**Datorspēle “Mērķēšanu trenētājs”**

PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS (PPA)

SS.PPA.1.0.7.

Izstrādātāji:

Gustavs Narvils

Gustavs Pauls

Liepāja 2023

|  |  |
| --- | --- |
| **Dokumenta identifikācija** | |
| Dokumenta ID: | SS.PPA.1.0.7. |
| Dokumenta nosaukums: | Datorspēle “Mērķēšanu trenētājs” Programmatūras prasību apraksts |
| Versija: | 1.0.7. |

# Projekta darba grupa

No izpildītajā puses:

Gustavs Narvils SIA “Start Strong” IT projekta vadītājs

Gustavs Pauls SIA “Start Strong” sistēmu analītiķis

No pasūtītāja puses:

Kristaps Rāvalds SIA “Mērķēšanu trenētājs” projekta vadītājs

# Izmaiņu lapa

**Dokumenta versijas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versija** | **Izmaiņas** | **Autors** | **Datums** |
| 1.0.0. | Izveidots dokuments | Gustavs Narvils  Gustavs Pauls | 16.03.2023. |
| 1.0.1. | Izveidota nodaļa “Ievads” | Gustavs Pauls | 16.03.2023. |
| 1.0.2. | Izveidota nodaļa “Dekompozīcijas apraksts” | Gustavs Narvils  Gustavs Pauls | 20.03.2023. |
| 1.0.3. | Izveidotas diagrammas | Gustavs Narvils | 21.03.2023. |
| 1.0.4. | Izveidota nodaļa “Atkarības apraksts” | Gustavs Pauls | 22.03.2023. |
| 1.0.5. | Izveidota nodaļa “Detalizēts projektējums” un “Lietotāju saskare”, un “Izstrādes rīki” | Gustavs Narvils  Gustavs Pauls | 23.03.2023. |
| 1.0.6. | Izveidota nodaļa “Pievienotā vērtība” un dokuments noformēts | Gustavs Narvils | 26.03.2023. |
| 1.0.7. | Pielabots dokuments | Gustavs Narvils  Gustavs Pauls | 25.06.2023. |

Saturs

[1. Ievads 6](#_Toc130742606)

[1.1. Dokumenta nolūks 6](#_Toc130742607)

[1.2. Darbības sfēra 6](#_Toc130742608)

[1.3. Definīcijas un saīsinājumi 7](#_Toc130742609)

[1.4. Saistība ar citiem dokumentiem 7](#_Toc130742610)

[1.5. Dokumenta pārskats 7](#_Toc130742611)

[2. Dekompozīcijas apraksts 9](#_Toc130742612)

[2.1. Moduļu dekompozīcija 9](#_Toc130742613)

[2.1.1. Galvenās izvēlnes apraksts 9](#_Toc130742614)

[2.1.2. Iestatījumu loga apraksts 9](#_Toc130742615)

[2.1.3. Pirmsspēles loga apraksts 9](#_Toc130742616)

[2.1.4. Spēles loga apraksts 9](#_Toc130742617)

[2.2 Datu dekompozīcija 10](#_Toc130742618)

[3. Atkarības apraksts 12](#_Toc130742619)

[3.1. Starp moduļu atkarības 12](#_Toc130742620)

[3.2. Starpprocesu atkarības 13](#_Toc130742621)

[4. Detalizēts projektējums 14](#_Toc130742622)

[4.1. Moduļu detalizēts projektējums 14](#_Toc130742623)

[4.1.1. Galvenās izvēlnes detalizējums 14](#_Toc130742624)

[4.1.2. Iestatījuma loga detalizējums 14](#_Toc130742625)

[4.1.3. Pirmsspēļu loga detalizējums 14](#_Toc130742626)

[4.1.4. Spēles loga detalizējums 14](#_Toc130742627)

[4.2. Datu detalizēts projektējums 15](#_Toc130742628)

[4.2.1. Datu entītijas “Mērķēšanu trenētājs” detalizējums 15](#_Toc130742629)

[4.2.2. Datu entītijas “GalvenāIzvēlne” detalizējums 15](#_Toc130742630)

[4.2.3. Datu entītijas “Iestatījumi” detalizējums 15](#_Toc130742631)

[4.2.4. Datu entītijas “Audio” detalizējums 15](#_Toc130742632)

[4.2.5. Datu entītijas “Pirmsspele” detalizējums 16](#_Toc130742633)

[4.2.6. Datu entītijas “Spēle” detalizējums 16](#_Toc130742634)

[4.2.7. Datu entītijas “Objekti” detalizējums 16](#_Toc130742635)

[4.2.8. Datu entītijas “Fona mūzika” detalizējums 16](#_Toc130742636)

[4.2.9. Datu entītijas “Rezultāti.txt” detalizējums 16](#_Toc130742637)

[4.2.10. Datu entītijas “Iestatījumi.txt” detalizējums 16](#_Toc130742638)

[5. Lietotāju saskare 17](#_Toc130742639)

[5.1.1. Moduļa “Galvenā izvēlne” saskare 17](#_Toc130742640)

[5.1.2. Moduļa “Iestatījuma logs” saskare 17](#_Toc130742641)

[5.1.3. Moduļa “Pirmsspēles logs” saskare 18](#_Toc130742642)

[5.1.4. Moduļa “Spēles logs” saskare 18](#_Toc130742643)

[6. Izstrādes rīki 20](#_Toc130742644)

[7. Pievienotā vērtība 21](#_Toc130742645)

[Pielikumi 22](#_Toc130742646)

# 1. Ievads

Šis dokuments ir izstrādāts saskaņā ar datorspēles “Mērķēšanu trenētājs” izveidošanu. Paredzēts izstrādāt datorspēli “Mērķēšanu trenētājs”, kura prasības apraksts šis dokuments.

## 1.1. Dokumenta nolūks

Šis programmatūras projektējuma apraksta (PPA) nolūks ir saprotami paradīt kā tiks izpildītas un realizētas programmatūras prasības (ar prasībām var iepazīties šī projekta programmatūras prasību specifikāciju dokumentā (PPS)). Ar PPA palīdzību tiek noteikts, kas programmai ir jādara un kā šīs darbības tiek realizētas ar izstrādes procesā izmantoto rīku palīdzību. Dokumenta nolūks ir detalizēti attēlot programmu “Mērķēšanu trenētājs” un katra tās moduļa uzbūvi individuāli, kā arī specificē izstrādes risinājumus.

Datorspēles PPA ir paredzēts izstrādātājiem, bet nav paredzēts to nodot lietošanā produkta pasūtītājam. Šis dokuments parasti nav saprotams lietotājiem bez pieredzes informācijas sistēmu izstrādāšanā un projektēšanā, jo var saturēt nozarei specifiskus apzīmējumus. Pamatojoties uz iepriekš minēto iemeslu, datorspēles “Mērķēšanu trenētājs” PPA ir noteikts par izstrādātāju organizācijas iekšējo dokumentu.

Attiecīgais PPA ir lietojams arī kā palīgs analīze, plānošanā, implementēšanā un lēmumu pieņemšanā.

Visas prasības ir sadalītas entītijās, kur katrai entītijai ir savi atribūti, piemēram, nosaukums, entītijas veids, nolūks, funkcija, pakļautība, atkarības, saskare, apstrāde, dati.

