LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

PROGRAMMĒŠANAS KATEDRA

**SPĒLES “MEMEDIEVAL” MINIMĀLĀ DARBSPĒJĪGĀ PRODUKTA IZSTRĀDE**

KVALIFIKĀCIJAS DARBS

Autors: **Sīmanis Mikoss**

Studenta apliecības Nr.: sm11131

Darba vadītājs: B. dat. Ģirts Ķesteris

RĪGA 2018

ANOTĀCIJA

Līdz 850 rakstzīmēm, ieskaitot atstarpes. Izklāsta problēmas būtību, pētījuma mērķi, uzdevumus, iegūtos rezultātus

**Atslēgvārdi:** 4-8 atslēgvārdi

ABSTRACT

DEVELOPMENT OF THE MINIMUM VIABLE PRODUCT FOR THE GAME “MEMEDIEVAL”

Kaut kas

**Keywords:**

RÉSUMÉ

Kaut kas

Mots-clés:

SATURS

[IZMANTOTIE SAĪSINĀJUMI 6](#_Toc514855322)

[IEVADS 7](#_Toc514855323)

[1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA 8](#_Toc514855324)

[1.1. Vispārīgs apraksts 8](#_Toc514855325)

[1.2. Komandas mērķi 8](#_Toc514855326)

[1.3. Darbības sfēra 8](#_Toc514855327)

[1.4. Lietotājstāsti 9](#_Toc514855328)

[1.5. Lietotāja saskarne 10](#_Toc514855329)

[1.6. Ierobežojumi un pieņēmumi 10](#_Toc514855330)

[2. PROJEKTĒJUMA PRASĪBU APRAKSTS 11](#_Toc514855331)

[2.1. Funkcionalitātes apraksts 11](#_Toc514855332)

[2.2. Datu struktūru apraksts 12](#_Toc514855333)

[2.3. Lietotāja saskarnes apraksts 12](#_Toc514855334)

[3. PROGRAMMATŪRAS KODS 13](#_Toc514855335)

[4. TESTĒŠANAS DOKUMENTĀCIJA 15](#_Toc514855336)

[4.1. Iekšējā testēšana izstrādes laikā 15](#_Toc514855337)

[4.2. Daļēji atvērtā testēšana izstrādes laikā 15](#_Toc514855338)

[4.3. Iekšējā testēšana pēc izstrādes 15](#_Toc514855339)

[5. PROJEKTA ORGANIZĀCIJA 17](#_Toc514855340)

[5.1. Projekta dalībnieki 17](#_Toc514855341)

[5.2. Projekta organizācija 17](#_Toc514855342)

[5.3. Galvenās aktivitātes 19](#_Toc514855343)

[5.4. Komandas komunikācija 20](#_Toc514855344)

[5.5. Rezultāti 20](#_Toc514855345)

[6. KVALITĀTES NODROŠINĀŠANA 21](#_Toc514855346)

[6.1. Veiktie pasākumi 21](#_Toc514855347)

[6.2. Ieplānotie pasākumi 21](#_Toc514855348)

[7. KONFIGURĀCIJAS PĀRVALDĪBA 22](#_Toc514855349)

[7.1. Versiju kontrole 22](#_Toc514855350)

[7.2. Versiju numuri 22](#_Toc514855351)

[7.3. Direktoriju struktūra 22](#_Toc514855352)

[7.4. Prasību trasējamība 23](#_Toc514855353)

[7.5. Dokumentu konfigurācijas pārvaldība 23](#_Toc514855354)

[8. DARBIETILPĪBAS NOVĒRTĒJUMS 24](#_Toc514855355)

[SECINĀJUMI 24](#_Toc514855356)

[IZMANTOTĀ LITERATŪRA 25](#_Toc514855357)

[DOKUMENTĀRĀ LAPA 26](#_Toc514855358)

# IZMANTOTIE SAĪSINĀJUMI

**AI** – mākslīgais intelekts

IAP

**LU** – Latvijas Universitāte

**PPA** – programmatūras projektējuma apraksts

**PPS** – programmatūras prasību specifikācija

camelCasing

piezīme par saīsinājumiem, kas izmantoti atsevišķās nodaļās

# IEVADS

Kvalifikācijas darbs apraksta mobilas spēles projekta “Memedieval” minimālā darbspējīgā produkta izstrādi. Projekta pasūtītājs ir spēles izstrādātājs un kvalifikācijas darba autors (Sīmanis Mikoss).

