbība ar sistēmu .....	39
4.2.4	Darbību beigšana .....	40
5.	Projekta organizācija .....	41
6.	Kvalitātes nodrošināšana.....	42
7.	Konfigurāciju pārvaldība.....	43
8.	Darbietilpības novērtējums. ....	44
9.	Izmantotās Tehnoloģijas .....	45
10.	Programmatūras izejas kods .....	46
11.	Pateicības.....	52
12.	Izmantotā literatūra un avoti.....	53

## Ievads

Cilvēkiem vienmēr ir vēlme izmēģināt ko jaunu, piemēram, jaunās tehnoloģijas un to iespējas. Viena no jaunajām cilvēku iekārotājam tehnoloģijām ir skārienjūtīgā ekrāna ierīces, tādas kā kompānijas „Apple” ražotās „iPhone” un „iPad”. Cilvēki, kuri iegādājās šos produktus, vēlās izbaudīt pieredzi, kuras rodas izmantojot šos produktus. Nav jāšaubās, ka cilvēkiem, kuriem pieder šīs ierīces izmanto arī personīgos datorus. Šajā darbā izstrādāju sistēmu, kura savieno labākās īpašības no lietotāja skārienjūtīgā ekrāna ierīces „iPhone” un lietotāja datoru, sniedzot izklaidējošu pieredzi.

„Tīmekļa vietnes manipulācija no skārienjūtīga ekrāna ierīces” ir sistēma, kura apvieno „iPhone” izmantojot tā skārienjūtīgo ekrānu un sistēmas lietotāja personīgā datora jaudu, attēlojot tīmekļa pārlūkprogrammā spēli „Pong”. Sistēmai tika izstrādāts serveris, kurš nodrošina komunikāciju starp „iPhone” un spēli „Pong”, tāpat arī serveris nodrošina tīmekļa pārlūkprogrammām piekļuvi, lai parādītu spēli „Pong”. Priekš „iPhone” tika izstrādāta lietotne „Controller”, kura nodrošina to ka „iPhone” tiek izmantots kā vadības pulsts spēlei „Pong”.

Pati spēle ir izstrādāta balstoties uz spēli gala tenis, tas ir „Pong” sastāv no reāllaikā kustīgām komponentēm – bumbas, un raketēm. Vienu raketi kontrolē sistēmas lietotājs izmantojot „iPhone” kā pulti, otra rakete, spēlētāja pretinieks ir spēles mākslīgais intelekts. Spēles mērķis ir neļaut bumbai aiziet garām savai raketei, un pasist bumbu garām pretinieka raketei.

Šī sistēma tika izstrādāta kā mans kvalifikācijas darbs. Kvalifikācijas darba laikā es ieguvu zināšanas un pieredzi šādas sistēmas izstrādē. Šādu sistēmu nolēmu izstrādāt ar mērķi pārbaudīt vai tas ir iespējams, kā padarīt šādas sistēmas iespējamību un noskaidrot vai šādu sistēmas ideju ir vērts turpināt paplašināt un iegūt kādu labumu to darot

Kvalifikācijas darba ietvaros tika izstrādāta atbilstošā dokumentācija, kura arī sevī ietver programmatūras prasību specifiku, programmas prasību aprakstu, lietotāja dokumentāciju, testēšanas aprakstu.

# 1. Programmatūras Prasību Specifikācija

## 1.1 Ievads

Dokumenta nolūks ir apkopot un aprakstīt sistēmas "Tīmekļa vietnes manipulācija no skārienjūtīgā ekrāna ierīces" prasības – gan funkcionālās, gan nefunkcionālās.

### 1.1.1 Darbības sfēra

Sistēma paredzēta lietotājiem, kuri vēlas savu „iPhonei” kombinācijā ar savu datoru, izmantot izklaidei. Sistēma izmanto iPhone” kā pulti, lai manipulētu tīmekļa pārlūkprogrammā ievietoto spēli „Pong”.

### 1.1.2 Definīcijas

Lietotājs – lietotājs kurš izmanto šo sistēmu.

### 1.1.3 Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standarta LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis” prasības

### 1.1.4 Dokumenta pārskats

Dokuments sastāv no 5 nodaļām: ievads, vispārējais apraksts, funkcionālās prasības, ārējā saskarne, nefunkcionālās prasības.

Ievadā aprakstīti dokumenta mērķi, nolūks, izmantotie jēdzieni, kas sastopami dokumentā.

Vispārējais apraksts tiek sniegts pārskats par funkcijām, kas sistēmai ir jāpilda, kā arī aprakstītas lietotāja raksturiezīmes. Šajā nodaļā ir aprakstīti ierobežojumi un atkarības, kas var ietekmēt sistēmas izstrādi.

Funkcionālo prasību nodaļā tiek aprakstītas sistēmā iekļautās funkcijas, to ieejas dati, šo datu apstrādes darbības, izejas dati, funkciju atkarības.

## 1.2 Vispārējais apraksts

Sistēma sastāv no trīs daļām. „iPhone” lietotnes „Controller”. „Controller” tiek izmantots, kā vadības pults spēlei „Pong”. 2D Spēle „Pong” ideoloģiski, seko galda tenisa principiem. Spēli „Pong” var spēlēt viens spēlētājs, kurš spēlē pret MI. Gan spēlētājs gan MI pārvieto pa spēles laukumu raketi ar kuru atsit bumbu. Spēle skaita punktus, tas ir kurš un cik reizes

spēlētājs ir iesitis bumbu pretinieka pusē – pieskāries laukuma galējai sānu malai ar bumbu. Spēlētājs savu raketi pārvieto izmantojot „iPhone”.

### **1.2.1 Produkta perspektīva**

Sistēma plānota kā vispasaules tīmeklī brīvi pieejams resurss, kas nodrošina tās lietotājam izklaides iespējas.

### **1.2.2 Biznesa prasības**

- a) Sistēmas lietotājam jānodrošina iespēja kontrolēt tīmeklī ievietotu spēli „Pong” ar „iPhone” lietotni „Controller”.

### **1.2.3 Produkta funkcijas**

Produkta funkcijas pēc būtības var iedalīt trīs kopās. Katrai funkcijai ir savs identifikators, kas sastāv no divām vērtībām kopas apzīmējums (Burts), un kārtas numurs tajā kopā.

1.„iPhone” lietotnes „Controller” sistēmas daļas funkcijas:

- a) Izveido lietotāja saskarni (C-1)
- b) Uztvert lietotāja komandas no lietotāja saskarnes. (C-2)
- c) Komandu pārsūtīšana uz sistēmas serveri. (C-3)

2.Sistēmas servera funkcijas:

- a) Uztvert un saglabāt atmiņā lietotāja sūtītās komandas no „Controller” lietotnes. (S-1)
- b) Sūtīt lietotāja komandas spēlei „Pong”. (S-2)
- c) Attēlot spēli tīmekļa pārlūkprogrammā. (S-3)

3.Spēles „Pong” funkcijas:

- a) Sazināties ar sistēmas serveri un iegūt lietotāja komandas. (P-1)
- b) Sākt spēli pēc lietotāja pieprasījuma. (P-2)
- c) Parādīt grafiski spēles komponentes spēlētājam (Spēlētāja un MI raketes, bumbu, spēles laukumu, punktu skaitu). (P-3)
- d) Mainīt spēlētāja raketes atrašanās vietu attiecīgi spēlētāja komandām. (P-4)
- e) Mainīt bumbas atrašanās vietu pēc spēlē notiekošā. (P-5)
- f) Mainīt MI raketes atrašanās vietu pēc spēlē notiekošā. (P-6)
- g) Rādīt spēlētāja un MI punktu skaitu (P-7)

#### 1.2.4 Lietotāja raksturiezīmes

Sistēmas lietotājam jābūt pieredzei lietojot „iPhone” un populārākās tīmekļa pārlūkprogrammas.

#### 1.2.5 Vispārējie ierobežojumi

Sistēmas funkcionēšanai un lietošanai nepieciešamais.

Lai lietu sistēmu, lietotājam jāpieder tālrunis „iPhone” ar uzstādītu šīs sistēmas lietotni „Controller”. Lietotājam uz sava datora jābūt uzstādītai modernam tīmekļa pārlūkam (Firefox, Chrome) ar Java atbalstu.

Sistēmas serverim jābūt Apache „TomCat”, vai arī kādam citam, kurš atbalsta Java EE 7.

Gan lietotāja datoram un tālrunim „iPhone” gan serverim jābūt ar piekļuvi tīmeklim.

#### 1.2.6 Pieņēmumi un atkarības

Sistēmas spēles daļa lietotājam ir pieejama tikai globālā tīmekļa vidē. Sistēma ir atkarīga no interneta pieejamības.

### 1.3 Funkcionālās prasības

#### 1.3.1 „iPhone” lietotnes „Controller” funkcijas.

Izvedot lietotāja saskarni

ID	C-1
Mērķis	Šī funkcija automātiski iedarbinās, kad lietotne „Controller” tiek iedarbināta. Funkcija izveido lietotāja saskarni – pogas ar kurām lietotājs vadīs spēli „Pong”. Saskaņā ir no trim pogām ar nosaukumiem „UP” pogas krāsa Zaļa, „STOP” - Dzeltēna, „DOWN” - Sarkana. Poga ir „iPhone” skārienjūtīgajā ekrānā lauks, kurš uztver pirkstu pieskārienus tajā.
Ievaddati	
Apstrāde	
Izvaddati	Uz „iPhone” ekrāna parāda trīs pogas ar nosaukumiem uz tām „UP”, „STOP”, „DOWN”.

Uztvert lietotāja komandas no lietotāja saskarnes

ID	C-2
Mērķis	Šī funkcija automātiski iedarbinās, kad lietotne „Controller” tiek iedarbināta. Funkcija uztver lietotāja veiktās darbības- komandas-, lietotāja saskarnē.
Ievaddati	Sistēmas iekšienē, atkarībā no lietotāja saskarnē nospiestas pogas (UP, STOP, DOWN) funkcijai tiek padots attiecīgās pogas nosaukums.
Apstrāde	Saglabā nospiestās pogas nosaukumu, lietotnes sistēmā.
Izvaddati	

Komandu pārsūtīšana uz sistēmas serveri.