Savukārt katram moduļiem tiek dota precīza projektējuma informācija.

Dokumenta struktūra atvieglo orientēšanos programmatūras darbībai nepieciešamo moduļu izstrādāšanā. Uz šī dokumenta pamata iespējams veidot arī testpiemērus, kuru izdotajiem rezultātiem pilnībā jāatbilst šeit aprakstītajām prasībām.

## 1.2. Darbības sfēra

Dokumentā ir aprakstīts datorspēles “Mērķēšanu trenētājs” programmatūras projektējums. Programmatūras mērķis ir palīdzēt cilvēkiem ar kustības traucējumiem un izklaidēt bērnus, kā arī uzlabot cilvēku datoru peles kontroli. Datorspēles mērķis ir ar moderno tehnoloģijas palīdzību izklaidēt bērnus ar interaktīvo šaušanas spēli un vienlaikus uzlabot datoru peles kontroli, trenēt tās lietotāja uzmanību, kā arī mazināt kustības traucējumus.

Datorspēles uzdevums ir lietotājam uzlabot peles kontroli, kā arī trenēt lietotāju reakciju, ātrumu, precizitāti un telpisko uztveri.

Svarīgi ir panākt situāciju, ka ikviens šīs datorspēles lietotājs varētu tajā intuitīvi orientēties un nedotu lietotājam nekādas grūtības tās izmantošanā.

## 1.3. Definīcijas un saīsinājumi

Lai pareizi un viennozīmīgi interpretētu šo PPA dokumenta 1. un 2. Tabulā ir apkopoti šeit izmantotie termini un saīsinājumi.

**1.tabula**

**Termini**

|  |  |
| --- | --- |
| **Termini** | **Skaidrojumi** |
| Izstrādātājs | Indivīds vai organizācija, kas veido datorspēli pēc pasūtītāja iniciatīvas un prasībām. |
| Lietotājs | Persona vai personas, kas lieto datorspēli noteiktu uzdevumu veikšanai. |
| Pasūtītājs | Indivīds vai organizācija, kas nosaka un akceptē projekta prasības un nodevumus. |
| Produkts | Jebkurš taustāms vienums, kas rodas projekta funkciju, aktivitāšu vai uzdevu rezultātu. |
| Entitijs | Objekts, kas eksistē un, kas satur visas funkcijas priekš tās strādāšanas. |

**2. tabula**

**Saīsinājumi**

|  |  |
| --- | --- |
| **Saīsinājumi** | **Skaidrojumi** |
| PPA | Programmatūras prasību apraksts |
| PPS | Programmatūras prasību specifikācija |

## 1.4. Saistība ar citiem dokumentiem

PPA sastādīšanai kā palīglīdzeklis tika izmantots Latvijas Valsts Standarta “Ieteicamā prakse programmatūras projektējuma aprakstīšanai” (LVS 72:1996) dokuments.

Kopā ar attiecīgo programmatūras projektējuma aprakstu, lietojama programmatūras prasību specifikācija (SS.PPS.1.0.3.) datorspēlei “Mērķēšanu trenētājs”.

## 1.5. Dokumenta pārskats

Dokuments sastāv no 7 nodaļām, kur dažām nodalām ir vairākas apakšnodaļas.

1. Nodaļa “Ievads” ir aprakstīts dokumenta nolūks, darbības sfēra, doti izmantoto termiņu un saīsinājumu skaidrojumi, kā arī parādīta saistība ar citiem šī projekta dokumentiem.

2. Nodaļa “Dekompozīcijas apraksts” ir definēti un aprakstīti visi datorspēļu moduļi un datu dekompozīcijas.

3. Nodaļa “Atkarības apraksts” ir definētas un aprakstītas visas saistītās nodaļas, starp moduļu un starpprocesu atkarības.

4. Nodaļa “Detalizētais projektējums” ir definētas un aprakstītas visas nefunkcionālās prasības datorspēlei “Mērķēšanu trenētājs”.

5. Nodala “Lietotāju saskare” ir definētas un aprakstītas lietotāju iespējas izmantojot datorspēli “Mērķēšanu trenētājs”.

6. Nodaļa “Izstrādes rīki” ir definētas un aprakstītas visas pieejamās izstrādes rīku un valodas izvēles, kuras var izmantot datorspēles realizēšanā.

7. Nodaļa “Projekta ieguldījums” ir aprakstīts, katra projekta izstrādāta paveiktais darbs.

Šajā PPA dokumentā ir pievienoti trīs pielikumi, kurā ir attēlota datorspēles “Mērķēšanu trenētājs” USE CASE diagramma (Sk. 1.Pielikums), Iestatījuma loga aktivitāšu diagramma (Sk. 2.Pielikums) un Pirmsspēles loga aktivitāšu diagramma (Sk. 3.Pielikums).

# 2. Dekompozīcijas apraksts

Šajā nodaļa tiek aprakstīta projektu moduļu un datu dekompozīcija.

## 2.1. Moduļu dekompozīcija

Attiecīgajā projektā tiek izdalīti četri atsevišķi moduļi: Galvenā izvēlne, Iestatījumu logs, Pirms spēles logs un Spēļu logs.

### 2.1.1. Galvenās izvēlnes apraksts

Galvenās izvēlnes modulis ir pamats visai datorspēlei, jo šis modulis ļaus lietotājam pārejiet starp moduļiem izmantojot pogu. Šis modulis arī dod iespēju mums viegli pievienot jaunus moduļus un viegli tos integrēt datorspēlē.

### 2.1.2. Iestatījumu loga apraksts

Iestatījuma loga modulis ir vieta, kur lietotājs varēs mainīt visa viņa personīgos iestatījumus. Šo moduli var atvērt no galvenās izvēlnes un tad, kad lietotājs ir pabeidzis visas savas vēlamās iestatījuma maiņas lietotājs no šī moduļa var atgriezties galvenā izvēlnē. Un lietotāja visi iestatījumi tiek saglabāti teksta failā uz viņa datora.

### 2.1.3. Pirmsspēles loga apraksts

Pirmsspēļu loga modulis ir vieta, kur lietotājs varēs mainīt visa viņa spēles iestatījumus. Šo moduli var atvērt tikai no galvenās izvēlnes un arī būs iespēja lietotājam atgriezties galvenā izvēlne. Kad lietotājs ir izvēlējies sev attiecošos spēles iestatījumus, lietotājs varēs sākt spēli ar tiem iestatījumiem izmantot pogu.

### 2.1.4. Spēles loga apraksts

Spēles logs modulis ir vieta, kur notiek visa galvenā spēles daļa. Kura mērķis ir nošaut lietotājam tik daudz mērķis cik viņš var noteiktā laikā. Kad laiks beidzas arī spēle beidzas un parādās rezultātu logs, kur lietotājs var apskatīt savas spēles rezultātus un arī atgriezties galvenā izvēlnē.