Pirmais spēles prototips tika radīts 2017. gada septembrī hakatona “Game Camps Riga 2017” ietvaros četru cilvēku komandā. Jaunas prototipa iterācijas tika izveidotas “Startup Slalom IT Rēzekne” un “Startup Slalom Fintech” hakatonos un dažādi papildinājumi, labojumi un dokumentācija par tēmu radās grupu darbos studiju laikā.

Projekta vadītāja lomu līdz šim visos gadījumos ir uzņēmies kvalifikācijas darba autors. Skaņu efektu, mūzikas bet centrālais komandas sastāvs ir mainījies vairākas reizes.

Spēle ir veidota uz Unity dziņa bāzes un paredzēta lietošanai Android operētājsistēmas viedtālruņiem vai Windows darbstacijām. Versiju vadībai sākotnēji tika izmantots Unity piedāvātais risinājums “Unity Collaborate”, bet laika gaitā tika veikta pāreja uz GitHub.

Rakstīšanas brīdī spēles Memedieval prototips daļēji ilustrē ieplānoto spēles funkcionalitāti. Iespējams noalgot čempionus, kuriem sejas vietā tiek piešķirta spēlētāja uzņemta fotogrāfija. Čempioniem ir dažādas īpašības – prasmes (angliski – *skills*), biogrāfija, un personīgā devīze. Čempioni kalpo kā spēlētāja armijas komandieri un palīdz viņam iekarot pretinieku pilis. Spēlētāja pretinieks ir mākslīgais intelekts (turmpāk tekstā – *AI*).

Prototips līdz šim nav testēts, izmantojot formālu pieeju, bet ir apzinātas dažādas nepilnības, kas noteikti jānovērš jebkurā versijā, kas paredzēta publicēšanai.

Nākošā šķirtne spēles Memedieval izstrādē ir minimālā darbspējīgā produkta izveide palaišanai testa tirgū, kas arī ir kvalifikācijas darba mērķis. Ar minimālo darbaspējīgo produktu šajā gadījumā tiek saprasta spēles versija, kas atbilst programmatūras prasību specifikācijā (turpmāk tekstā – PPA) aprakstītajai.

Izmantotās pētniecības metodes?

Izmantotie avoti?

Darba struktūra?

# 1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

Prasību specifikācija (turpmāk tekstā – PPS) ir izveidota, vadoties pēc spējās izstrādes principiem [1]. Atsevišķās apakšnodaļās tā iekļauj vispārīgu projekta aprakstu, komandas mērķus un darbības sfēru. Prasības sīkāk aprakstītas lietotājstāstos. Nodaļas beigās uzskaitīti

## 1.1. Vispārīgs apraksts

Dalībnieki:

* projekta īpašnieks – Sīmanis Mikoss
* komanda – Sīmanis Mikoss (programmētājs, projekta vadītājs), Joakim Stenmark (audio dizains un implementācija)
* ieinteresētās puses – projekta īpašnieks, komanda

Statuss: prototipēšanas fāze

Termiņš: 2018. gada jūnijs

## 1.2. Komandas mērķi

Galvenais mērķis ir izveidot lietotnes versiju, kas būtu piemērota palaišanai testa tirgū. Mērķis tiks uzskatīts par sasniegtu, ja izveidotā versija atbildīs šai PPS.

## 1.3. Darbības sfēra

Videospēļu izstrāde.