ID	C-3
Mērķis	Pārsūtīt lietotāja komandas datus uz sistēmas serveri. Funkcija ieslēdzas tad kad ir nospiesta kāda poga uz „iPhone” ekrāna.
Ievaddati	Lietotnes iekšienē esošais, nospiestās pogas nosaukums.
Apstrāde	<p>Izveido savienojumu ar sistēmas serveri, un nosūta lietotāja komandu kā parametru serverim - sūta URL. Pirms sūtīšanas notiek pārbaude vai nospiestās pogas nosaukums atbilst (UP, STOP, DOWN) .</p> <p>Pārsūtītais parametrs atbilst šādām vērtībām:</p> <ul style="list-style-type: none"> <li>• Ja nospiestās pogas nosaukums ir UP parametrs tad ir 1.</li> <li>• Nosaukums STOP parametrs 0.</li> <li>• Nosaukums DOWN parametrs -1.</li> </ul> <p>Parametra nosaukums ir „direction” (Tulkojumā - Virziens). Lietotājs klikšķina pogas kuru nosaukums atbilst kustības virzieniem. Tas ir, ja parametrs ir vienāds ar 1 tad tas tiek pieņemts kā virzīties uz augšu, 0 ir apstāties, un -1 ir virzīties uz leju..</p>
Izvaddati	

### 1.3.2 Sistēmas servera funkcijas:

Uztvert un saglabāt atmiņā lietotāja sūtītās komandas no „Controller” lietotnes

ID	S-1
Mērķis	Uztver lietotāja komandas parametru, tas ir gaida attiecīgo URL , kuru izsūta no „Controller” lietotnes un to saglabā.
Ievaddati	Saņemtais URL.
Apstrāde	No saņemtā URL atdala parametru kurš apzīmē komandu un saglabā to uz servera kā mainīgo. Pirmo reizi palaižot sistēmu mainīgais tiek saglabāts ar noklusēto vērtību 0.
Izvaddati	

Sūtīt lietotāja komandas spēlei „Pong”.

ID	S-2
Mērķis	Sistēmā esošo parametru pārsūtīt pēc pieprasījuma spēlei „Pong”, Tas ir pārsūta ja vērtība ir -1 vai 0 vai +1.
Ievaddati	Servera sistēmā esošais mainīgais – parametrs kurš atbilst lietotāja komandai – kustības virzienam.
Apstrāde	Pēc HTTP pieprasījuma, atbild pieprasītājam – lietotnei „Pong” pārsūtot mainīgo kurš glabā lietotāja komandu.
Izvaddati	

Attēlot spēli tīmekļa pārlūkprogrammā

ID	S-3
Mērķis	Pēc lietotāja sūtīta tīmekļa pārlūkprogrammas pieprasījuma, attēlot tīmekļa pārlūkprogrammā spēli „Pong”
Ievaddati	Lietotāja tīmekļa pārlūkprogrammas URL pieprasījums serverim.
Apstrāde	
Izvaddati	Lietotāja tīmekļa pārlūkprogrammā, logs ar spēli „Pong”

### 1.3.3 Spēles „Pong” funkcijas

Sazināties ar sistēmas serveri un iegūt lietotāja komandas.

ID	P-1
Mērķis	Spēlē lietotājs kontrolē savu raketi ar „iPhone”. Funkcija sazinās ar serveri un uzzina, kādā virzienā spēlētājs vēlas kustēties.
Ievaddati	Spēlētāja pieprasītais kustības virziens.
Apstrāde	Spēles iekšienē nomaina spēlētāja raketes virziena mainīgo uz spēlētāja pieprasīto.
Izvaddati	

Sākt spēli pēc lietotāja pieprasījuma

ID	P-2
Mērķis	Uzsāk spēli kad lietotājs to pieprasa. Spēle „Pong” norisinās lietotāja datorā, palaižot spēli, tā gaida kad lietotājs nospiedīs taustiņu „Space”, lai sāktu spēli. Spēle „Pong” izmanto Java iebūvēto taustiņu klausīšanos.
Ievaddati	Spēlētāja nospiests taustiņš.
Apstrāde	Pārbauda kurš klaviatūras taustiņš tika nospiests, ja tas ir „Space”, jeb taustiņš 32 (taustiņa Space numurs), tad sāk spēli.
Izvaddati	

Parādīt grafiski spēles komponentes spēlētājam (Spēlētāja un MI raketes, bumbu, spēles laukumu, punktu skaitu).

ID	P-3
Mērķis	Šī funkcija izmanto Java grafisko vidi, lai attēlotu spēlē notiekošo. Pirms spēle ir sākusies uz ekrāna jāuzrāda spēles instrukcijas „ Use your „iPhone” to controll the paddle, press Space to start.”. Spēles laukumam krāsa - Zaļa,
Ievaddati	Iekšējie mainīgie kuri apzīmē spēles komponentes. Tie var dinamiski mainīties.
Apstrāde	Izmantojot Java grafisko vidi un komponentu mainīgos, iegūst datus kurus attēlot lietotājam.
Izvaddati	Spēles grafiska reprezentācija.

Attēlot spēlētāja raketes atrašanās vietu attiecīgi spēlētāja komandām.

ID	P-4
Mērķis	Kad spēlētāja kustības komanda ir iegūta no servera un saglabāta mainīgajā. Tad funkcijai spēlētāja atsūtītais kustības virziens ir jāpiešķir spēlē esošajai spēlētāja raketei. Spēlētāja raketes krāsa –Sarkana.
Ievaddati	Sistēmas iekšējais spēlētāja virziena mainīgais.
Apstrāde	Attiecīgi pēc ievaddatiem maina spēlētāja raketes virzienu.
Izvaddati	

Mainīt bumbas atrašanās vietu pēc spēlē notiekošā.

ID	P-5
Mērķis	<p>Tā kā „Pong” norisinās reāllaikā spēles komponentēm jāatjauninās atbilstoši. Komponentei bumba, nav jāiziet no spēles laukuma, sasniedzot laukuma augšējo un apakšējo malu, bumbai no tām ir jāatsitas.</p> <p>Sasniedzot labo vai kreiso malu – spēlētāju aizsargātās malas bumbas atrašanās vieta tiek ielikta atpakaļ sākuma vietā – centrā.</p> <p>Tāpat arī bumbai ir jāatsitas no spēlētāja raketēm. Bumba pārbaudot savas koordinātas var izsekot tam vai ir sasniegusi kāda spēlētāja aizsargāto malu tādejādi var noskaidrot kurš spēlētājs iegūs punktus.</p> <p>Sākot spēli bumbai jāsāk kustēties nejaušā virzienā. Atsišanās nozīmē ja bumba savā ceļā saduras ar kādu no norādītajām komponentēm, bumba maina savu kustības virzienu uz pretējo. Bumbas krāsa –Zila.</p>
Ievaddati	<p>Spēles iekšējie mainīgie - Bumbas sākuma koordinātas, laukuma izmēri, spēlētāju rakešu atrašanās vietas, to izmērs.</p>
Apstrāde	<p>Visi apstrādes dati nokļūst grafiskās attēlošanas funkcijā. Bumba sākuma vērtības tiek ievadītas tādas kādas padotas ievaddatos. Laukuma izmērus izmanto metodē kas neļauj bumbai iziet no laukuma.</p> <p>Spēlētāju atrašanās vietas un to izmērus izmanto metodē kura liek bumbai atsisties no spēlētāju raketēm.</p> <p>Ja bumba sasniegusi kāda spēlētāja malu tad tiek izsaukta metode kura saglabā punktus spēlētājiem</p>
Izvaddati	

Mainīt MI raketes atrašanās vietu pēc spēlē notiekošā

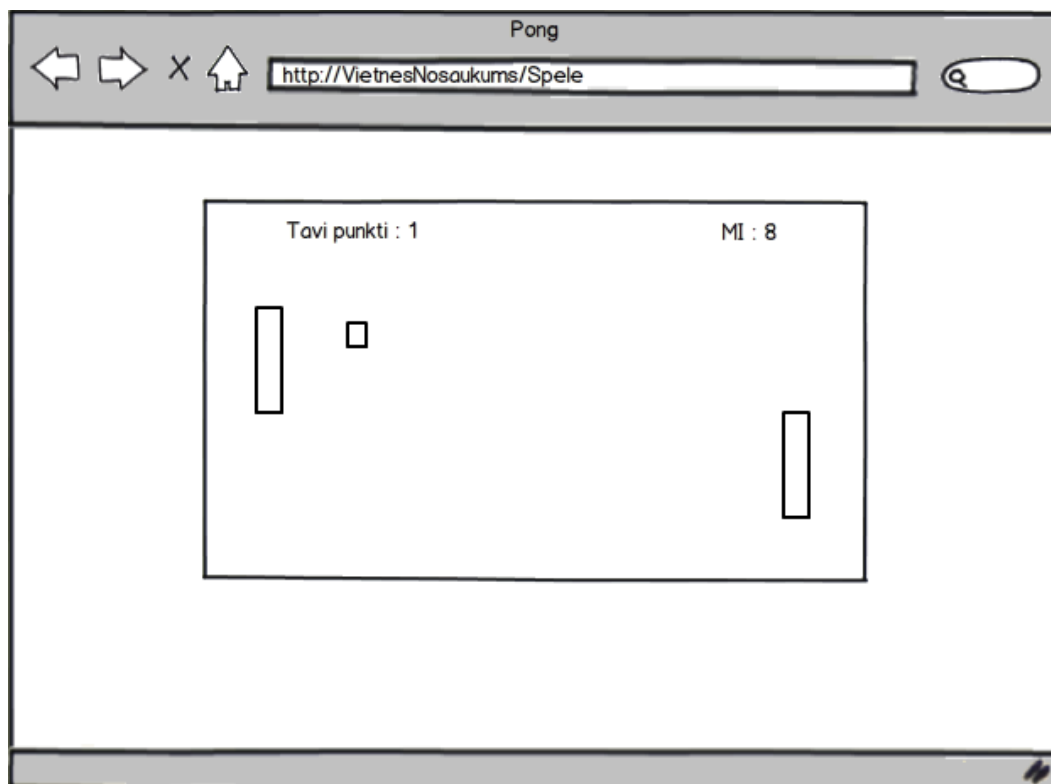
ID	P-6
Mērķis	MI raketē jāķer bumba, tas ir jāatsit tā pirms pieskaras tai spēles laukuma pusei kuru MI aizsargā. MI rakete nedrīkst iziet no laukuma. Krāsa – Melna.
Ievaddati	MI raketes sākuma atrašanās vieta, bumbas atrašanās vieta, laukuma izmērs.
Apstrāde	Attiecīgi no bumbas atrašanās vietas maina MI raketes virzienu. MI raketes atrašanās vietu padod metodei kura neļauj iziet tai no laukuma robežām
Izvaddati	

Rādīt spēlētāja un MI punktu skaitu.

