



Liepājas Valsts tehnikums

“SyberShock” videospēles

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Personālā kvalifikācija:

Programmēšanas tehniķis

Grupas nosaukums:

4PT

Projekta izstādātājs: Kārlis Melderis

Eksāmena datums 2024. gada 20. jūnijs

Liepāja 2024

Saturs

Saturs	2
Ievads	5
1. Uzdevuma formulējums	6
2.1. Produkta perspektīva	7
2.2. Sistēmas funkcionālās prasības	8
2.2.1. Izvēlēties līmeni poga.....	8
2.2.2. Iestatījuma izvēles atvēršanas poga.....	8
2.2.3. Iziēt poga	8
2.2.4. Izvēlēties grūtības iestatījumu logs	8
2.2.5. Izvēlēties grūtības iestatījumu	9
2.2.6. Sākt spēli poga.	9
2.2.7. Pauzēšana	10
2.2.8. Atsākt spēli poga	10
2.2.9. Iziēt no līmeņa	10
2.2.10. Lasīt līmeņa datus priekš spēles sākšanas	11
2.2.11. Pārskatīt visu līmeņu datus.....	11
2.2.12. Lādēšanās ekrāns.....	11
2.2.13. Spēles laikā nospiesta kāda no pozīcijas pogām	12
2.2.14. Spēles noteikto bultu ielādēšana	13
2.2.15. Īsā klikšķa bultiņa.....	13
2.2.16. Garu spiedienu bultiņa	13
2.2.17. Fotogrāfijas un tekstūras apstrāde mainīšanas gadījumā.	14
2.2.18. Sākulapas kursora sekojos aizmugures lauks.....	14
2.3. Sistēmas nefunkcionālās prasības	15
2.3.1. Veikt spējas prasības	15
2.3.2. Vizuāls izskats.....	15
2.3.3. Skaņu dizains.....	15

2.3.4. Modificējama	15
2.4. Gala lietotāja raksturiesīmes	15
3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums.....	16
3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts	16
3.1.1 Izstrādes līdzekļi.....	16
3.1.2 Dizaina rīki	16
3.1.3 Datu glabāšanas formatēšanā	17
3.2. Iespējamo risinājuma līdzekļu un valodu apraksts.....	17
3.2.1 Izstrādes līdzekļi.....	17
3.2.2 Dizaina rīki	18
3.2.3 Datu glabāšanas formatēšanā	18
3.2.4 Datu glabāšanas modifikācija.....	18
4. Sistēmas modelēšana un projektēšana.....	19
4.1. Sistēmas struktūras modelis	19
4.2. Klašu diagramma.....	20
4.3. State diagramma	21
4.4. Algoritma diagramma	23
4.5. Scenāriju diagramma.....	24
5. Lietotāju ceļvedis	25
5.1 Sākumlapa	25
5.2 Iestatījumi.....	26
5.3 līmeņu izvēle	28
6. Testēšanas dokumentācija	29
6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums	29
6.2. Testpiemēru kopa	29
6.3. Testēšanas žurnāls	31
7. Individuālais ieguldījums	41
7.1. 2D mākslas pieredze.....	41

7.2 3D mākslas pieredze.....	41
7.3 Unity pieredze	41
7.4 C# pieredze.....	41
7.5 Vispārējā pieredze	41
8. Secinājumi	42
9. Lietoto terminu un saīsinājumu skaidrojumi.....	43
10. Literatūras un informācijas avotu saraksts	44
Mākslinieki.....	44
Pielikumi	45

Ievads

Šis ievads ir domāts, lai iedotu vienkāršu aprakstu par “SyberShock” dokumentāciju, lai gan lielākā daļa informācijas par videospēli būs aprakstīta uzdevuma formulējuma sadaļā.

Specifikācijas nodaļa ir domāta, lai izveidotu labāku saprašanu par šo projektu, spēles funkcionalitāti un spēles vizuālo dizainu. Lai to dokumentētu tas tiks izskaidros funkcionālā, nefunkcionāla daļā un kopējas perspektīvs par šo projektu.

Izstrādes līdzekļu nodaļā tiks izskaidrots kādus instrumentus un rīkus es lietošu, lai izveidotu šo videospēli, kā piemēram kuru video spēļu dzini, 3d modelēšanas rīkus, 2d modelēšanas un jebkurus papildinājumus kurus es lietoju klāt pie tiem.