## 1.4. Lietotājstāsti

*1.1. tabula*

**Lietotājstāstu pārskats**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **D** | **Nosaukums** | **Apraksts** | **Minimālās prasības** | **Prioritāte** |
| CH | Čempionu pārvalde | Es kā lietotājs vēlos dažādos veidos mijiedarboties ar sev padotiem čempioniem, lai iejustos varena komandiera ādā | Čempionus var:  CH1 – noalgot  CH2 – personalizēt  CH3 – uzlabot  CH4 – izsaukt kaujā  CH5 – komandēt kaujā  CH6 – atlaist  CH7 – zaudēt kaujā | obligāts |
| BA | Kaujas pārvalde | Es kā lietotājs vēlos komandēt sev padoto karaspēku un mijiedarboties ar kaujas lauku, lai panāktu uzvaru pār pretinieku un iegūtu ārēju apstiprinājumu savām prasmēm | Kaujā var:  BA1 – izsaukt karavīrus  BA2 – dot komandas  BA3 – zaudēt  BA4 – uzvarēt | obligāts |
| RE | Resursu pārvalde | Es kā lietotājs vēlos, lai man būtu iespēja iegūt un tērēt dažādus resursus, lai rīcība spēlē labāk simulētu uz realitāti balstītu fantāziju | Resursus var:  RE1 – iegūt  RE2 – apmainīt  RE3 – tērēt | vēlams |
| MA | Kartes pārvalde | Es kā lietotājs vēlos, lai mans progress un virzība spēlē tiktu attēlots uzskatāmā veidā | Kartē attēlotas:  MA1 – iekarotās pilis  MA2 – neiekarotās pilis  MA3 – nezināmās pilis | vēlams |

## 1.5. Lietotāja saskarne

## 1.6. Ierobežojumi un pieņēmumi

Projekta mērķis ir potenciāliem klientiem demonstrēt galveno produkta funkcionalitāti, tāpēc netiks izstrādātas, kas ir laikietilpīgas un/vai neilustrē, ar ko produkts atšķirtos no citiem, kas jau pieejami tirgū. Funkcijas, kas ir ārpus projekta mēroga, ietver spēles progresa sinhronizāciju mākonī, trešās puses autentifikāciju, un sejas vaibstu atpazīšanu.

# 2. PROJEKTĒJUMA PRASĪBU APRAKSTS

Projektējuma apraksts ir izveidots vēlā izstrādes stadijā, jo vairums funkcionalitātes tika ieviesta dažādu hakatonu un pasākumu ietvaros, kur laika un resursu ierobežojums noteica vajadzību iztikt ar minimālu dokumentāciju, liekot uzsvaru uz uzskatāmiem rezultātiem.

Līdz ar to, konkrētā PPA mērķis ir radīt vispārēju priekšstatu par principiem, pēc kuriem darba autors vadījās izstrādes laikā. Projektējumā nav iekļauti sistēmas moduļi, kas **nav tieši saistīti ar PPS** minētajiem lietotāja stāstiem.

## 2.1. Funkcionalitātes apraksts

– kādi ir galvenie funkcionālie moduļi, to nolūks un kā tie savstarpēji sadarbojas.

*2.1. tabula*

**Galvenie funkcionālie moduļi**

|  |  |  |
| --- | --- | --- |
| **Modulis** | **Galvenā funkcionalitāte** | **Lietotājstāsti** |
| Kameras izmantošanas modulis | Nodrošina iespēju uzņemt fotogrāfijas ar ierīces kameru, kuras var izmantot čempiona personalizēšanai | CH (CH1, CH2) |
| Kaujas modulis | Ļauj spēlētājam komandēt savu karaspēku, kontrolē pretinieka darbības | BA |
| Kartes modulis | Nodrošina, ka kartē pilis ir iezīmētas atkarībā no to stāvokļa, inicializē kaujas parametrus atkarībā no izvēlētās pils | MA |
| Čempionu modulis | Saglabā čempionu īpašības, nodrošina jaunu čempionu izveidi | CH |
| Resursu modulis | Saglabā spēlētājam piederošos resursus | RE |
| Spēlētāja profila modulis | Saglabā spēlētājam piederošos resursus un čempionus starp spēles līmeņiem | RE, CH |
| Datu glabāšanas modulis | Saglabā lietotājam piederošos resursus un čempionus ierīces atmiņā starp dažādām spēles sesijām | RE, CH |

## 2.2. Datu struktūru apraksts

– datubāzes vai citu datu struktūru projektējums.