ID	P-7
Mērķis	Parādīt lietotājam, viņa un MI iegūtos punktus. To var izdarīt , kad bumba ir sasniegusi kādu malu noskaidrot kurai malai bumba pieskārs, tātad kāds spēlētājs vai MI iegūst punktu. Krāsa – Sarkana.
Ievaddati	Iekšējie spēles „Pong” mainīgie, kuri glabā spēlētāja un MI punktu skaitu.
Apstrāde	Mainīgos pārvērš par grafiski parādāmām komponentēm.
Izvaddati	

## 1.4 Arējās saskarnes prasības

Tā kā, spēle „Pong” jābūt pieejamai izmantojot tīmekļa pārlūkprogrammu, spēle tajā jāattēlo kā rādīts attēlā 1. Pašas Java sīklīetotnes izmēriem jābūt 800 px platumā un 400 px augstumā.



[Attēls 1]

### 1.4.1 Sakaru saskarne

„iPhone” lietotne „Controller” saziņai ar sistēmas serveri izmanto HTTP protokolu. HTTP protokolu arī izmanto Java sīklīetotne, spēle „Pong” saziņai ar serveri.

## 1.5 Nefunkcionālās prasības

### 1.5.1 Ātrdarbība

Sistēmai jānodrošina viena lietotāja darbības ar to. Sistēmas ātrdarbība ir atkarīga no „iPhone” lietotāja datora, servera ātrdarbības tāpat arī no to interneta pieslēguma. Šos lielumus sistēma nevar ietekmēt.

Lietotāja komandām, kuras tiek sūtītas no „iPhone” spēlē jāveic ne lēnāk kā 0,5 sekundēs – tā lai cilvēks nobīdi nevar ievērot.

### 1.5.2 Datu drošība

Sistēmā neglabā nekādus jūtīgus datus.

## **2. Programmatūras projektējuma apraksts**

### **2.1 Ievads**

Šajā nodaļā tiek aprakstīts sistēmas „Tīmekļa vietnes manipulācija no skārienjūtīgā ekrāna ierīces”, mērķauditorija, tiek aprakstīta saistība ar citiem dokumentiem, aprakstīts dokumenta nolūks.

### **2.2 Nolūks**

Šī dokumenta nolūks ir aprakstīt „Tīmekļa vietnes manipulācija no skārienjūtīga ekrāna ierīces” sistēmas uzbūvi. Dokuments paredzēts sistēmas izstrādātājiem. Dokumenta nolūks ir palīdzēt sistēmas plānošanā, analīzē un implementēšanā, kā paredzēts pēc šīs sistēmas programmatūras prasību specifikācijā

### **2.3 Darbības sfēra**

Šī sistēma ir paredzēta izklaidei. Sistēmu lieto izmantojot „iPhone” kā pulti, tīmekļa vietnē ievietotai spēlei „Pong”.

### **2.4 Izmantotie jēdzieni un saīsinājumi**

ID – Unikāls identifikators .

MI- Mākslīgais Intelekts

PPS - Programmatūras prasību specifikācija.

PPA - Programmatūras projektējuma apraksts.

LVS – Latvijas valsts standarts.

Java – Programmēšanas valoda.

Objective-C – Programmēšanas valoda.

### **2.5 Saistība ar citiem dokumentiem**

LVS 72:1996 "Ieteicamā prakse programmatūras projektējuma aprakstīšanai"

Šīs sistēmas izstrādāto PPS.

## 2.6 Dekompozīcijas apraksts

PPS sistēma tika sadalīta trīs kopās, kuras kopā veido vienu sistēmu:

Sistēma

- „iPhone” lietotne „Controller”.
- Sistēmas serveris.
- Spēle „Pong”.

Sistēmu var sadalīt šīm kopām atbilstošos moduļos. Katru moduli nodala pēc funkcijām kuras tam jāveic. Funkciju mērķi atrodas PPS. Katru moduli var sadalīt apakš moduļos, lai smalkāk aprakstītu un palīdzētu izveidot, izprast sistēmas procesus, kuriem jāizpilda PPS pieprasītās funkcijas.

### 2.6.1 Moduļu dekompozīcija

Sistēmas galveno moduļu apraksts.

„iPhone” lietotnes „Controller” modulis.

Izpildāmās funkcijas.	C-1, C-2, C-3
Nosaukums.	Controller
Apraksts.	Šis modulis, atbild par: <ul style="list-style-type: none"><li>• „iPhone” lietotnes lietotājas saskarni.</li><li>• Informācijas saglabāšanu par lietotāja darbībām ar saskarni.</li><li>• Lietotāja darbību informācijas pārsūtīšanu sistēmas serverim.</li></ul>

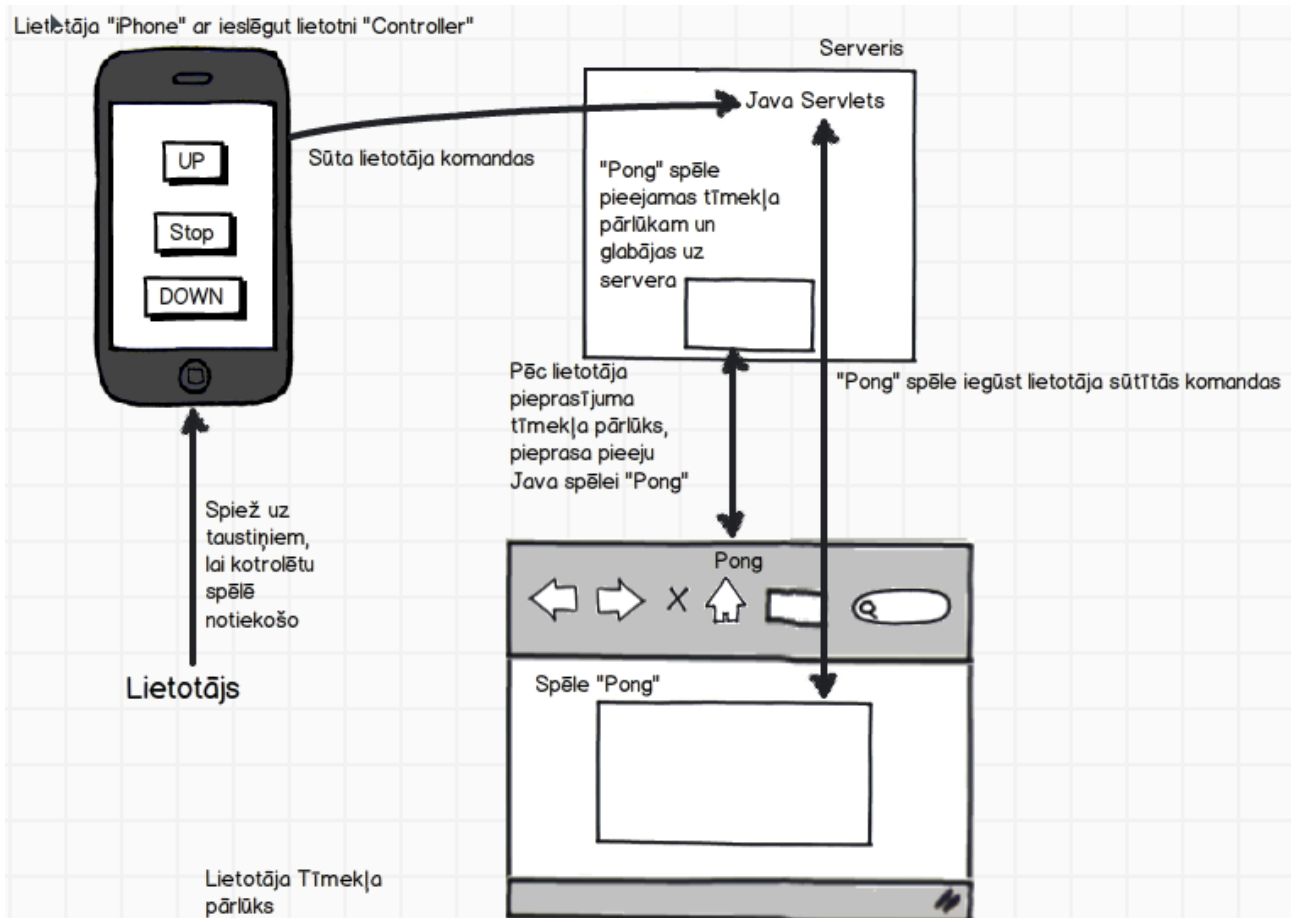
Sistēmas servera modulis.

Izpildāmās funkcijas.	S-1, S-2, S-3
Nosaukums.	Serveris
Apraksts.	<p>Sistēmas serveris, daļu kura nodrošina saziņu starp „Controller” un „Pong”, nodrošina ar Java serversīkietotni (Java „Servlet”).</p> <p>Serverim arī jānodrošina lietotājam iespēja savā tīmekļa pārlūkprogrammā spēlēt spēli „Pong”, tas nozīme, ka uz servera jāatrodas tīmekļa vietnei.</p> <p>Šis modulis, atbild par:</p> <ul style="list-style-type: none"> <li>• Lietotāja komandu saņemšanu no „iPhone” lietotnes „Controller”.</li> <li>• Saņemto datu pārbaudi.</li> <li>• Komandu pārsūtīšanu spēlei „Pong”.</li> <li>• Iespēju spēli „Pong” parādīt lietotāja tīmekļa pārlūkprogrammā.</li> </ul>

Spēles „Pong” modulis.