Sistēmas modelēšanā un projektēšanā tiks izskaidrots manējās spēles datu bāzes uzbūvi un struktūru kura būs demonstrēta izmantojot ER diagrammu. Lai demonstrētu videospēles funkcijas un kontroles tiks izmantots funkcionālais un dinamiskais sistēmas modelis. Aktivitāšu diagramma tiks izmantota, lai parādītu funkcionāli solim pa solim kā videospēle tiks organizēta. Lietojum gadījumu diagramma (Use Case) tiks izmantota, lai reprezentētu kā spēlētājs un modifikators mainīs ar videospēli. Sistēmas moduļu apraksts un algoritmu shēmas sadaļā tiks izskaidrots dziļāk kā strādā funkcijas kuras tika pieminētas iepriekšējās diagrammās.

Lietotāja ceļvedis, tiek rakstīts, lai dotu instrukcijas kā pārvietoties cauri šīs videospēles iestatījumiem un līmeņiem. Kā arī kā modificēt to priekš tekstūras, mūzikas un jaunu līmeņu ieviešanai.

Projekta veidošanas laikā tiek dokumentēti visas kļūdas un problēmas kas izveidojas projekta taisīšanas laikā, kā arī tiks dokumentēti kādi instrumenti kuri tiek izmantoti, lai atrastu, labotu un dokumentētu šīs kļūdas.

1. Uzdevuma formulējums

Šī projekta mērķis ir, lai izveidotu ritma videospēli cyberpunk stilā ar 3d neon krāsas vizuāliem, dinamiski mainās un atskaņo kopā ar mūziku, lai veidotu unikālu spēles sajūtu.

Galvenais mērķis ir izveidot ritmu spēli kura saprot kas padara citas ritmu spēles jautras un interesantas, lai gan vienlaicīgi, kas ir viņu vājākās daļas un uzlabot šobrīd pieņemto normu. Pat ja mērķis ir pārveidot šo žanru, šīs spēles mērķis nav to pārtaisīt uz kaut ko jaunu, un uzturēt ritmu spēles jautrību, interesi un iemīļoto sajūtu.

SyberShock programmatūras arhitektūrā prioritāte ir modularitāte, lai veicinātu trešās puses iesaistīšanos, izmantojot modifikācijas. Aizmugursistēmas dizaina vienkāršības mērķis ir nodrošināt optimālu veiktspēju, koncentrējoties uz preventīvu dizainu, lai izvairītos no iespējamām kļūdām. Kā piemēru, ja lietotājs nomaina līmeņus, izdzēs, vai modificē tās asētus programmai ir jāsaprot tas un jāturpina strādāt.

Izstrādes process sākas ar cyberpunk stilu un tēmu, kā mūzikas stilu konceptuālu izpēti un piemēru atrašanu. Tālāk seko prototipu izstrāde, lai pārbaudītu spēles galveno mehānismu un iekļautu lietotāju atsauksmes, tiks izveidota funkcionālā versija ar minimāliem vizuāliem, bet galveno spēļu ritmu. Vizuālo efektu versiju kurā ir vizuālie aspekti, bez vairākiem funkcionāliem aspektiem, lai pārbaudītu dizainu un stilu. Mākslinieciskā attīstības vienkāršums un skaņas dizains, ārpus mūzikas, ir jāizveido, lai darbojas vienlaikus, lai izveidotu “cyberpunk” gaisotni. Lai gan veidojot šādu gaisotni aizņem daudz eksperimentāciju. Pēdējās fāzēs tiek uzsvērti kļūdu labošana, optimizācija un stratēģiskā izlaišana ar pastāvīgu atbalstu, izmantojot atjauninājumus un kopienas iesaisti.

Spēles CyberShock tehnoloģiskā infrastruktūra ir pielāgota pašreizējiem veiktspējas standartiem, bet arī piemērota nākotnei un vecākām ierīcēm, lai nodrošinātu saderību ar jaunām un vecām platformām. Tas ietver apsvērumus par pielāgojamību mainīgajām un nozares standartu ievērošanu, nodrošinot, ka spēle joprojām ir pieejama un atbilstoša tehnoloģijai progresējot. Lietotāja vieglā modificējamība ir grūti pievienojama pie vairākām ierīcēm, kā piemēram Apple telefoniem un konsolēm, lai gan šī dokumentācija ir izveidota, lai demonstrētu datora versiju, tādēļ konsoles pieejamība netiks pieminēta tālāk.

Kopumā šī spēle ir līdzīga citām ritmu spēlēm, lai gan dēļ unikālā stila, jaunām kontrolēm, plāniem izveidot pieejamu visām ierīcēm un 3d vizuāliem. Šis projekta mērķis ir liels, bet dēļ šī unikāluma ir liela cerība ka ritmu spēles fani un lietotāji kuri nekad nav spēlējuši šāda stila spēli varēs to izbaudīt.