Klases,

serializējamas klases

## 2.3. Lietotāja saskarnes apraksts

# 3. PROGRAMMATŪRAS KODS

Viss autora kods ir rakstīts C# valodā un pēc GitHub repozitorija datiem 97% no programmatūras koda ir C#.

Autora rakstīts varētu būt aptuveni 40-60% no koda. Tas neatspoguļojās GitHub repozitorija izpilžu uzskaitē, jo pēc tās autors ir iesūtījis ap 97% no visām izpildēm. Iemesls neatbilstībai ir tāds, ka sākotnēji projektā tika izmantota cits versiju kontroles risinājums (Unity Collaborate), no kura tika veikta pāreja uz GitHub pēc tam, kad Unity Collaborate kļuva par maksas pakalpojumu.

Nodaļas rakstīšanas brīdī spēlē ir 70 dažādas klases, no kurām svarīgākās ir uzskaitītas tabulā 3.1.

Kods vairākās vietās nav pārskatāms un neatbilst labākajai praksei, kas izriet no projekta specifikas un līdzšinējās izstrādes gaitas – liela daļa programmēšanas darbu tika veikti hakatonos komandās, kuru dalībnieki iepriekš nebija sadarbojošies un bija ar ļoti atšķirīgiem zināšanu līmeņiem. Tā kā šādos pasākumos ir stingri ierobežots laika limits un uzsvars ir uz demonstrējamas funkcionalitātes izveidi, programmēšanas stilam un veidam tika pievērsta maza uzmanība.

Tomēr turpinot projekta attīstību ārpus hakatoniem, jaunākās funkcijās ir ieviesta sakārtotāka pieeja:

* mainīgie tiek definēti, izmantojot camelCasing (piem. unitCost);
* klasēm un metodēm tiek izmantots PascalCasing (piem. ChampionSkillGenerator);
* mainīgie, klases, metodes, komentāri u.c. koda elementi ir angļu valodā.

Tāpat ir mēģināts kodu padarīt pēc iespējas lasāmāku, jaunas metodes rakstot tā, lai tās ir viegli saprotamas vai tās komentēt gadījumā, ja tas nav iespējams.

Ņemot vērā, ka

*3.1. tabula*

**Svarīgākās klases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Klase** | **Galvenā Funkcionalitāte** | **Saistītais modulis** | **Lietotājstāsts** |
| GameData |  |  |  |
| ChampionData |  |  |  |
| Champion |  |  |  |
| PlayerProfile |  |  |  |
| Map |  |  |  |

# 4. TESTĒŠANAS DOKUMENTĀCIJA

Spēles testēšana tika veikta trīs dažādos projekta izstrādes posmos:

1. Funkciju izstrādes laikā;
2. Prezentāciju laikā;
3. Pēc funkciju izstrādes.

## 4.1. Iekšējā testēšana izstrādes laikā

Izstrādes laikā spēle tika testēta izstrādes, gan testa vidē, kas attiecīgi ir Unity programma un Android operētājsistēmas viedtālrunis/viedtālruņi. Šajā posmā pārsvarā tika izmantota ad-hoc jeb brīva testēšana, kas ir neplānota veida pieeja, un vienībtestēšana, kas tika veikta ar mērķi pārliecināties par jaunieviestu funkciju pareizu darbību atsevišķi vai kopā ar iepriekš izstrādātām funkcijām.

## 4.2. Daļēji atvērtā testēšana izstrādes laikā

## 4.3. Iekšējā testēšana pēc izstrādes

|  |  |
| --- | --- |
|  |  |
|  |  |

Play testing

Ir izstrādāta testēšanas dokumentācija, kas apliecina pretendenta patstāvīgi veiktu vienībtestēšanu

<https://en.wikipedia.org/wiki/Game_testing>

<https://www.gamasutra.com/view/feature/132398/quality_quality_assurance_a_.php>

Videospēļu izstrādē tiek izmantots salīdzinoši maz automatizēto testu.