Izpildāmās funkcijas.	P-1, P-2, P-3, P-4, P-5,P-6,P-7
Nosaukums.	Pong
Apraksts.	<p>Spēle „Pong” norisinās reāllaikā.</p> <p>Šis modulis, atbild par:</p> <ul style="list-style-type: none"> <li>• Grafisku spēles reprezentāciju lietotājam.</li> <li>• Spēlētāja komandu iegūšanu no servera.</li> <li>• Spēlētāja komandu izpildi.</li> <li>• Spēles atbilstību PPS norādītajām prasībām.</li> </ul>

Šādu iegūst tad, sistēmas konceptuālo uzbūvi.



[Attēls 2]

Galvenās funkcijas apraksts - Lietotājs operē gan ar „iPhone” gan ar tīmekļa pārlūkprogrammu. Servera java „Servlets” –sīklietotnes- nodrošina komunikāciju starp moduļiem. Spēle tiek attēlota tīmekļa pārlūkprogrammā.

Sistēmas moduļu, apakš moduļu apraksts.

Apakš moduļus apzīmē ar ID, funkciju kuru tas izpilda, aprakstu. ID tiek veidots no virs moduļa nosaukuma pirmā burtā pievienojot tam a (a-apakš modulis) un apakš moduļa kārtas numura.

„Controller” apakš moduļi.

ID.	aC1
Nolūks/funkcija.	Veido lietotāja saskarni. Saskarni veido „iPhone” paredzēto programmēšanas rīku „Inerface Builder”.
Atbilstošais fails	ControllerAppViewController.xib

ID.	aC2
Nolūks/funkcija.	Pēc lietotāja veiktās darbības (Klikšķa uz pogas), saglabā pogas nosaukumu atmiņā.
Faili	ControllerAppViewController.h, ControllerAppViewController.m

ID.	aC3
Nolūks/funkcija.	Izveido un pieprasa serverim URL pieprasījumu. URL tiek pievienots parametrs.
Faili	ControllerAppViewController.h, ControllerAppViewController.m

Servera apakš moduļi.

ID.	aS1
Nolūks/funkcija.	Lai saņemtu datus no „Controller” tiek izmantota Java serversīkietotne. Lietotāja pārsūtītās komandas tiek uztvertas kā URL, kurš pienāk pie serversīkietotnes. Saņemtie dati tiek pārbaudīti vai tie atbilst, PPS norādītajiem nosacījumiem ,tad tiek saglabāti.
Faili (Java serversīkietotne)	Listener.java

ID.	aS2
Nolūks/funkcija.	Lai saņemtu datus un sūtītu tos „Pong” tiek izmantota Java serversīkļietotne. Pēc sīkļietotnes „Pong” pieprasījuma, tai sūta uz servera glabājošās lietotāja pieprasītās komandas. Pārsūta objektu ar tām.
Fails (Java serversīkļietotne)	Listener.java

ID.	aS3
Nolūks/funkcija.	Pēc lietotāja tīmekļa pārlūkprogrammas pieprasījuma attēlo spēli „Pong”.
Tīmekļa vietnes fails.	Index.jsp

„Pong” apakš moduļi.

ID.	aP1
Nolūks/funkcija.	Attēlo grafiski spēli „Pong”. Izmantojot Java grafisko vidi attēlo visas spēles komponentes.
Fails	Pong.java

ID.	aP2
Nolūks/funkcija.	Spēles komponente bumba. Pārvieto bumbu pēc PPS norādēm.
Fails	Ball.java

ID.	aP3
Nolūks/funkcija.	Spēles komponente MI rakete. Pārvieto MI raketi pēc PPS norādēm.
Fails	AIPaddle.java

ID.	aP4
Nolūks/funkcija.	Spēles komponente punktu skaits. Parāda spēlētāja un MI punktus.
Fails	Pong.java

ID.	aP5
Nolūks/funkcija.	Spēles komponente lietotāja rakete. Pārvieto spēlētāja raketi pēc PPS norādēm. (Pārvieto atkarīgi no dotās komandas.)
Fails	PlayerPaddle.java

ID.	aP6
Nolūks/funkcija.	Iegūts lietotāja komandas no Servera. Izveido savienojumu ar serveri un pieprasa komandu saturošo objektu. Kad tas saņemts saglabā lietotāja komandu atmiņā.
Fails	Pong.java

ID.	aP7
Nolūks/funkcija.	Iegūst no bumbas vērtības spēlētāja un MI punktus.
Fails	Pong.java

## 2.7 Moduļu atkarības apraksts.

Dekompozīcijas rezultātā veidojošos moduļu atkarību tabula. Tabula paredzēta lai parādītu kuri moduļi no kuriem ir atkarīgi. Tāpat arī parāda ja kādam modulim tiek mainīta funkcionalitāte, kuri moduļi dēļ tā arī būs jāmaina.

Modulis:	Izmanto moduļus.	Tiek izmantos moduļos.	Kuru funkciju izpildē piedalās.	Īss Apraksts.
„Controller”	aC1, aC2, aC3		C-1,C-2,C-3	„iPhone” lietotne.
aC1		Controller, aC2	C-1	Veido lietotāja saskarni.
aC2	aC1	Controller, aC3	C-2	Glabā veiktās darbības.
aC3	aC2,aS1	Controller ,Serveris, aS1	C-3	Veiktās darbības pārsūta serverim
Serveris	aS1, aS2, aS3		S-1,S-2,S-3	
aS1	aC3	Serveris , aC3	S-1	Uztver darbību Sūtījumu.
aS2	aP6	Serveris , aP6	S-2	Pārsūta veiktās darbības spēlei
aS3		Serveris	S-3	Attēlo spēli.
Pong	aP1,aP2,aP3,aP4,aP5,aP6		P-1,P-2,P-3,P-4,P-5,P-6,P-7	Spēle.
aP1	aP2,aP3,aP4,aP5,aP6	Pong	P-3	Attēlo spēli grafiski.
aP2	aP1, aP7, aP4	Pong , aP1,aP2	P-5	Komponente bumba.
aP3		Pong , aP1	P-6	MI rakete.
aP4	aP7	Pong , aP1	P-7	Punktu skaits.
aP5	aP6	Pong , aP1	P-4	Spēlētāja rakete.
aP6	aS2	Pong , aP1, aS2	P-1	Iegūst lietotāja komandas no servera.
aP7	aP2,aP4	Pong , aP1	P-7	Iegūst punktus no bumbas.

## 2.8 Moduļu detalizēts apraksts

Nodaļas mērķis smalkāk apskatīt dekompozīcijā iegūto moduļu darbības, PPS pieprasīto funkciju izpildei. Nodaļa izstrādāta, kā norādījumi, lai izstrādātu katru moduli.

Detalizēto aprakstu veido katram modulim. Katram modulim būs tāds apraksts, lai varētu pēc iespējas vieglāk ieviest PPS aprakstītās prasības.

Apraksts sastāvs:

- Veiktās funkcijas.
- Nepieciešamie mainīgie.
- Metožu prototipi, moduļa funkcijas izpildē.
- Apraksts un ieteikumi kādas tehnoloģijas izmantot labākai funkciju izpildei.

Metodes un mainīgie norādīti kā sistēmas funkciju izpildes prototipi. Tas ir norāda to iespējamus datu tipus un atbilstošus nosaukumus

### **1. Modulis: Controller**

Veiktās funkcijas – C-1, C-2, C-3

Apraksts - Nodrošina „iPhone” lietotnes „Controller” funkcionalitāti. Sastāv no apakšmoduļiem.. Modulis tiek veidots programmēšanas valodā Objective-C.

### **2. Apakš Modulis: aCI**

Veiktās funkcijas – C-1

Nepieciešamie mainīgie – Pogų nosaukumi (PPS prasības, UP, STOP, DOWN)

Metožu prototipi -

Apraksts - Izmanto „iPhone” programmatūras izstrādes rīku „Interface builder” tas veido .xib failu saturu. Rīks automātiski katrai saskarnes komponentei (Pogai) piešķir ID.

### **3. Apakš Modulis: aC2**

Veiktās funkcijas – C-2

Nepieciešamie mainīgie – pogas ID pēc kura nosaka, kura poga tika nospiesta. NSString nameOfButton, glabā pogas nosaukumu.

Metožu prototipi - (IBAction)Click:(id)sender.

Apraksts – Metodē saņem pogas ID, noskaidro pēc tā pogas nosaukumu, tad to saglabā mainīgajā nameOfButton.

### **4. Apakš Modulis: aC3**

Veiktās funkcijas - C-3

Nepieciešamie mainīgie – NSURL \*url

Metožu prototipi - Pārbauda ar vai pogas atbilst PPS prasībām, ja jā tad izveido Savienojumu ar serveri un pārsūta tam parametru.

Apraksts - Parametra pārsūtīšana notiek izmantojot URL. Pārsūtītais parametrs ir atkarīgs no mainīgā nameOfButton. Ja tas ir UP tad parametrs ir 1, ja tas ir STOP parametrs ir 0 , ja tas ir DOWN tad parametrs ir -1.URL veido savienojumu ar sistēmas servera serversīkietotni.

Parametra nosaukums ir „direction”

### **5. Modulis: Serveris**

Veiktās funkcijas – S-1, S-2, S-3

Nepieciešamie mainīgie – Int dir, glabā lietotāja pieprasīto kustības virzienu.

Apraksts – Apakš moduļi aS1 un aS2 ir Java serversīkietotne „Listener”. aS3 ir tīmekļa vietne.

## **6. Apakš Modulis: aS1**

Veiktās funkcijas - S-1

Nepieciešamie mainīgie – String dirFromURL –glabā saņemtā parametra vērtību.

Metožu prototipi - doGet (), setDirection();

Apraksts – Gaida URL pieprasījumu doGet (). Kad tas notiek No URL atdala parametru ar nosaukumu „direction”. Tā vērtību saglabā dirFromURL. Metodē setDirection(), dod saņemtā parametra vērtību un glabā mainīgajā dir.