Šī specifikācija ir domāta, lai izskaidrotu gala mērķi un domu gājienu kā funkcionēs, un izskatīsies šī spēle. Šī nodaļa ir sadalīta 4 daļās. Produkta perspektīvs, kurā ir aprakstīta spēles funkcijas kopumā. Sistēmas funkcionālās prasības tiks izskaidrota spēles funkcijas un spēlētāja ievade. Nefunkcionālās prasībās tiks izskaidrota katra daļa no projekta, kas nav funkcija, kā piemēram izskats, programmēšanas valoda.

2.1. Produkta perspektīva

No sākuma es izskaidrošu šī projekta spēli. Šī ir ritmu spēle, tādēļ seko ierasto ritmu spēles formātu, kā spēlētājs tavš uzdevums ir nospiegt un turēt pogas ritmiski ar spēles mūziku, ar vairākiem līmeņiem un grūtības iestatījumiem. Šie līmeņi izmaina kura mūzika tiek spēlēta, un vizuāli kā izskatās spēle. Kamēr grūtības iestatījumi nomaina cik daudz pogas un cik ātri ir jāspiež ritmā ar mūziku.

Sāksim ar UI un ģenerālu izkārtojumu pirms sākas paša spēle. No spēles sākumlapas vajadzētu būt ļoti vienkāršām 3 pogām. Pogu, lai sāktu spēli, pogu, lai nomainītu iestatījumus un, lai izietu no programmas. Iestatījumu poga vienkārši atvērs līdzīgu apskatu ar iestatījuma pogām un slīdņiem. Lai gan “sākt spēli” poga rādīs visus iespējamus līmeņus, kuros varēs redzēt mūzikas nosaukumu un albūma foto vai mākslas darbu. Nospiežot uz šī līmeņa parādīs izvēli nomainīt grūtumu un spēju spēlēt šo līmeni.

Pēc līmeņa izvēles un spēles sākšanas atveras īss sagatavošanās periods kur var redzēt pašu spēli, bet vel nesākas mūzika un līmenis. Šī gaidīšana ir domāta, lai sagatavotu spēlētāju un lēnām iesākt līmeni. Šīs spēles visi līmeņi izskatās līdzīgi, ar 3 ceļiem centrā un 3d vizuāliem aspektiem apkārt. Spēles gaitā, ārpus no spēlētāja redzes lauka nokrīt instrukcionālas bultiņas kuras korespondē ar kurām pogām ir lietotājam jāspiež. Kad šīs instrukcionālas bultiņas nokrīt pie ekrāna apakšas, kur ir atzīmēta līnija. Kad šīs nokrīt ir jābūt muzikāla instrumenta skaņa vai vienkārši jābūt ritmā ar mūziku. Tad lietotāja objektīvs ir nospiegt, vai turēt šīs korespondējošās pogas. Ja pārāk daudzas bultiņas pāriet pāri atzīmētai līnijai, tad samazinās dzīvība spēle beidzas kad vai nu mūzika beidzas vai lietotāja dzīvība beidzas. Jebkurā laikā var paust un mainīt iestatījumus vai iziet no līmeņa. Lai gan tavš progress līmenī netiks saglabāts.

2.2. Sistēmas funkcionālās prasības

2.2.1. Izvēlēties līmeni poga

Mērķis:

Funkcija atver jaunu izvēlni ar visiem līmeņiem.

Ievaddati:

- 1) Peles kursors tiek nospiests uz šīs pogas.
- 2) līmeņa nosaukumi
- 3) līmeņa grūtuma līmeņi

Nosacījumi:

- 1) Poga ir redzama tikai sākumlapā.

Apstrāde:

Pēc nospiešanas lietotājs tiek pārnests uz spēles līmeņa izvēlnes.

2.2.2. Iestatījuma izvēles atvēršanas poga

Mērķis:

Funkcija atver jaunu izvēlni ar visiem iestatījumiem.

Ievaddati:

- 1) Peles kursors tiek nospiests uz šīs pogas.

Nosacījumi:

- 1) Poga ir redzama tikai sākumlapā un pauzēšanas sadaļā.

Apstrāde:

Pēc nospiešanas lietotājs tiek pārnests uz Iestatījumu izvēlnes.

2.2.3. Iziēt poga

Mērķis:

Funkcija lai aizvērtu aplikāciju.

Ievaddati:

- 1) Peles kursors tiek nospiests uz šīs pogas.