<https://www.testbytes.net/blog/types-of-game-testing/>

The most popular test automation frameworks for Android testing:

* Appium
* Calabash
* Robotium
* uiautomator
* Espresso

# 5. PROJEKTA ORGANIZĀCIJA

Spēles izstrādes organizācija neatbilst standarta projektam. Tā tika iesākta hakatona ietvaros četru cilvēku komandā un tika turpināta pēc autora iniciatīvas bez stingri noteiktiem termiņiem vai atvēlētiem resursiem, vienīgi ar mērķi produktu attīstīt līdz tas nezināmā nākotnē būs piemērots palaišanai tirgū.

Tomēr konkrēti termiņi un prasības kļuva aktuāli, spēles izstrādi iesaistot dažādos hakatonos un mācību projektos. Lai kvalificētos minētajos pasākumos, bieži bija nepieciešams piesaistīt papildu komandas biedrus, kuri līdz ar pasākuma beigām izbeidza aktīvu iesaisti projektā.

Gan pasākumi, kuros projekts tika attīstīts, gan komandas biedri, kuri tajos piedalījās, tiks precizēti turpmākajās apakšnodaļās. Tāpat šajā nodaļā tiks izklāstīti projekta organizācijas principi un komandas galvenās aktivitātes un komunikācijas veidi. Nodaļas noslēgums būs veltīts kopsavilkumam par projekta rezultātiem un iespējamo nākotni.

## 5.1. Projekta dalībnieki

Projekta pasūtītājs ir Sīmanis Mikoss (SM). Tā kā spēlē šobrīd nav pieejama, tai nav aktīvu lietotāju.

Galvenie izstrādātāji ir SM un Joakim Stenmark (JS). SM galvenie pienākumi ir projekta vadība, programmēšana un satura izstrāde, savukārt JS nodrošina spēles mūziku un skaņu efektus. Citi izstrādātāji ir uzskaitīti tabulā 5.1 kolonnā “Komanda” (sk. XX lpp.).

## 5.2. Projekta organizācija

Darba organizācija projekta gaitā mainījās, lai pielāgotos ārējām prasībām, kas izrietēja no projekta iesaistes dažādos hakatonos un studiju kursos. Šīs izmaiņas galvenokārt izpaudās jaunu komandas biedru piesaistē, precīzāku termiņu un mērķu noteikšanā.

Izstrādes posmus raksturoja ierobežota un neformāla dokumentācija, izmantojot piemēram Google Sheets vai Trello. Izņēmums bija studiju kursi, bet šajos gadījumos dokumentācija bija ārēji noteikta prasība.

Procesi kopumā atbilda spējās izstrādes manifestā paustajiem principiem, bet apzināti netika izvēlēts precīzs metodoloģisks ietvars kā piemēram SCRUM vai Kanban. Vairums veicamie uzdevumi tika definēti konkrētā posma sākumā un deleģēti vienam vai vairākiem komandas biedriem. Atkarībā no apkārtējām izmaiņām, veicamo uzdevumu saraksts tika papildināts vai modificēts. Visā projekta laikā komandas organizācijas struktūra bija horizontāla ar vienu vadītāju.