## **7. Apakš Modulis: aS2**

Veiktās funkcijas – S-2

Nepieciešamie mainīgie – ResponseObject response;

Metožu prototipi - doPost(), respondToApplet();

Apraksts – Gaida savienojumu ar doPost() metodi. Kad tas notiek izveido objektu kurā ievieto mainīgo dir. Un atbild sūtot šo objektu metodes pieprasītājam. Objektam jārealizē Java saskarne java.io.Serializable, lai objektu būtu iespējams pārsūtīt izmantojot globālo tīmekli.

## **8. Apakš Modulis: aS3**

Veiktās funkcijas – S-3. Pēc lietotāja pieprasījuma rādīt spēli „Pong”

Nepieciešamie mainīgie –

Metožu prototipi -

Apraksts - Glabā spēli .jar formātā ievietotu index.html. Lai Java spēli „Pong” parādītu lietotājam, tās failus ir nepieciešams saarhivēt .jar failā. aS3 izveido sava veida tīmekļa vietni, kuru parāda pievienojoties serverim ar tā adresi. .jar fails tiek ievietots <APPLET> tagos index.html failā. Spēle izmanto darbības, kuras tiek lietotāja sistēmā bloķētas (Java „Applet” drošība). Tādēļ jāizveido drošības sertifikāts, kurš brīdina lietotāju par „Applet” darbībām. Drošības sertifikātu pievieno .jar failam, izmantojot, Java „keyChain”.

### **9. Modulis: Pong**

Veiktās funkcijas – P-1 līdz P-7.

Nepieciešamie mainīgie – Int mainīgie - Laukuma izmēri, Bumbas sākuma koordinātas. Lietotāja raketes sākuma koordinātas. MI Raketes sākuma koordinātas. Spēlētāju punktu rādīšanas vietu koordinātas, punktu skaitu saturošie mainīgie - int scorePlayer, scoreAI. Visu komponentu klases. int playerDir – glabā saņemtās lietotāja kustības virziena komandas.

Metozu prototipi - Norāda visu komponentu sākuma vērtības (setMetodes). run().

KeyListener() setPaddleYDir(playerDir)

Apraksts – Reāllaikā spēle „Pong”. To var nodrošināt ar Java saskarni „Runnable” tās metodi run(). Tiek izmantoti java „Threads”, kuri nodrošina spēles kustības reāllaikā. PPS prasībās norādīts, ka spēli sāk pēc lietotāja nospiesta taustiņa „Space”, to nodrošina Java saskarne „KeyListener”.

### **10. Apakš Modulis: aP1**

Veiktās funkcijas – P- 3

Nepieciešamie mainīgie – Java „Graphics” nepieciešamie mainīgie String instructions

Metozu prototipi - Grafiskās attēlošanas funkcijas paint() draw() update().

Apraksts – Grafiski attēlo spēli, to dara attēlojot, katru spēles komponenti (Spēlētāju raketes, bumbu, laukumu, punktu skaitu). Kad spēle inicializēta, pēc PPS prasībām uz spēles laukuma jāparāda spēles instrukcijas, kuras parāda pirms sākas spēle un pazūd tikko kā sākas spēle.

### **11. Apakš Modulis: aP2**

Veiktās funkcijas – P- 5

Nepieciešamie mainīgie – Noklusētās sākuma koordinātas Int x, y.Kustības virzieni.Int dirX, dirY.Bumbas izmērs.Int ballSize.Komponenšu izmēru un atrašanās vietu glabāšani..Int locationX, locationY, SizeHeight,/SizeWidht Int scorePlayer, scoreAI

Metožu prototipi - run() moveBall() randomDirection() restetBall() get metodes komponentes mainīgo mainīšanai setPayer() setAI () setFieldSize() whoScores(). draw()

Apraksts – Spēles komponente bumba Bumbai pārvietošanas notiek pēc PPS norādītā principa reāllaikā. Lai bumba atsistos no abām raketēm un spēles laukuma malām, bumba ir jānodrošina ar to atrašanās vietām un izmēriem. PPS prasību realizācijai punktu iegūšanai. Metodi izsauc grafiskās attēlošanas modulis. Metode glabā komponentes PPS norādītās grafiskās īpašības (krāsu, izmēru)

### **12. Apakš Modulis: aP3**

Veiktās funkcijas - P - 6

Nepieciešamie mainīgie – Sākuma koordinātas Int x, y.Kustības virzieniInt dirYMI raketes izmērsInt AIPaddleSize

Metožu prototipi - chaseBall() moveBall() setYdir() run() draw()

Apraksts – Spēles komponente MI rakete. Pārvietojās pēc PPS norādēm. Nodrošina MI raketes pārvietošanos reāllaikā Metodi izsauc grafiskās attēlošanas modulis. Metode glabā komponentes PPS norādītās grafiskās īpašības (krāsu, izmēru)

### **13. Apakš Modulis: aP4**

Veiktās funkcijas – P -7

Nepieciešamie mainīgie – Atrašanās koordinātas Int x, y.

Metožu prototipi - drawString();

Apraksts – Spēles komponente punktu skaits, tiek attēlots pēc PPS norādēm. Java grafiskās attēlošanas metode. Attēlojamās vērtības iegūs no Pong

#### **14. Apakš Modulis: aP5**

Veiktās funkcijas – P- 4

Nepieciešamie mainīgie – Sākuma koordinātas Int x, y. Kustības virzieni Int dirY spēlētāja raketes izmērs Int playPaddleSize

Metožu prototipi - setPaddleYDir() movePaddle() draw() run()

Apraksts – Spēles komponente lietotāja rakete. Pēc PPS norādītajām prasībām iegūst kustības virzienu no Pong. Raketei pārvietojās reāllaikā.

#### **15. Apakš Modulis: aP6**

Veiktās funkcijas – P -1

Nepieciešamie mainīgie – Objekts kuram piešķir, no servera iegūtā objekta vērtības. URL ar servera sīklietotnes adresi.

Metožu prototipi - getDirectionUsingHttp() setPlayDir()

Apraksts – Iegūts lietotāja komandas no Servera sīklietotnes, kā objektu. Servera sīklietotne spēli „Pong” atbild un gaida pieprasījumu ar POST metodi Glabā Servera PPS norādīto sīklietotnes tīmekļa adresi. Saglabā sistēmā, no servera iegūto lietotāja pieprasīto kustības virzienu.

#### **16. Apakš Modulis: aP7**

Veiktās funkcijas – P -7

Nepieciešamie mainīgie –

Metožu prototipi - checkScore()

Apraksts – Iegūst punktus spēlētājam un MI. Bumba glabā cik reizes tā ir sasniegusi spēles malas, kuras sargā spēlētāju raketes, šī metode iegūst tās vērtības un uzstāda tās kā punktus mainīgajos Pong

### 3. Testēšanas dokumentācija

#### 3.1 Ievads

Testēšana notika paralēli sistēmas izstrādei, tādējādi tas ļāva novērst lielāko daļu kļūdu jau izstrādes procesā. Sistēmas moduļi („iPhone” lietotne „Controller”, Serveris, spēle „Pong”) tika testēti to izstrādes laikā, vai tie atbilst PPS norādītajām prasībām.

#### 3.2 Manuālie testi

Ņemot vērā to ka sistēma sastāv no „iPhone” lietotnes „Controller”, Sistēmas servera, spēles „Pong” testēšana notika pa moduļiem:

- Controller
- Serveris
- Pong

Controller moduļa testēšana. Modulī veiktās darbības vizuāli tiek attēlotas tīmekļa pārlūkprogrammā, kura ir atvērusi tīmekļa vietni, kura attēlo spēli „Pong”

Testa ID	Veiktā Darbība	Sagaidāmais rezultāts	Iespējamās kļūdas	Vai testa izpildās
tC1	„iPhone” telefonā tiek ieslēgta lietotne „Controller”.	„iPhone” ekrānā parādās lietotāja saskarne.	„iPhone” – lietotne „Controller” sabojāta, „iPhone” sistēma to nevar atvērt. „iPhone” - beidzās baterija, darbības tika pārtrauktas.	Jā

Testa ID	Veiktā Darbība	Sagaidāmais rezultāts	Iespējamās kļūdas	Vai testa izpildās
tC2	Tiek klikšķināts uz lietotāja saskarnes pogām.	Ar tīmekļa pārlūkprogrammu ir ieiets sistēmas tīmekļa vietnē un spēle ir palaista. Spēlētāja rakete spēle, sāk kustēties ,atbilstošam pogas nosaukuma kustības virzienam.	Nav no sistēmas atkarīgas	Jā
		Spēle nav palaista. Nenotiek nekādas izmaiņas.		

Server moduļa testēšana. Vienīgā darbība kuru vizuāli var pārbaudīt it spēlei „Pong” pieeja izmantojot tīmekļa pārlūkprogrammu.

Testa ID	Veiktā Darbība	Sagaidāmais rezultāts	Iespējamās kļūdas	Vai testa izpildās
tS1	Tīmekļa pārlūkprogrammā veic pieprasījumu uz Servera atrodošās tīmekļa vietnes adresi.	Tīmekļa pārlūkprogrammā parādās spēle „Pong”	Ja nav pieejams internets vai serveris ir izslēgts, nepieejams ,tad tīmekļa pārlūkprogrammā parādās tās noklusētais logs ar paziņojumu ka resurss nav pieejams.	Jā

## Pong moduļa testēšana

Testa ID	Veiktā Darbība	Sagaidāmais rezultāts	Iespējamās kļūdas	Vai testa izpildās
tP1	Spēle tiek iedarbināta, bet nekādas darbības netiek tālāk veiktas.	Spēles logā ir redzamas visas spēles komponentes arī spēles instrukcijas.		Jā

Testa ID	Veiktā Darbība	Sagaidāmais rezultāts	Iespējamās kļūdas	Vai testa izpildās
tP2	Kad spēle ir iedarbināta, tiek nospiests taustiņš „Space”	Reāllaikā sāk mainīties spēles logā redzamais- sākas spēle. Pazūd spēles instrukcijas.		Jā

## **4. Lietotāja dokumentācija**

### **4.1 Ievads**

Šajā dokumentācija tiek aprakstīts kā lietot sistēmu, lai izmantotu tās funkcionalitāti.