## 2.2 Datu dekompozīcija

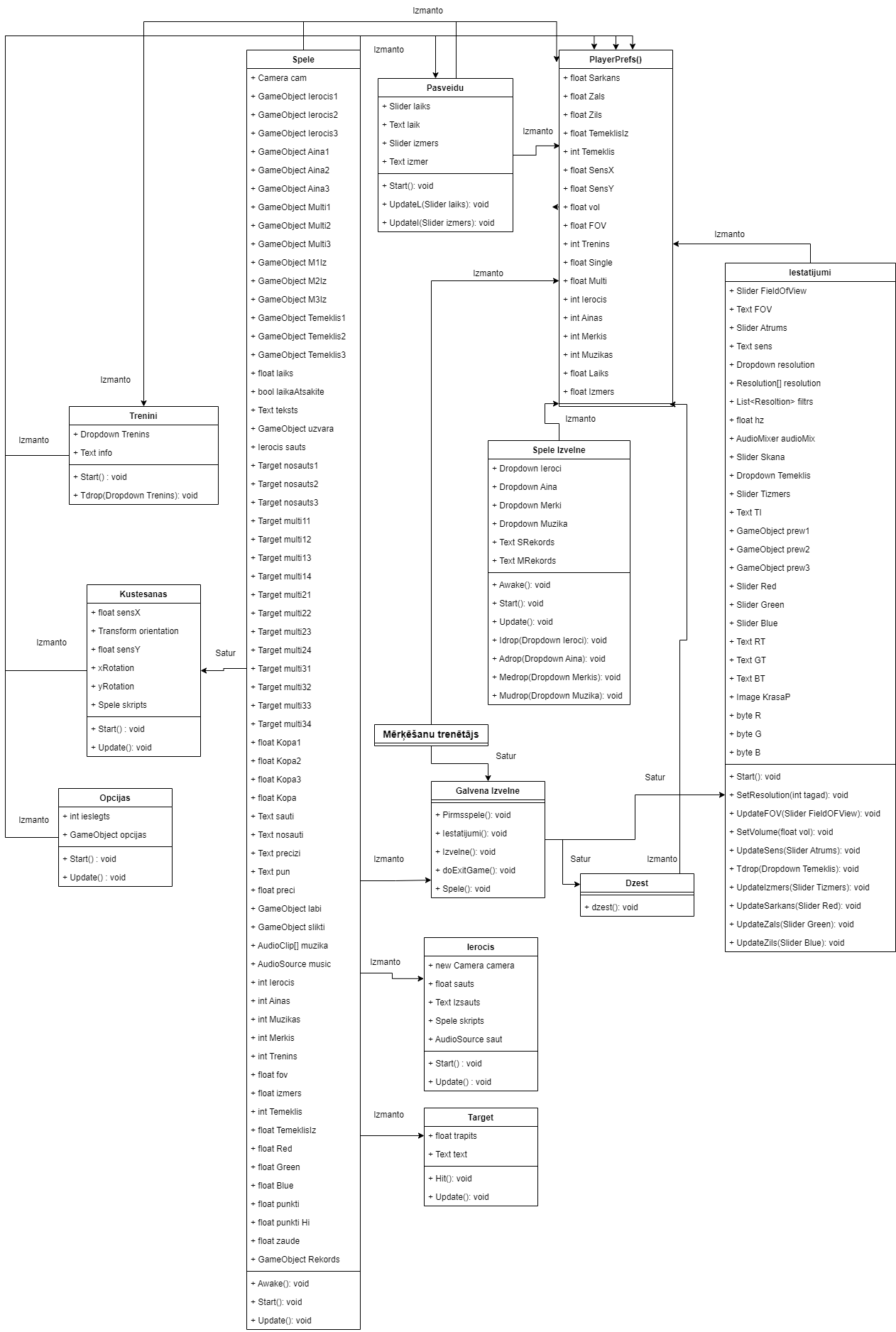
Šī projekta datu dekompozīcija attēlota 3. Tabulā un 1. Attēlā redzamajā produktu klašu diagrammā.

**3. tabula**

**Datu dekompozīcija**

|  |  |
| --- | --- |
| **Entitijas** | **Apraksts** |
| Mērķēšanu trenētājs | Satur visas entititjas, kuras tiks izmantotas projektā. |
| Galvena Izvelne | Entitija, kura tiek saglabātas visas funkcijas un pogas priekš ainu pāriešanas. |
| Iestatījumi | Entitija, kura tiek saglabāti visu iestatījumu opcijas, visas funkcijas saistītas ar tām. |
| Kustesanas | Entitija, kura tiek glabātas funkcijas priekš lietotāja kustināšanas. |
| Ierocis | Entitija, kura tiek glabātas visas funkcijas saistībā ar šaušanu. |
| Spele Izvelne | Entitija, kura satur visas funkcijas saistībā ar modeļu maiņu, kā arī satur funkcijas rezultātu parādīšanai. |
| Opcijas | Entitija, kas satur funkciju priekš papildu treniņu opcijas parādīšānas. |
| Pasveidu | Entitija, kura satur laika un izmēru pogas un funkcijas, kas to kontrolē priekš pašveidu treniņa režīma. |
| Trenini | Entitija, kas satur funkciju priekš treniņu maiņas. |
| Dzest | Entītija, kas satur funkciju izdzēst iepriekšējos rezultātus. |
| Spele | Entītija, kas satur pilnība visus objektus un funkcijas, kas ir saistītas ar spēles norisi. |
| Target | Entitija, kas satur funkciju priekš mērķu nošaušanas. |
| PlayerPrefs() | Unity izveidots fails, kur tiek saglabātas visas vērtības priekš spēles |

**1. Attēls. Klašu diagramma**



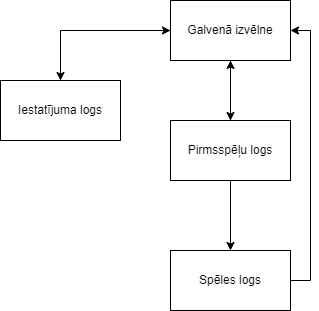
# 3. Atkarības apraksts

Šajā nodaļā ir aprakstīts datorspēles “Mērķēšanu trenētājs” starp moduļu un starpprocesu atkarības.

## 3.1. Starp moduļu atkarības

Lai labāk izprastu sadarbības stratēģiju starp projekta moduļiem, apskatām to saistības un moduļu saķēdēšanas tipu 2. attēlā.

Kā redzam no moduļu atkarībām, galvenā izvēlne ir visas sistēmas pamatā, jo tas sākumā ir vienīgais vaids kā parejiet starp moduļiem. Citu moduļu darbība tiek balstīta uz galvenās izvēlne moduli. Arī var redzams, ka spēļu logu modulis ir atkarīgs no pirmsspēļu moduļa. Bet arī ir redzams kā no spēļu logu var atgriezties atpakaļ uz galveno izvēlni.