*5.1. tabula*

**Pārskats par projekta izstrādes nozīmīgākajiem posmiem**

|  |  |  |  |
| --- | --- | --- | --- |
| **Posms** | **Komanda** | **Mērķi** | **Galvenie rezultāti** |
| **“Game Camps Rīga 2017”**  Hakatons par spēļu izstrādi  (29.09.2017. - 01.10.2017.) | * SM * JS * Silver Kirotar – programmēšana * Jekaterina Aleinika – māksla | Pirmā spēles prototipa izstrāde | Izveidots prototips, kas ļauj:   * iekarot vienu pili * izmantot viedtālruņa/ datora kameru, lai mainītu čempiona izskatu |
| **“Startup Slalom IT Rezekne”**  Hakatons par informāciju tehnoloģijām  (27.10.2017. - 29.10.2017.) | * SM * Gundega Strautmane – māksla * Kaspars Vogulis – programmēšana * Valdis Platonovs –programmēšana * Edgars Judovičs – programmēšana | Čempionu sistēmas uzlabošana | Čempioniem pievienotas dažādas īpašības:   * vārds * dzimums * biogrāfija * prasmes *(skills)* |
| **“Startup Slalom FinTech”**  Hakatons par finanšu tehnoloģijām  (10.11.2017. - 12.12.2017.) | * SM * Artis Rozentāls – programmēšana * Valērija Vanaga – māksla * Leo Davidov – māksla * Ģirts Ķesteris – IAP prototipa izstrāde | IAP prototipa izstrāde | Spēlei pievienoti divi resursi (dukāti un sāls), izveidots IAP prototips |
| **LU kurss “Programminženierija”**  (2017.g. septembris – 2018.g janvāris) | * SM * Anete Katrīna Logina * Laura Šakale * Zane Nāgele * Anastasija | Dokumentācijas izstrāde, demonstrējama prototipa izstrāde. | Izstrādāts PPS, PPA, Lietotāja ekrāna formu projektējums) |
| **LU kurss**  **“Uzņēmējdarbība”**  (2017.g. septembris - 2018.g janvāris) | * SM * Anete Katrīna Logina – nolaišanās lapas izstrāde * Maksims Sņežkins * Kaspars Neiders * Lauris Lauks | Minimālā darbspējīgā produkta izstrāde, pirmo lietotāju iegūšana, nolaišanās lapas izstrāde. |  |
| **Ludum Dare 41 Tartu**  Spēļu izstrādes hakatons (20.04.2018. - 22.04.2018.) | * SM * JS * Fredy Schmidt – audio dizains | Čempionu īpašo spēju izstrāde. |  |
| **Brīvais izstrādes posms**  Viss spēles izstrādei veltītais laiks ārpus dažādiem pasākumiem un studijām  (25.09.2017. - …) | * SM * JS * Renāte Logina – māksla * Līga Irbe Mikosa – māksla | Minimālā darbspējīgā produkta izstrāde |  |

## 5.3. Galvenās aktivitātes

Aktivitātes, kas tika veiktas projekta ietvaros, ir iedalītas divās kategorijās: “Izstrāde” (1) un “Organizācija un Komunikācija” (2).

Galvenās aktivitātes no pirmās kategorijas ir:

1. spēles dizains *(game design)* – spēles elementu definēšana, dažādu vērtību balansēšana ar mērķi uzlabot lietotāja pieredzi, dažādu izstrādāto materiālu (kods, vizuālie materiāli, skaņu efekti) ieviešana spēlē vai prototipā;
2. prototipēšana;
3. programmēšana (C#);
4. iekšējā testēšana;
5. vizuālo materiālu izstrāde;
6. skaņu efektu un mūzikas izstrāde;
7. cita veida satura (dažādi teksti, spēles sižets, u.c.) izstrāde.

Galvenās aktivitātes no otrās kategorijas (Organizācija un Komunikācija) ir:

1. pārrunas – dažādas diskusijas par darbu deleģēšanu, projekta virzību un citiem aktuāliem jautājumiem;
2. plānošana – veicamo uzdevumu definēšana, laika menedžments. Vairums gadījumu katrs plānošanu veic individuāli savām vajadzībām;
3. prezentācijas – prototipu un ideju prezentēšana komandas biedriem vai cilvēkiem ārpus tās ar mērķi informēt par projektu vai pārliecināt;
4. dokumentācija – ieplānoto izmaiņu un uzdevumu, konstatēto blusu, īstenotu funkciju un citu faktu piefiksēšana elektroniskā vai papīta formā.

## 5.4. Komandas komunikācija

kā notika komunikācija projekta dalībnieku starpā?

Hakatonos komunikācija notika klātienē

Komunikācija tika organizēta caur:

FaceBook

Trello

Google Drive

## 5.5. Rezultāti

Vai izstrādātā programmatūra ir akceptēta no pasūtītāja / lietotāja puses, vai tā ir vai būs produkcijā?