#### **4.1.1 Auditorijas raksturojums**

Sistēmas lietotājiem ir prasmes „iPhone” lietošanā, tīmekļa pārlūkprogrammas lietošanā.

#### **4.1.2 Lietojamība**

Lai lietotu sistēmu jābūt uzstādītai modernai pārlūkprogrammai jaunākā versija – Mozila Firefox, Google Chrome. Tīmekļa pārlūkprogrammai jābūt ar Java atbalstu. Tāpat arī jābūt „iPhone” ar uzstādītu lietotni „Controller”.

#### **4.1.3 Dokumenta nolūks**

Šī dokumenta mērķis ir izglīt lietotājus par sistēmas saskarni un operācijām, kā arī kalpot par informācijas avotu neskaidru jautājumu izskaidrošanā.

#### **4.1.4 Saistītie dokumenti**

Dokuments tiek bāzēts uz LVS 66:1996 «Programmatūras lietotāja dokumentācija» standarta.

Šīs sistēmas PPS.

#### **4.1.5 Dokumenta lietošanas apraksts**

Dokumentu ir vēlums izlasīt visu secīgi. Ja ir kāda neskaidrība, vajadzētu meklēt atbilstošus šīs dokumentācijas nodaļu nosaukumus.

#### **4.1.6 Pieņemtie apzīmējumi un vienošanās.**

Sistēmas daļa – Serveris ar servera sīkietotni un tīmekļa vietni ir ieslēgti un pieejami, izmantojot internetu. Tāpat arī uz lietotāja „iPhone” ir uzstādīta lietotne „Controller”.

Tīmekļa vietnes adrese caur kuru pieejama spēle „Pong” ir [martinsberzins.no-ip.org:8080/ WebContentManipulation/](http://martinsberzins.no-ip.org:8080/WebContentManipulation/)

Tiek uzskatīts ka lietotājam ir pieredze „iPhone” un tīmekļa pārlūkprogrammu izmantošanā.

#### **4.1.7 Problēmu ziņošana**

Par problēmām vai neskaidrībām, ziņot [martins.ber@inbox.lv](mailto:martins.ber@inbox.lv).

## 4.2 Sistēmas lietošana

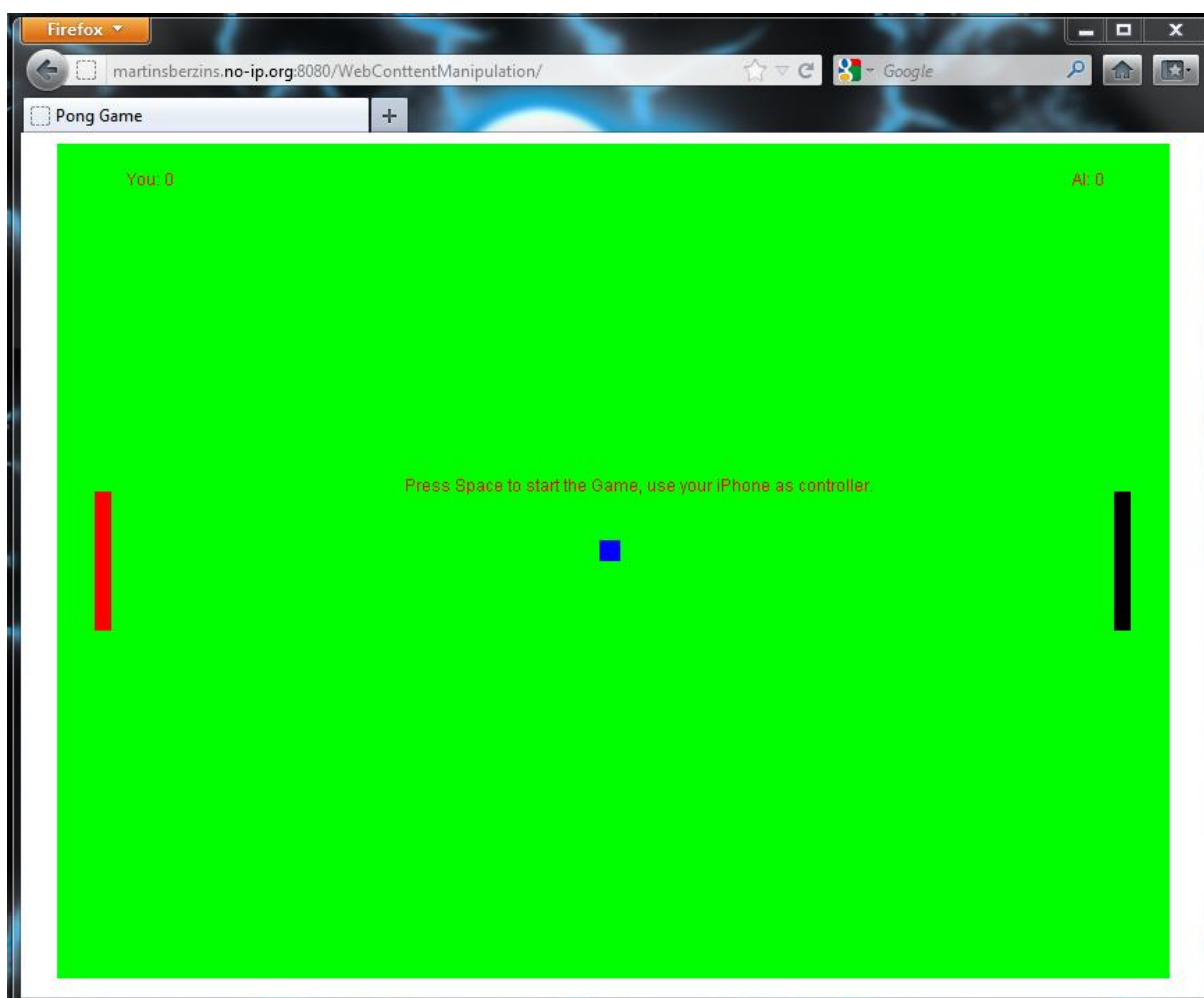
### 4.2.1 Ievads

„Tīmekļa vietnes manipulācija no skārienjūtīga ekrāna ierīces” – turpmāk sistēma, ir paredzēta izklaidei. Sistēmas lietošana ir – vienlaicīgi izmanto „Controller” lietotni uz „iPhone” un tīmekļa pārlūkprogrammā atvērta vietne, kura atrodas uz sistēmas servera, parādot pārlūkprogrammā spēli „Pong”.

### 4.2.2 Darbības sākums

Sistēmas darbības uzsākšana notiek šādi:

- Lietotājs ar savu tīmekļa pārlūkprogrammu atver sistēmas vietnes adresi



[Attēls. 3]

Tīmekļa pārlūkprogrammā parādās spēle „Pong”. Piezīme, spēle ir 800\*600 px liela, tādēļ lai redzētu visu spēli, datora ekrānam vai tīmekļa pārlūkprogrammas logam jābūt atbilstošā lielumā.

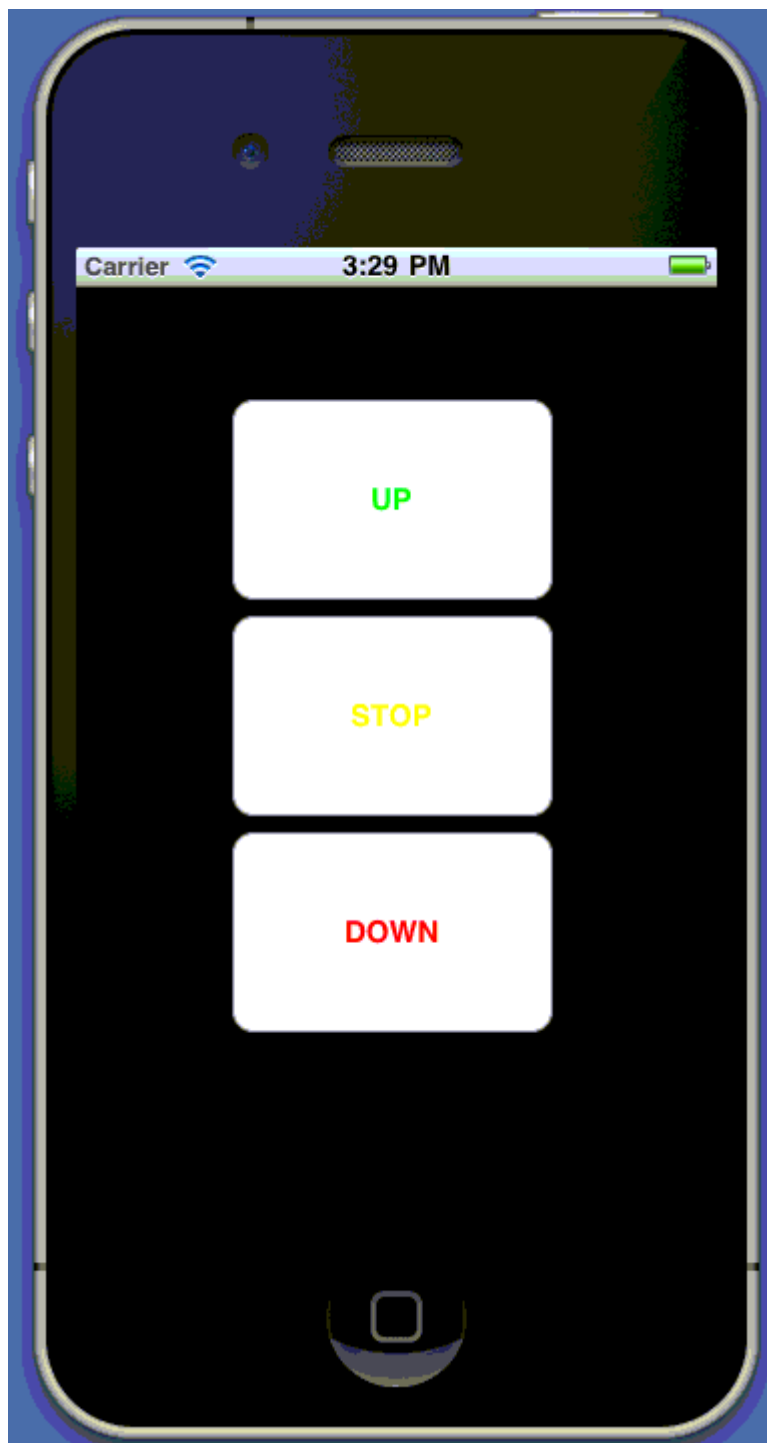
Spēle attēlo:

- Laukuma kreisajā pusē – vidū spēlētāja rakete, augšā spēlētāja iegūto punktu skaits.
- Laukuma vidū- Spēles spēlēšanas instrukcijas, zem tām bumba.
- Laukuma labajā pusē – vidū MI rakete, augšā MI iegūtais punktu skaits.