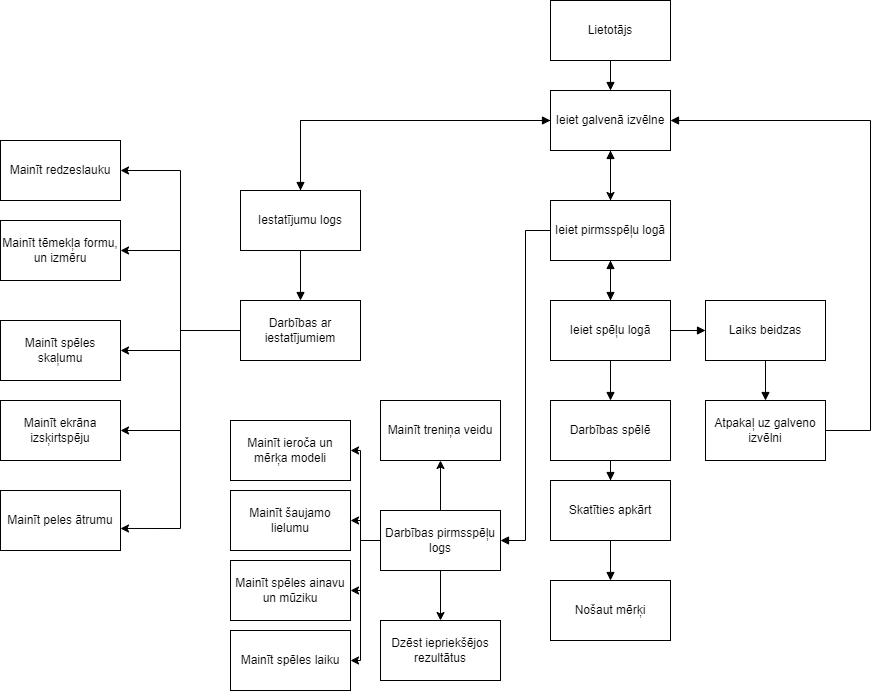


**2. Attēls. Starp moduļu atkarības**

## 3.2. Starpprocesu atkarības

Lai labāk izprastu sadarbības stratēģiju starp projekta procesiem, apskatām to saistības un procesu saķēdēšanas tipu 3. attēlā.

Kā redzams no moduļu atkarībām lietotāja pirmā saskare ar programmu būs galvenā izvēlne, kur lietotājs var iziet uz darbvirsmu, ieejiet iestatījuma logā un mainīt tur iestatījumus, un arī var ieejiet pirmsspēļu logā, kur lietotājs var mainīt spēles iestatījumus pirms sākšanas. Pēc lietotāja iestatījumu regulēšanas viņš var sākt spēli no pirmsspēļu logu, kur mērķis ir nošaut cik daudz mērķus var noteiktā laikā. Pēc tam kad tas ir pabeigts lietotājs var atgriezties galvenā izvēlne.



**3.attēls starpprocesu atkarības**

# 4. Detalizēts projektējums

Šajā nodaļa tiek aprakstīts moduļu un datu detalizēts projektējums.

## 4.1. Moduļu detalizēts projektējums

Moduļu detalizētajā projektējumā ir aprakstīts kādus procesus īsteno katrs modulis, kādam nolūkam tas ir izveidots.

### 4.1.1. Galvenās izvēlnes detalizējums

“Galvenā izvēlne” modulis ir pirmā lieta ar, ko lietotājs saskarsies. Šis modulis ir veidots lai dotu ērtu veidu lietotājam parejiet starp moduļiem izmantot pogas. Pirmā poga nodrošina iespēju tikt pirmsspēļu logu. Otrā nodrošina tikt uz iestatījuma logu. Trešā nodrošina iespēju lietotājam atgriezties uz viņa datora darbvirsmas.

### 4.1.2. Iestatījuma loga detalizējums

“Iestatījuma logs” modulis atļauj lietotājam mainīt spēles uzliktos iestatījums. Lietotājs var ieslēgt vai izslēgt spēles skaņu, kā arī regulēt tās skaļumu. Ir iespējams mainīt tēmēkļa krāsu, kā arī pašu tēmēkli tā izmēru, platumu u.t.t.. Iestatījumu logā ir arī iespējams mainīt lietotāja redzeslauku, kursora ātrumu un arī sava ekrāna izšķirtspēju.

### 4.1.3. Pirmsspēļu loga detalizējums

“Pirmsspēles logs” modulis kurš satur visus lietotāja spēļu iestatījumus, kurus lietotājs var regulēt priekš savām vajadzībām, lai izmainītu spēles gaitu. Pirmsspēļu logs dos iespēju lietotājiem mainīt treniņa veidu, kur viena treniņa veids ir pašveida, kurš dod lietotājam vel mainīt spēles laiku un mērķa izmēru. Pirmsspēļu logs arī dod iespēju lietotājam mainīt ieroča un mērķa modeli, kā arī mainīt laukuma ainavu un mūziku. Un pirmsspēļu logs arī dod iespēju lietotājam dzēst iepriekš iegūtos rezultātus.

### 4.1.4. Spēles loga detalizējums

Modulis “Spēles logs” dod lietotājam iespēju veicināt spēles progresu šaujot izvēlotos objektus, lai sasniegtu spēles beigu. Spēles beigšana sasniedzama, ja spēles laiks ir beidzies. Kad laiks beidzas arī parādās rezultātu logs, kas ļauj apskatīt cik daudz lietotājs reizes lietotājs trāpīja, cik precīzi viņš ir bijis un arī punktu vērtējums. Pēc rezultāta apskatīšanas lietotājs var atgriezties uz galveno izvēlni.

## 4.2. Datu detalizēts projektējums

Datu detalizētāja projektējumā ir aprakstīts kādus procesus īsteno katrs datu entitijs, kādām nolūkam tas ir izveidots

### 4.2.1. Datu entītijas “Mērķēšanu trenētājs” detalizējums

Šī datu entītija ir vieta no kuras visas pārējās datu entītijas strādā. Tā īsti nekādas darbības neveic, bet viņas galvenais uzdevums ir vienot visas pārējās, lai viņas varētu labi strādāt kopā. Šī entītija ir savienota kopā ar “GalvenāIzvēlne” entītiju un “PlayerPrefs()” entītīju.

### 4.2.2. Datu entītijas “GalvenāIzvēlne” detalizējums

Šī datu entītija satur visu vajadzīgo priekš galvenās izvēlnes strādāšanas. Tā glabā visas funkcijas, kuras vajag priekš starp moduļu pāriešanas. Šī entītija ir savienota ar “Dzest” un “Iestatijumi” entītijām.

### 4.2.3. Datu entītijas “Iestatījumi” detalizējums

Šī datu entītija satur gandrīz visu vajadzīgo priekš spēles iestatījuma loga funkcionēšanas. Visas vērtības un funkcijas ir saistītas ar vismaz vienu no piedāvātajām opcijām un viņas tiek izmantotas, lai realizētu šo opciju funkcionēšanu. Šī entītija ir savienota tikai ar PlayerPrefs(), jo izmanto tās funkciju, lai saglabātu vērtības.

### 4.2.4. Datu entītijas “Kustesanas” detalizējums

Šī datu entītija satur visu vajadzīgo priekš lietotāja kameras kustināšanas spēles laikā. Tā glabā lietotāja peles ātrumu, kā arī funkcijas, kas ļauj lietotājam skatīties apkārt izmantojot datora pelīti. Šī entītija ir savienota ar “PlayerPrefs()” entītiju, lai būtu viegli dabūt vērtības priekš peles kustināšanas ātruma.

### 4.2.5. Datu entītijas “Ierocis” detalizējums

Šī datu entītija satur visu vajadzīgo priekš lietotāja šaušanas spēles laikā un arī kontrolē kas notiek pēc mērķa nošaušanas. Šī entītija satur funkciju priekš šaušanas mehānisma strādāšanas, kā arī priekš mērķa parādīšanās pēc nošaušanas.