Ja tiek rakstīts, ka projekts notika pēc spējās (agile) pieejas, tad arī citās sadaļās jāparādās konkrētām spējās programmatūras izstrādes praksēm (piemēram, lietotāju stāsti, fiksēta garuma iterācijas, testu virzīta izstrāde u.tml.).

# 6. KVALITĀTES NODROŠINĀŠANA

Kvalitātes nodrošināšanai videospēļu industrijā var nošķirt trīs kategorijas [2 https://www.theseus.fi/bitstream/handle/10024/94503/Ruuska\_Essi.pdf?sequence=1]:

1. Funkcionalitātes nodrošināšana – lai panāktu, ka spēle darbosies kā iecerēts;
2. Lokalizācijas kvalitātes nodrošināšana – lai panāktu, ka spēlē izmatotie teksti un to tulkojumi ir salasāmi un cilvēkam saprotami;
3. Platformas sertifikācija – lai panāktu, ka spēle atbilst konkrētās platformas standartiem.

Visas trīs kategorijas kā galveno kvalitātes nodrošināšanas metodi izmanto dažādu veidu testēšanu. Projektam vissvarīgākā ir funkcionalitātes nodrošināšana, bet otrā kategorija ir salīdzinoši mazsvarīga, jo spēle ir vienīgi angļu valodā. Platformas sertifikācija ir īpaši svarīga brīdi, kad spēle tiek publiskota uz kādas izplatīšanas platformas, piemēram Google Play.

Turpmākajās apakšnodaļās uzskaitīti konkrētie kvalitātes nodrošināšanas pasākumi, kas veikti un kurus vēlams veikt, ja projekta izstrāde tiks turpināta ārpus kvalifikācijas darba ietvariem.

## 6.1. Veiktie pasākumi

Programmatūras kvalitātes

## 6.2. Ieplānotie pasākumi

Funkcionalitātes nodrošināšana

Slēgta beta versija.

Atvērta beta versija.

Atvērta saziņa ar lietotājiem.

Platformas sertifikācija

# 7. KONFIGURĀCIJAS PĀRVALDĪBA

Nodaļā aprakstītas galvenās metodes, ar kuru palīdzību tiek organizēta projekta koda uzturēšana, atjaunošana, dublēšana, kā arī nodrošināta programmatūras atbilstība prasībām. Galvenās metodes iekļauj versiju kontroles risinājumus, versiju numerāciju un direktoriju organizāciju. Nodaļas beigās paskaidrots, kā tiek veikta konfigurācijas pārvaldība projekta dokumentācijai.

## 7.1. Versiju kontrole

Versiju kontrolei tiek izmantots GitHub kombinācijā ar Git Bash un GitHub Desktop. Nodaļas rakstīšanas brīdi repozitorijam ir pieeja pieciem cilvēkiem, ieskaitot darba autoru. Visiem pieciem ir aizgādāšanas tiesības un darba autoram ir arī administratora tiesības.

Projekta sākumā tika uzmantots Unity piedāvātais risinājums versiju kontrolei – Unity Collaborate, jo tas uz laiku bija pieejams par velti.

## 7.2. Versiju numuri

Sākot ar 2018. gada martu, spēles būvējumiem tiek piešķirti versiju numuri, kas sastāv no trīs cipariem. Pēdējais versijas numurs nodaļas rakstīšanas brīdi ir 0.1.3.

Pēc noklusējuma katram nākošajam būvējumam pēdējais cipars ir par vienu augstāks, līdz tas sasniedz deviņi vai arī konkrētajā būvējumā ir ieviestas pietiekami būtiskas izmaiņas, lai pamatotu otrā cipara palielināšanu. Tas nozīmē, ka nākošais versijas numurs pēc 0.1.3. var būt gan 0.1.4., gan 0.2.0.

Galvenās izmaiņas, kas veiktas katrā versijā tiek reģistrētas izmaiņu žurnālā (sk. Pielikumu).