Instrukcija ir – „Press Space to start the Game, use your iPhone as controller”.

Tulkojumā Latviski – „Spied pogu „Space”, lai sāktu spēli, izmanto savu „iPhone”, kā pulti”.

- Lietotājs ieslēdz „iPhone” lietotni „Controller”



[Attēls 4]

Uz „iPhone” ekrāna parādās trīs pogas – UP, STOP, DOWN. Tulkojumā – Uz augšu, apstāties, uz leju.

### 4.2.3 Darbība ar sistēmu

Pieskaroties ar pirkstu uz kādas no attēlā.4 redzamajām pogām (UP, STOP, DOWN)

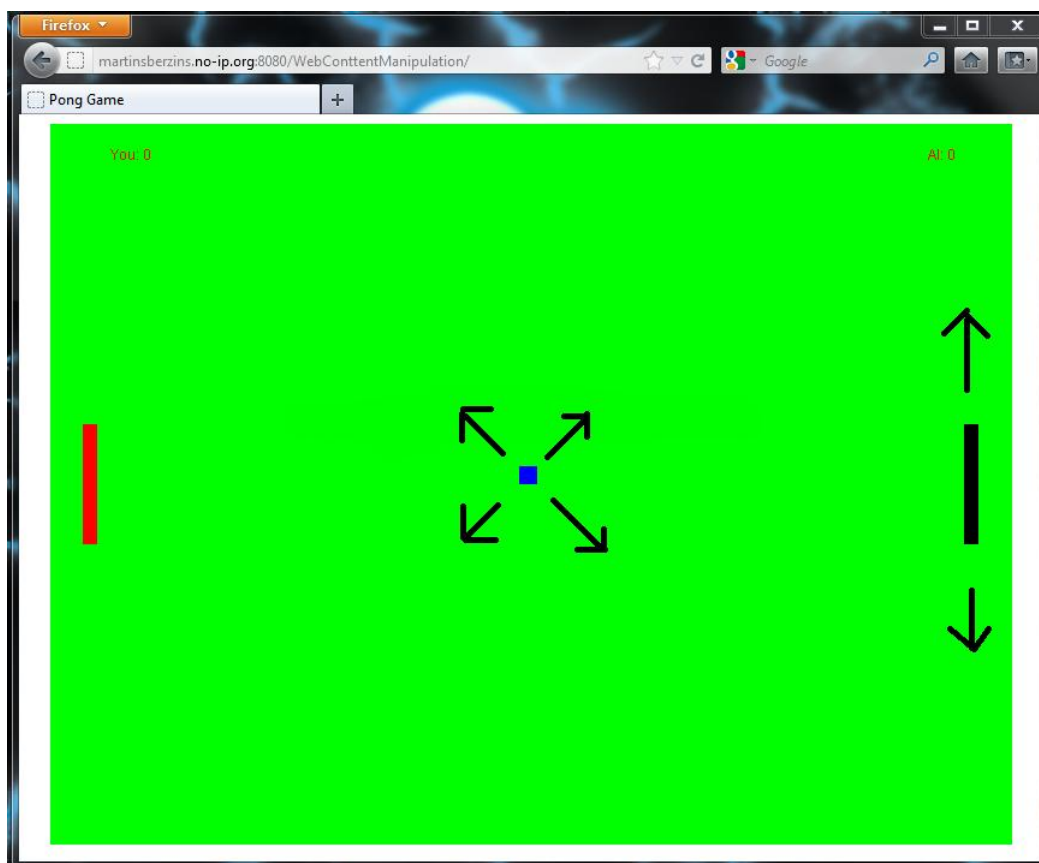
Nospiežot pogu (ar pirkstu pieskaroties pogas laukam), ja spēle ir sākta tiks mainīts spēlētāja raketes virziens. Ja spēle nav sākta nekas nenotiks.

Spēles sākums:

Spēli sāk, kad lietotāja pārlūkprogrammā parādās spēle un tās instrukcijas. Instrukcijās norādīts ka spēli sāk nospiežot datorā taustiņu „Space”.

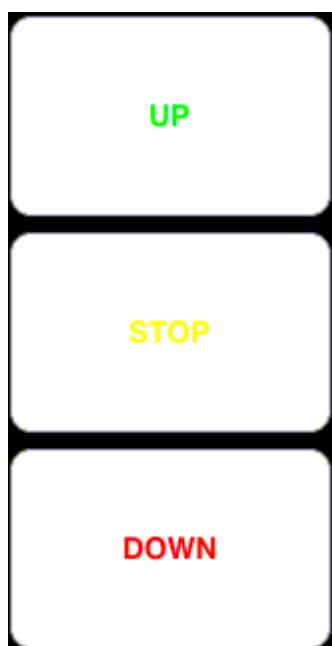
Kad „Space” ir noklikšķināts sākas spēle. Bumba sāk kustēties, MI rakete cenšas to atsist. Ja bumba pieskaras kādai sānu malai (kuras sargā spēlētāja vai MI rakete), otras puses raketes punktu skaits tiek palielināts par viens.

Attēls 5 attēlo, spēles komponentu bumbas un MI raketes iespējamus kustības virzienus.



[Attēls 5]

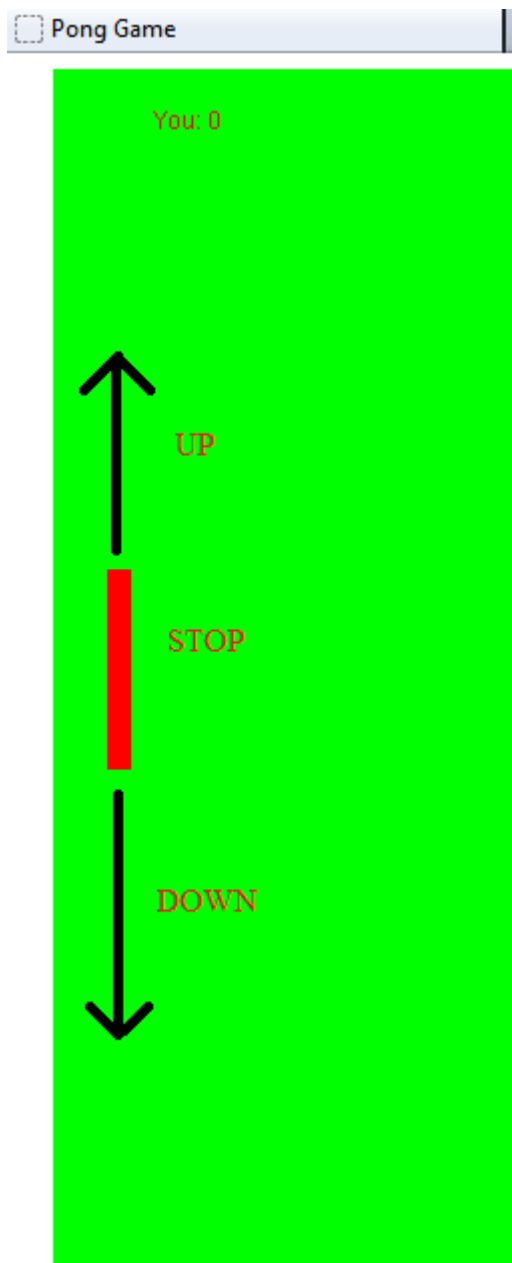
Spēlētāja raketes kustināšana:



Spiežot pogu UP  
 ,spēlētāja rakete sāk  
 pārvietoties UP  
 norādītāja virzienā.

STOP, liek raketei  
 apstāties.

DOWN, sāk pārvietot  
 raketi norādītājā  
 virzienā



#### 4.2.4 Darbību beigšana.

Kad lietotājs izlēmis vairs spēli nespēlēt, spēles logs tīmekļa pārlūkprogrammā jāizver un jāiziet no „iPhone” lietotnes „Controller”.

## **5. Projekta organizācija**

Projekta darba organizācija tika izpildīta pēc evolucionārās prototipēšanas dzīves cikla modeļa. Jo izstrādes sākumā tika precizētas pirmās idejas par to, ko produktam būs jādara. Pēc ideju apkopošanas tika nolemts par galvenajām sistēmas izpildāmajām funkcijām. Sistēmas izstrādi tika paredzēts sadalīt moduļos. Pabeidzot vienu sistēmas moduli, var tikt izvirzītas jaunas prasības. Bija svarīgi saskaņot jaunās prasības ar sistēmas pamat funkcionalitāti.

Dokumentācijā prezentētie PPS un PPA ir visas sistēmas prototipu versiju, prasību un projektējumu apkopojumi.

Šai sistēmai nav pasūtītāju, tādēļ sistēmas veidotājs pats izvirzīja prasības sistēmai, konsultējoties ar kvalifikācijas darba vadītāju.