### 4.2.6. Datu entītijas “Spele Izvelne” detalizējums

Šī datu entītija satur funkcijas priekš spēles modeļu maiņas, gan arī priekš iepriekš iegūto rekordu attēlošanas. Šī entītija satur modeļu maiņu priekš četriem objektiem un tie ir Ierocis, Aina, Mērķis un arī Mūzika. Šī entītija ir tikai savienota ar “PlayerPrefs()” entītiju, jo izmanto to priekš vērtību saglabāšanas un nolasīšanas.

### 4.2.7. Datu entītijas “Opcijas” detalizējums

Šī datu entītija tikai satur funkciju, kas ir tikai priekš papildu opcijas paradīšanas priekš pašveidu treniņiem. Šī entītija ir tikai savienota ar “PlayerPrefs()” entītiju priekš datu nolasīšanas, lai funkcija strādātu.

### 4.2.8. Datu entītijas “Pasveidu” detalizējums

Šī datu entītija satur visus pašveidu treniņa iestatījumus, ko lietotājs var regulēt. Tās funkcijas ir priekš laiku un mērķa izmēra maiņas. Šī entītija ir savienota tikai ar “PlayerPrefs()” entītiju, lai pareizi strādātu saturētās funkcijas.

### 4.2.9. Datu entītijas “Trenini” detalizējums

Šī datu entītija tikai satur tekstu, kas pastāsta lietotājam, kā pareizi spēlēt izvēlēto treniņa veidu. Šī entītija ir tikai savienota ar “PlayerPrefs()” entītiju, lai pareizi varētu nolasīt informāciju.

### 4.2.10. Datu entītijas “Dzest” detalizējums

Šī datu entītija satur funkciju priekš pogas, kas lietotājam ļauj izdzēst iepriekš iegūtos rekordus. Šī entītija ir tikai savienota ar “PlayerPrefs()” entītiju, lai funkcija varētu strādāt.

### 4.2.11. Datu entītijas “Target” detalizējums

Šī entītija satur funkcija satur vajadzīgo funkcija priekš mērķiem un pasaka viņiem, kur viņiem nākošreiz ir jāparadās.

### 4.2.12. Datu entītijas “Spele” detalizējums

Šī entītija satur visas darbības priekš spēles norises un šī entītija tiek tikai izmantota spēles norises laikā. Tas nozīmē, ka šī entītija kontrolē, kas notiek gan spēles laikā, gan pēc spēles beigšanas. Šī entītija galvenā saite ir ar “PlayerPrefs()” entītiju, jo no tās lasa visas vērtības.

### 4.2.13. Datu entītijas “PlayerPrefs()” detalizējums

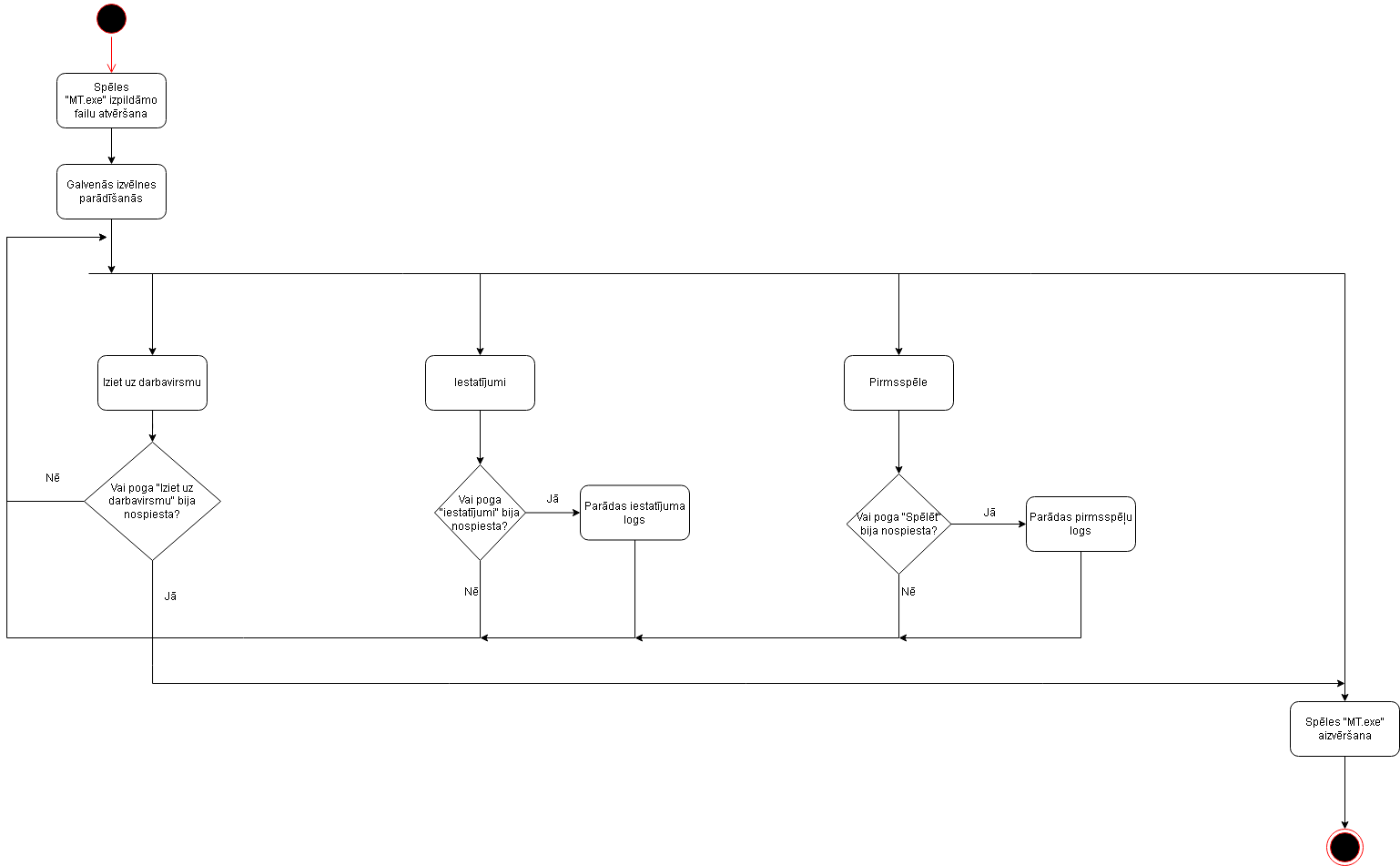
Šī entītija glabā visas vajadzīgās vērtības priekš pārējām entītijām, lai tās varētu viegli saglabāt un paņemt tās starp spēles ainām. Šī entītija ir ļoti svarīga, jo viņa ir savienota ar gandrīz visām pārējām entītijām.

# 5. Lietotāju saskare

Lietotāja saskare ļaus lietotājam izmantot datorspēles “Mērķēšanu trenētājs” funkcijas, izvēloties attiecīgo iespēju vai iespējas.