## 7.3. Direktoriju struktūra

Projekta galvenās direktorijas struktūra atbilst noklusējuma struktūrai Unity dziņa projektiem. Gandrīz visas datnes, ko projektam pievienojuši komandas darbinieki, atrodas direktorijā *Assets*. Šīs datnes sakārtotas pa dažādām apakšdirektorijām atkarībā no to funkcijām un failu tipiem.

Galvenās *Assets* apakšdirektorijas ir apskatāmas 7.1. tabulā. Direktorijas, kuras ir automātiski izveidotas dažādu Unity spraudņu darbības rezultātā, tabulā nav uzskaitītas.

*7.1. tabula*

**Direktorijas *Assets* galvenās apakšdirektorijas**

|  |  |
| --- | --- |
| **Nosaukums** | **Saturs** |
| Animations | Sagatavotu animāciju faili un animāciju kontrolieri |
| Audio | Audio faili |
| Fonts | Fonti, kas tiek izmantoti spēlē |
| Plugins | Dažādi projektam pievienotie spraudņi |
| Prefabs | Iepriekš sagatavoti objekti izmantošanai spēlē |
| Scenes | Unity objekti, kas reprezentē spēles ainas (t.i. spēles līmeņus) |
| Scripts | Komandas rakstītie skripti, kas nodrošina spēles funkcionalitāti |
| Sprites | Vizuālie materiāli |

## 7.4. Prasību trasējamība

Ņemot vērā projekta specifiku, izstrādes sākuma posmā nebija nepieciešamība (un hakatonu gadījumā – iespēja) izmantot īpašus identifikatorus katrai prasībai, kas palīdzētu trasēt katru prasību no tās definēšanas līdz izstrādei, testēšanai un apstiprināšanai.

Tomēr šī kvalifikācijas darba ietvaros projektam tika definētas prasības un to identifikatori, kas precizēti PPS.

## 7.5. Dokumentu konfigurācijas pārvaldība

PPS, PPA un testēšanas dokumentācija tiek glabāta vienā elektroniskā dokumentā ar manuāli izveidotiem dublējumiem iekš Google Drive un projekta Github repozitorija. Izmaiņas un labojumi šajā dokumentācijā netiek speciāli reģistrētas.

Dokumentācija, kas tieši neizriet no kvalifikācijas darba prasībām (dažādi projektējumi, darāmo uzdevumu reģistrs, veikto izmaiņu žurnāls, papildu informācijas avoti, u.c.) tiek glabāti iekš Google Drive, slēgtā FaceBook grupā vai reģistrēti Trello. Vairums no šiem dokumentiem nav dublējumu, jo tie galvenokārt nepieciešami iekšējai organizācijai un ir viegli aizstājami vai nenozīmīgi.

# 8. DARBIETILPĪBAS NOVĒRTĒJUMS

Darbietilpības novērtējums tika veikts projekta izstrādes beigu posmā.

Commitu skaits

Versiju skaits

Implementētās funkcijas

# SECINĀJUMI

# IZMANTOTĀ LITERATŪRA

1. D. Radigan, “Product requirements documents, downsized,” *Atlassian Agile Coach* [tiešsaiste]; – [atsauce 19.04.2018.]. Pieejams: https://www.atlassian.com/agile/product-management/requirements

* [2 https://www.theseus.fi/bitstream/handle/10024/94503/Ruuska\_Essi.pdf?sequence=1]
* <http://agilemanifesto.org/>
* <https://en.wikipedia.org/wiki/User_story>
* <https://www.gamasutra.com/blogs/JasonWoodward/20140519/218078/User_stories_How_to_maintain_structure_with_creativity.php>
* <http://blog.agilegamedevelopment.com/2016/04/user-story-mapping-for-games-example.html>
* http://www.modernanalyst.com/Resources/Articles/tabid/115/ID/923/Requirements-Specifications-on-Agile-Projects.aspx

Resursi:

Izmaiņu žurnāls versijām

Repozitorijs: pēc pieprasījuma

# DOKUMENTĀRĀ LAPA