## 6. Kvalitātes nodrošināšana.

Lai nodrošinātu izstrādātā projekta kvalitāti, tika veiktas šādas darbības:

- Programmatūras dokumentācija tika veidota saskaņā ar standartiem.;

LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis”

LVS 72:1996 "Ieteicamā prakse programmatūras projektējuma aprakstīšanai"

LVS 66:1996 «Programmatūras lietotāja dokumentācija»

- Programmas kods tika sadalīts moduļos, atkarībā no to veiktās darbības.
- Kods tika izstrādāts izmantojot objektorientēto programmēšanas stilu.
- Kods ir labi komentēts ir izskaidrotas metožu nepieciešamības.
- Kods tika izkārtots vienādā stilā, līdzīgajos moduļos.
- Sistēma ir izstrādāta tā lai to varētu viegli modificēt un papildināt.
- Sistēmā veidotās saskarnes ir veidotas intuitīvi un viegli saprotamas.
- Sistēmas izstrādē tika veikta koda testēšana, lai pārliecinātos, ka sistēma strādā pēc prasībām.

## **7. Konfigurāciju pārvaldība**

Programmprodukta izstrādē piedalījās viens programmētājs. Katras darbadienas beigās izstrādes projekts tika saglabāts SVN. Drošības pēc ar laika intervāliem tika izveidota visas sistēmas kopija, kura tika saglabāta atšķirīgos datu nesējos un „DropBox” failu glabātuvē.

Sistēmas versijas numuru veido trīs cipari, šādā sadalījumā X.Y.Z, X norāda uz sistēmas paaudzi, Y- lielas izmaiņas sistēmā, Z-nelielas izmaiņas sistēmā.

Kad šis programmatūras produkts tika izlaists, sistēmas versija ir 1.0.0.

## 8. Darbietilpības novērtējums.

Tika izmantots COCOMO tīmeklī pieejamais kalkulators,

Kalkulatora ievadītās vērtības.

- size = 1100
- mode = 1.05
- rely = 1.00
- data = 0.94
- cplx = 1.15
- time = 1.00
- stor = 1.00
- virt = 1.00
- turn = 1.00
- acap = 1.00
- aexp = 1.00
- pcap = 1.00
- vexp = 1.00
- lexp = 1.00
- modp = 0.91
- tool = 1.00
- sced = 1.00

Iegūtais rezultāts:

Effort = 3.15 Person Months

Schedule = 3.87 Months

Tulkojumā:

Nepieciešamais darbs ir 3.15 mēneši.

Darba Grafiks 3.87 mēneši.

## **9. Izmantotās Tehnoloģijas**

Sistēmas izstrāde tika izmatotas šādas tehnoloģijas

- Programmēšanas valoda Java;
- Programmēšanas valoda Objective-C;
- Iezīmēšanas valoda HTML;
- Programmēšanas vide „Spring ToolSuite”;
- Programmēšanas vide „XCode”;
- Tekstu redaktors „Notepad++”;
- „TomCat” serveris;
- Balsamiq Mockups

## 10. Programmatūras izejas kods

Šeit ievietotais kods atbild par sistēmas pamat funkciju – Saziņu starp „iPhone” lietotni „Controller”, serversīk lietotni un sīk lietotni „Pong”.

„Controller” atbilstošais kods :

```
//
// ControllerAppViewController.m
// ControllerApp
//
// Created by Martins Berzins on 5/7/12.
// Copyright 2012 __MyCompanyName__. All rights reserved.
//

#import "ControllerAppViewController.h"

@implementation ControllerAppViewController

-(IBAction)Click:(id)sender{
    NSString *nameOfButton = [sender titleForState:UIControlStateNormal];

    if ([nameOfButton isEqualToString:@"UP"]) {

        NSURL *url = [NSURL URLWithString:@"http://martinsberzins.no-
ip.org:8080/WebContentManipulation/Listener?direction=1"];
        NSURLRequest *request = [NSURLRequest requestWithURL:url];
        NSURLConnection *connection = [NSURLConnection
connectionWithRequest:request delegate:self];

        }else if ([nameOfButton isEqualToString:@"STOP"]) {

        NSURL *url = [NSURL URLWithString:@"http://martinsberzins.no-
ip.org:8080/WebContentManipulation/Listener?direction=0"];
        NSURLRequest *request = [NSURLRequest requestWithURL:url];
        NSURLConnection *connection = [NSURLConnection
connectionWithRequest:request delegate:self];

        }else if ([nameOfButton isEqualToString:@"DOWN"]) {

        NSURL *url = [NSURL URLWithString:@"http://martinsberzins.no-
ip.org:8080/WebContentManipulation/Listener?direction=-1"];
        NSURLRequest *request = [NSURLRequest requestWithURL:url];
        NSURLConnection *connection = [NSURLConnection
connectionWithRequest:request delegate:self];

        }

    }

/*
// The designated initializer. Override to perform setup that is required
before the view is loaded.
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle
*)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
```

```

        // Custom initialization
    }
    return self;
}
*/

/*
// Implement loadView to create a view hierarchy programmatically, without
using a nib.
- (void)loadView {
}
*/

/*
// Implement viewDidLoad to do additional setup after loading the view,
typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
}
*/

/*
// Override to allow orientations other than the default portrait
orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interf
aceOrientation {
    // Return YES for supported orientations
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc that aren't in use.
}

- (void)viewDidUnload {
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (void)dealloc {
    [super dealloc];
}

@end

```

Serversīklīetotnei atbilstošais kods:

```
// Listener.java
/*****
 * Gets data from users "iPhone" application "Controller" stores them in variable
 * Sends the variable to game "Pong"
 *****/
// Author: Mārtiņš Bērziņš

import java.io.IOException;
import java.io.ObjectOutputStream;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlets implementation class Listener
 */
@WebServlet("/Listener")
public class Listener extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Listener() {
        super();
    }

    //variable for storing players commands - In which direction to move.
    int dir;

    /**
     * This method listens "iPhone" applications "Controller" sent URLs. Then
     * gets the parameter from URL. Parameter is stored in memory as variable.
     * Parameter contains user commands. Commands are to move UP represented by
     * value +1, Stop represented by value 0 and DOWN represented by value -1.
     *
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     * response)
     */
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {
        response.setContentType("text/html");

        // ?direction= -1 || 0 || +1

        String dirFromUrl = request.getParameter("direction");

        if (dirFromUrl == null)
            dirFromUrl = "0";

        int direction = Integer.parseInt(dirFromUrl);

        if (direction == -1 || direction == 0 || direction == +1) {
            setDirection(direction);
        }
    }
}
```

```

        } else
            setDirection(0);
    }

    /**
     * Method that listens games requests for direction. when requests is made
     * method sends object with necessary data to the game.
     *
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     * response)
     */
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
and IOException {

        gameInfo responseObj = respondToApplet();
        response.reset();
        response.setHeader("Content-Type",
            "application/x-java-serialized-object");
        ObjectOutputStream outputToApplet;
        outputToApplet = new ObjectOutputStream(response.getOutputStream());
        outputToApplet.writeObject(responseObj);
        outputToApplet.flush();
        outputToApplet.close();

    }

    /**
     * This is response class is used to store necessary parameters, methods
and
     * then is sent over the Internet.
     *
     * @author Mărtiņš
     */
    public static class gameInfo implements java.io.Serializable {
        int direction;

        public gameInfo() {
            direction = 0;
        }

        public void setDirection(int dir) {
            direction = dir;
        }

        public int getDirection() {
            return direction;
        }

    }

    /**
     * Method does the function requested by game used to respond to game
     * "Pong". From Servlets memory gets the variable for storing layer
     * direction command. And then sets it in response object.
     *
     * @return myrespObj
     */

```

```

private gameInfo respondToApplet() {
    gameInfo myrespObj = new gameInfo();

    if (dir == -1 || dir == 0 || dir == 1) {
        myrespObj.setDirection(dir);
    } else
        myrespObj.setDirection(0);

    return myrespObj;
}

/**
 * method for returning direction objects direction variable. Used in
"Pong"
 * game.
 *
 * @param directionFromServlet
 * @return
 */
int getPlayerDir(gameInfo directionFromServlet) {
    int dirObjData = directionFromServlet.getDirection();
    return dirObjData;
}

void setDirection(int direction) {
    dir = direction;
}

int getDirection() {
    return dir;
}
}

```

„Pong” spēles kods kurš sazinās un iegūs datus no serversīkļietotnes.

```

/**
 *
 * Method connects to servlet on server and gets players direction.
 *
 * @return direction of player
 * @throws IOException
 */
public void getDirectionUsingHttp() throws IOException {

    // Makes servlet object, reason for doing it Servlet class holds
    // response object class
    Listener.gameInfo gameStatusObj = new Listener.gameInfo();

    URL servletURL = new URL(
        "http://martinsberzins.no-
ip.org:8080/WebContentManipulation/Listener");
    HttpURLConnection servletConnect = (HttpURLConnection) servletURL
        .openConnection();
}

```

```

        servletConnect.setDoOutput(true); // to allow write to the URL
        servletConnect.setUseCaches(false); // Write the message to the
servlet
                                                    // and not
from the browser's cache
        servletConnect.setRequestProperty("Content-Type",
            "application/x-java-serialized-object");
        servletConnect.setRequestMethod("POST");// Uses the POST method of
                                                    //
servlet "Listener"
        ObjectInputStream in = new ObjectInputStream(
            servletConnect.getInputStream());

        try {

            gameStatusObj = ((Listener.gameInfo) in.readObject());

            setPlayerDir(gameStatusObj.getDirection());

            in.close();

        } catch (ClassNotFoundException e1) {
            e1.printStackTrace();
        }

        in.close();

    }

```

## **11.Pateicības.**

Kvalifikācijas darba vadītājam : Oļegam Urzikam.

Uzņēmumam „Exigen”

<http://stackoverflow.com/> Komandai un censoņiem.

## 12. Izmantotā literatūra un avoti

1. LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis”
2. LVS 72:1996, Ieteicamā prakse programmatūras projektējuma aprakstīšanai.
3. LVS 66:1996 Programmatūras lietotāja dokumentācijas standarts.
4. <http://docs.oracle.com/javase/6/docs/>
5. <http://www.springsource.com/developer/sts>
6. <http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>
7. <http://termini.lv>
8. [http://sunset.usc.edu/research/COCOMOII/cocomo81\\_pgm/cocomo81.html](http://sunset.usc.edu/research/COCOMOII/cocomo81_pgm/cocomo81.html)