## 5.1.1. Moduļa “Galvenā izvēlne” saskare

Modulis “Galvenā izvēlne” ir redzams tad, kad tiek atvērta datorspēle “Mērķēšanu trenētājs”, kur lietotājs var nospiest trīs pogas, kas aizvedīs katru uz savu ainu. Poga “Spēlēt” novieto lietotāju uz pirms spēles logu, kur lietotājs varēs izvēlēties savu karti ar savu izvēlēto treniņu. Poga “Iestatījumi” ļauj lietotājam rediģēt esošos iestatījumus priekš savām personīgajām vajadzībām. Poga “Iziet uz darbvirsmu” lietotājam izslēdz ārā MT.exe un noved atpakaļ uz lietotāja darbvirsmu. Lai labāk saprastu lietotāja saskari izmantojot galveno izvēlni var apskatīt 4. attēla aktivitāšu diagrammu.



**4. attēls galvenās izvēlnes aktivitāšu diagramma.**

## 5.1.2. Moduļa “Iestatījuma logs” saskare

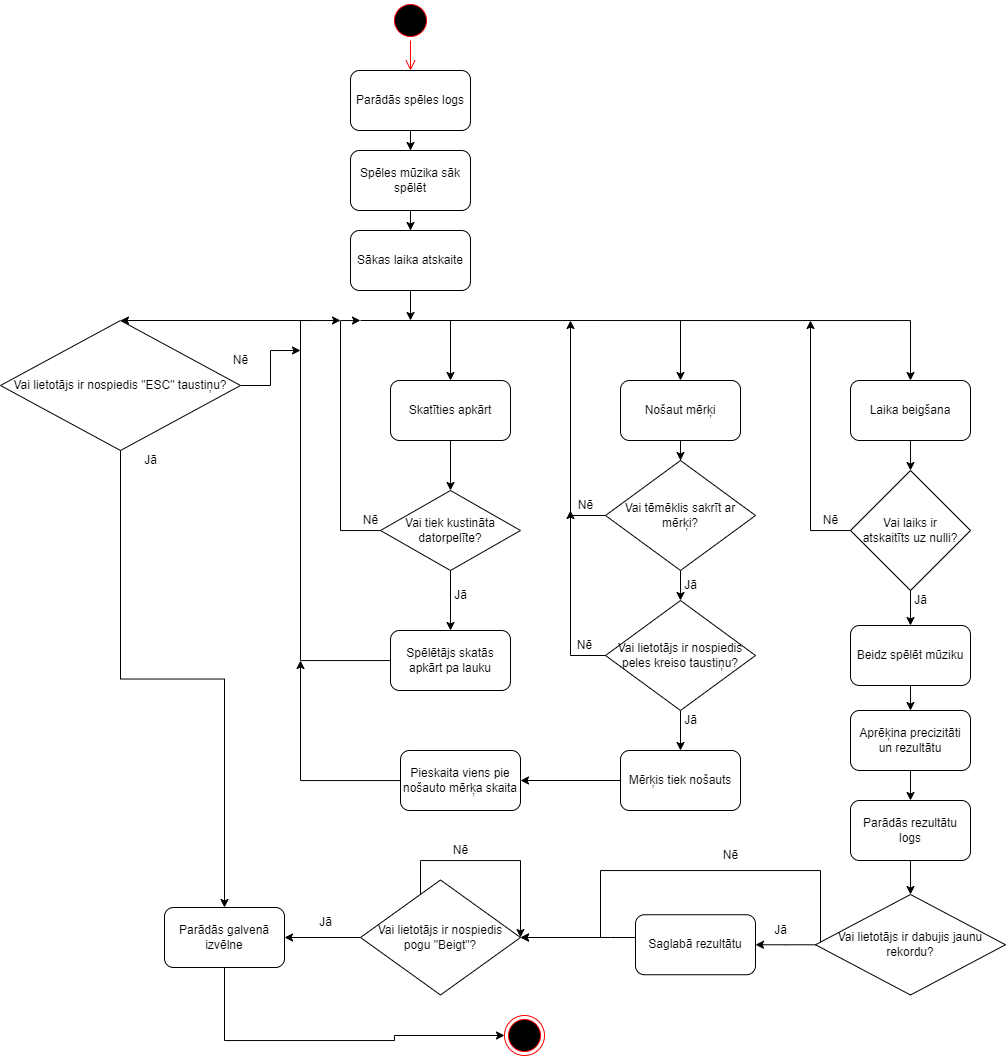
Modulis “Iestatījuma logs” ir redzams tikai tad , kad lietotājs ir nospiedis pogu “Iestatījumi” galvenā izvēlnē. “Iestatījuma logs” atrodas iestatījumi, kur var regulēt proporcionāli kursora tēmēkļa platumu un augstumu ar bīdņa palīdzību, kā arī var mainīt kursora tēmēkļa krāsu ar bīdņu palīdzību un arī mainīt to formu ar izvēlnes lauciņiem. Iestatījumos ir iespējams arī regulēt skaņas skaļumu ar bīdņa palīdzību. Šajā logā var arī mainīt lietotāju redzeslauku un peles ātrumu pateicoties diviem atsevišķiem bīdņiem. Lai labāk saprastu lietotāja saskari izmantojot iestatījuma logu var apskatīt 2. pielikuma aktivitāšu diagrammu.

## 5.1.3. Moduļa “Pirmsspēles logs” saskare

Modulis “Pirms spēles logs” ir redzams tikai tad, kad lietotājs ir nospiedis pogu “Spēlēt” galvenā izvēlnē. Kad nospiež pogu lietotājs tiek pārvietots uz pirms spēles loga, kur lietotājs var izvēlēties savu ainavu un mūziku, kā arī var mainīt ieroča un mērķa modeli. Pirms spēles logā var izvēlēties treniņus kā: Vienšaušana, Multišaušana un pašu veida. Lietotājs izvēlas noteikto treniņu un nospiež pogu “Sākt” , lietotājs tiek aiznests tālāk uz “Spēles logs”, kur notiek paša spēle. Ja lietotājs ir izvēlējies pašu vieda treniņu, lietotājam tiek dota iespēja mainīt spēles laiku un mainīt mērķa izmēru. Lietotājam ir arī dota iespēja mainīt savu lietotājvārdu un izdzēst savus iepriekšējos rezultātus ar pogu “Izdzēst”. Lai labāk saprastu lietotāja saskari izmantojot pirmsspēles logu var apskatīt 3. pielikuma aktivitāšu diagrammu.

## 5.1.4. Moduļa “Spēles logs” saskare

Modulis “Spēles logs” tikai sākas, kad lietotājs ir pārvietots no “Pirms spēles logs” un ir tikusi uzspiesta poga ”Sākt”. Lietotājs spēles logā var šaut par nekustamiem objektiem, kas viņam ir priekšā vai augšā. Ja lietotājs spēles laikā grib atgriezties atpakaļ uz galveno izvēlni viņam ir jānospiež “ESC” taustiņš uz klaviatūras. Pēc spēles beigšanas lietotāju var apskatīt savu iegūto rezultātu logā, kas parādīsies uz ekrāna, kur varēs redzēt cik mērķi ir nošauti, cik precīzi ir šauts, cik punktus ir ieguvis, kā arī pasaka vai tas iegūto punktu daudzums ir jauns rekords. Pēc rezultātu apskatīšanas, lietotājs var uzspiests pogu atpakaļ lai atgrieztos uz “Galvenās izvēlnes” un lietotāju rezultāti tiks saglabāti. Lai labāk saprastu lietotāja saskari izmantojot spēles logu var apskatīt 7. attēla aktivitāšu diagrammu.



**5. Attēls Spēles logs aktivitāšu diagramma**

# 6. Izstrādes rīki

Izstrādājot datorspēli “Mērķēšanu trenētājs”, mēs plānojam izmantot izstrādes vidē “Unity”, kur mēs plānojam veidot visu spēli. Spēles kodu mēs taisīsim iebūvētajā “Unity” Visual Studio koda redaktoru, priekš kodu rakstīšanas un atkļūdošanas. Valoda kurā kods tiks rakstīts ir C#.

Spēle tiek izstrādāta izmantojot ”Unity” vidi, jo mums tā ir viss pazīstamākā un arī viņa daudz labāk iet uz vājiem datoriem. Spēle varēja tikt izstrādāta citā vidē kā piemēram “Unreal Engine”, bet tas nav zināms vai tā būs piemērota pēc šāda apjoma projekta un arī vai mēs sapratīsim, kā to izmantot līdz vajadzīgam izstrādes datumam.

Tas nenozīmē ka tas ir iespējams tikai ar C#, mēs varam lietot arī citus alternatīvus kā JavaScript, Java vai C++, ja tas ir nepieciešams, bet tā kā tiks izmantota “Unity” platforma, tad tādas opcijas neradīsies. Spēles 3D moduļi tiks iegūtu no “Unity Asset Store”, bet ja tas ir vajadzīgs, tad varēs arī izstrādāt pašu savus vajadzīgos modeļus priekš spēles.

# 7. Projekta ieguldījums

Izstrādājot projektu abi projekta izstrādātāji sadalīja, ko katrs darīs projekta izstrādes laikā.

## 7.1. Gustavs Narvils

Sākumā veidojot šo projekta PPS mans galvenais darbs bija izveidot funkcionālās prasības priekš šīs datorspēles kura bija 3. nodaļa PPS dokumentā kā arī pēc tam to dokumentu noformēt pēc skolas prasībām. Kamēr rakstot funkcionālās prasības sāku domāt kā jau lēnām realizēt dažas šīs datorspēles funkcijas kā arī tās varētu reālistiski izskatīties un vai tāda prasība ir vispār bija noderīgas vai vajadzīgas priekš šī projekta. Pēc visu funkcionālo prasības uzrakstīšanas un pārdomāšanas man iznāca kopā 34 prasības priekš šī projekta.

Pēc PPS uzrakstīšanas bija jāraksta PPA no kura es rakstīju Dekompozīcijas aprakstu, Atkarībās aprakstu un Detalizētu projektējuma nodaļas, kuras šī gadījumā bija 2., 3. un 4. nodaļa. Tā pat kā PPS mans uzdevums rakstot PPA bija izdomāt kā visas spēles funkcijas un entītijas būs savienotas, un kā apmēram strādās spēles funkcionēšana, bez spēles taisīšanas. Man arī pēc tam vajadzēja veidot visas astoņas diagrammas, kuras bija klašu 1. attēls, starp moduļu atkarības 2. attēls, starp procesu atkarības, aktivitāšu diagramma priekš galvenās izvēlnes 4. attēls, aktivitāšu diagramma priekš iestatījuma loga 2. pielikums, aktivitāšu diagramma priekš pirmsspēļu loga 3. pielikums, aktivitāšu diagramma priekš spēles loga 5. attēls un USE CASE diagramma 1. pielikums.

Kad sāka projekta izstrādi mans darbs sākumā bija vispirms izveidot GitHub krātuvi, kur projekts tiks glabāts. Bet pēc tam kad bija jārealizē spēle mans darbs bija izveidot galveno spēles logu, kā arī izveidot pirmsspēļu logu. Tādēļ man bija jāizveido visas vajadzīgās funkcijas priekš šiem abiem logiem. Pirmsspēļu logā es izveidoju to, ka var mainīt lietotāja ieroci, mērķi, ainu un mūziku, kā arī atradu visus vajadzīgos modeļus un skaņas priekš visiem. Es arī uztaisīju to, ka var mainīt treniņa veidu un arī to, ka parādās iepriekš iegūtais rezultāts un ka to var lietotājs izdzēst. Es arī izveidoju to, ka lietotājs var mainīt spēles laiku un mērķu izmērus. Veidojot pirmsspēļu logu es arī pievienoju funkciju, kas parādīs katra treniņu aprakstu, kā arī pielāgoju visas spēles ainas tā lai visas pogas un teksti būtu ekrānā neatkarībā kāda ir lietotāja ekrānu malu attiecība.

Spēļu loga es izveidoju, ka var kustināt kameru, nošaut mērķi un funkciju, kas nodrošina mērķu parādīšanu cita vietā pēc mērķa nošaušanas. Pēc tam es pielāgoju spēli, lai viņa strādātu ar vairākiem mērķiem un arī izveidoju precizitātes un punktu sistēmu, kur pēc spēles tev ir parādīts vērtējums un ja tev ir punktu rekords tas tiek saglabāts.

Pēc spēļu loga uztaisīšanas es palīdzēju Paulam ar viņu iestatījuma loga opcijām un palīdzēju tās integrēt spēlē. Pēc spēles uztaisīšanas es pārgāju cauri un pielaboju visus dokumentus un diagrammas, lai tās atbilstu tagadējam projektam.

## 7.2. Gustavs Pauls

Pirms projekta sākšanas es izdomāju, ka šo spēli taisīsim un arī izdomāju šīs spēles galveno principu. Rakstot PPS es rakstīju Ievadi, Vispārējo aprakstu un Nefunkcionālās prasības, kuras bija 1., 2. un 4. nodaļas dokumentā. Kamēr rakstot PPS man bija jāizdomā nefunkcionālās prasības priekš šī projekta un pēc PPS uzrakstīšanas es arī izgāju cauri Narvila funkcionālām prasībām un pārbaudīju vai viņas bija loģiskas un vai viņām nebija pareizrakstību kļūdas.

Rakstot PPA man bija jāraksta Ievads, Lietotāju saskare un Izstrādes rīki, kuras bija 1., 5. un 6. nodaļas dokumentā. Rakstot PPA mans uzdevums bija izdomāt, kā varētu izskatīties lietotāju saskare un arī kādiem citiem izstrādes rīkiem varētu veikt projektu realizēšanu. Kamēr veidojot PPA es arī mazlietiņ palīdzēju Narvilam ar aktivitāšu diagrammu izveidošanu un arī pēc tam izgāju cauri viņa nodaļām priekš pareizrakstības kļūdām.

Veidojot projektu mans galvenais uzdevums bija izveidot Galveno Izvēlni un Iestatījuma logus. Galvenā izvēlnē es vienkārši izveidoju visas pogas priekš ainu pāriešanas un arī izveidoju pogas dizainu kura tika izmantota caur visu spēli.

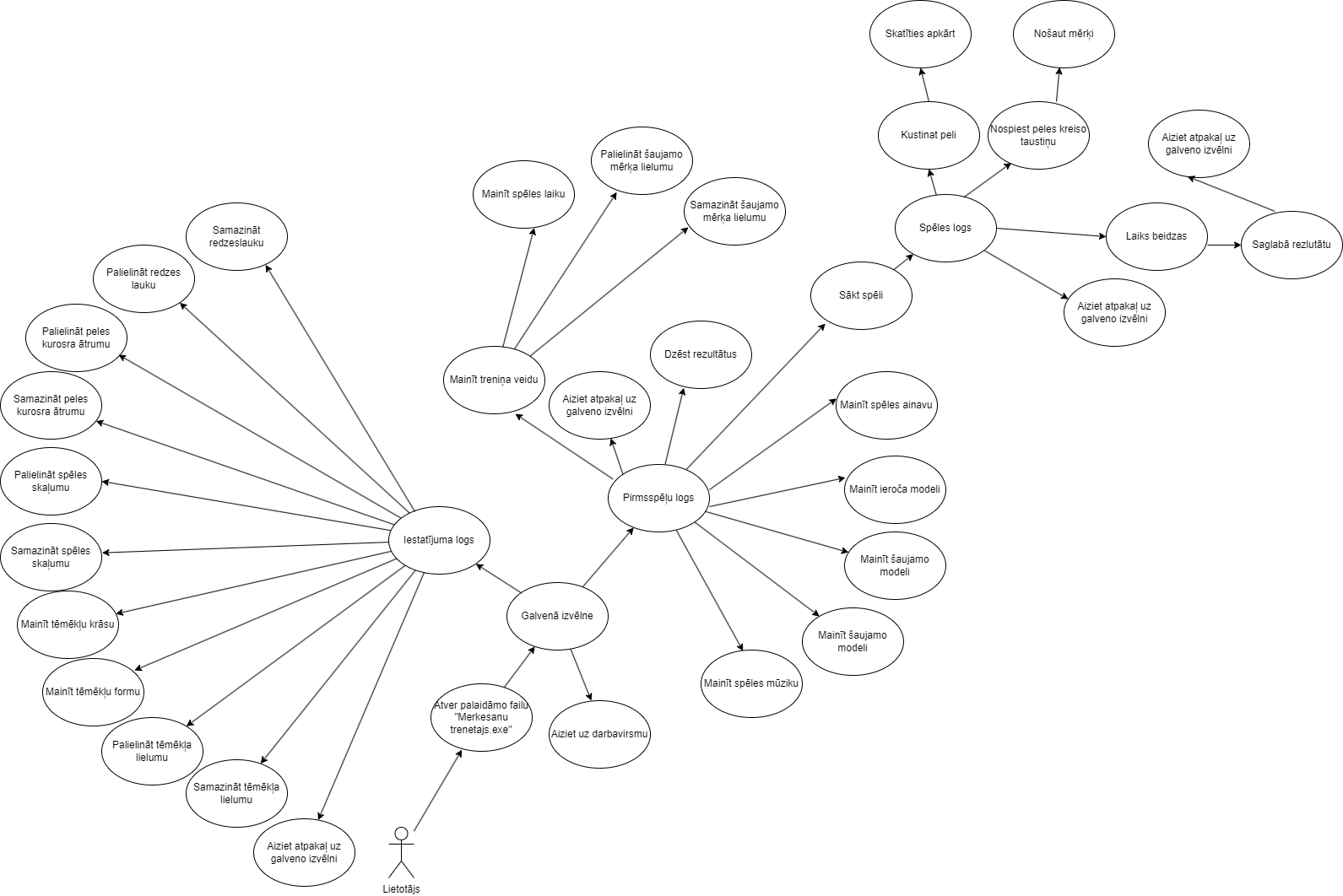
Pēc tam es izveidoju iestatījuma logu, kur es izveidoju rezolūcijas maiņu, peles ātrumu maiņu, redzeslauku maiņu un arī izveidoju skaļuma maiņu. Es arī atradu dizainu priekš visām tēmēkļu formām un izveidoju to maiņu, izmēra regulēšanu un krāsu maiņu ar trīs bīdņiem. Es arī izveidoju bīdņu dizainu, kuru izmantoja caur visu spēli.

Pēc savas daļas izveidošanas es palīdzēju Narvilam ar spēles testēšanu un palīdzēju atrast kļūdas kuras vel bija palikušas viņa kodā. Pēc spēles pabeigšanas es izgāju cauri Narvila rediģētiem PPS un PPA un pārskatīju tos priekš pareizrakstības kļūdām.

# Pielikumi

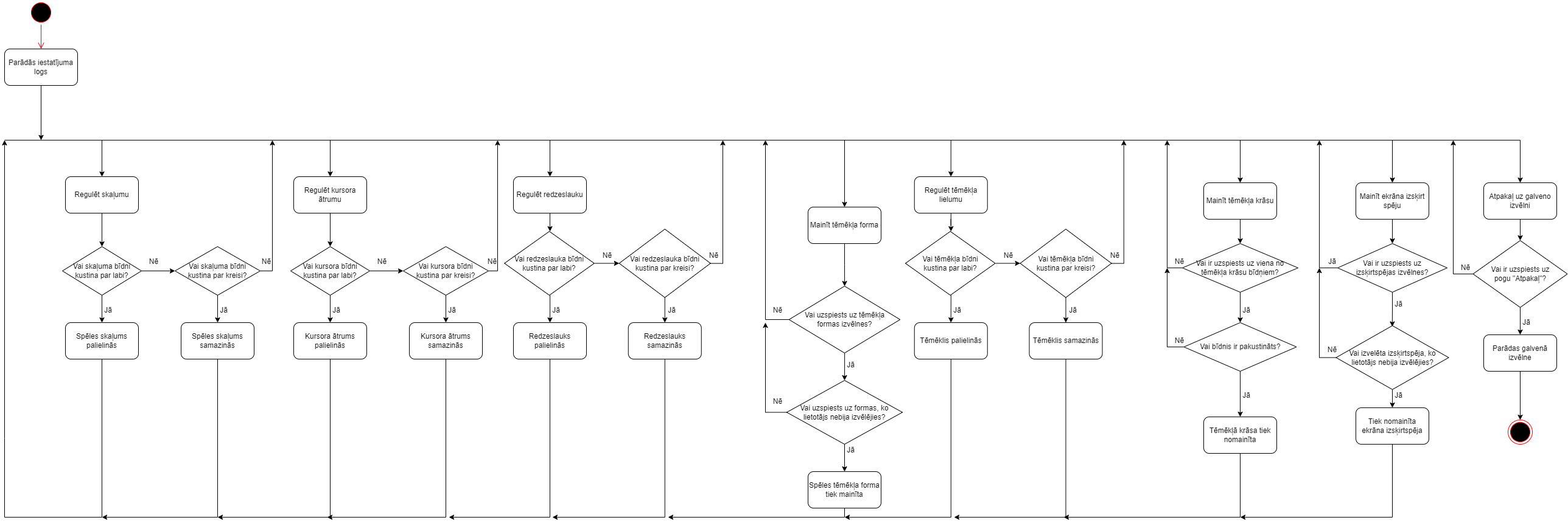
**1.Pielikums**

**USE CASE diagramma**



**2.Pielikums**

**Iestatījuma loga aktivitāšu diagramma**



**3.Pielikums**

**Pirmsspēles loga aktivitāšu diagramma**