Nosacījumi:

- 1) Poga ir redzama tikai sākumlapā.

Apstrāde:

Pēc nospiešanas aplikācija tiek apstādināta.

2.2.4. Izvēlēties grūtības iestatījumu logs

Mērķis:

Pēc nospiešanas uz konkrēta līmeņa tiek parādīta grūtības izvēle, pop-up logā.

Ievaddati:

- 1) Peles kursorš nospiež uz jebkuru līmeni.
- 2) Konkrētā līmeņa noteiktie dati JSON faila.

Nosacījumi:

- 1) Tiek parādīti tikai grūtības līmeņi kuri ir pieejami konkrētam līmenim.
- 2) Parādās tikai spēles izvēlnes sadaļā.

Apstrāde:

Pēc nospiešanas parādās visa svarīgā informācija par izvēlēto līmeni, kā arī izvēle grūtības līmeņiem.

2.2.5. Izvēlēties grūtības iestatījumu

Mērķis:

Pēc nospiešanas uz konkrēta līmeņa tiek parādīta grūtības izvēle, popup logā.

Ievaddati:

- 1) Līmeņa dati
- 2) Peles kursorš nospiež uz kādu no grūtības iestatījumiem.

Nosacījumi:

- 1) Nav iespējams noklikšķināt uz grūtības līmeni kas nefunkcionē.
- 2) Parādās tikai spēles grūtības iestatījuma sadaļā.

Apstrāde:

Pēc nospiešanas sākt spēli poga tiek atvērta un var nospiegt. Kā arī ir redzams kurš līmenis tika izvēlēts.

2.2.6. Sākt spēli poga.

Mērķis:

Atvērt spēles nodaļu, ar visiem datiem kas ir vajadzīgi, lai spēlētu iepriekš noteikto līmeni un grūtību.

Ievaddati:

- 1) Peles kursorš nospiež uz sākt spēli pogu kas atrodas grūtības iestatījumu logā.
- 2) Dati par kurš grūtības līmenis tiek izvēlēts.
- 3) Iestatījumi par spēles noteikumiem un grafiskajiem iestatījumiem.

Nosacījumi:

- 1) Nav iespējams nospiegt, ja lietotājs nav izvēlējis grūtības līmeni.
- 2) Ir redzama tikai grūtības izvēles logā.

- 3) Poga nosaka, vai visi dati ir pieejami un neļauj spēlei sākt ja ir kļūda ar līmeni.

Apstrāde:

Pēc nospiešanas lietotājs tiek pārnests uz spēles sadaļu.

2.2.7. Pauzēšana

Mērķis:

Atvēta spēles logu kurā ir iestatījuma poga, atsākt spēli poga un iziet no spēles poga.

Ievaddati:

- 1) Klaviatūras “esc” poga.

Nosacījumi:

- 1) Tikai paradās, ja lietotājs atrodas spēles sadaļā.

Apstrāde:

Pēc nospiešanas spēle tiek apstādināta tajā momentā, lai parādītu izvēlnes logu.

2.2.8. Atsākt spēli poga

Mērķis:

Pēc pauzēšanas tikt atpakaļ pie spēles progresā.

Ievaddati:

- 1) Peles kursors nospiež uz pogu.
- 2) Klaviatūras “esc” poga.

Nosacījumi:

- 1) Ir redzams tikai spēles sadaļā un tikai, ja ir atvērts pauzēšanas logs.
- 2) Tikai aizver logu ja tas ir jau atvērts.

Apstrāde:

Pēc atsākšanas tiek uzsākts 2 sekunžu taimers un spēle atsākas iepriekšējā pozīcijā kurā lietotājs pauzēja.

2.2.9. Iziēt no līmeņa

Mērķis:

- 1) Pēc pauzēšanas pārvietot lietotāju uz spēles izvēlnes sadaļu.

Ievaddati:

- 1) Peles kursors nospiež uz iziet no līmeņa pogas.

Nosacījumi:

- 1) Ir redzams tikai spēles sadaļā un tikai, ja ir atvērts pauzēšanas logs.

Apstrāde:

- 1) Spēles progress un dati netiek saglabāti un lietotājs tiek pārvietots uz izvēlēties līmeņa sadaļu.

2.2.10. Lasīt līmeņa datus priekš spēles sākšanas

Mērķis:

- 1) Izlasīt līmeņa failu datus un saglabāt viegli lasāmā formātā, priekš līmeņa

Ievaddati:

- 1) Izvēlēta līmeņa faila nosaukums
- 2) līmeņa izvēlētais grūtums

Nosacījumi:

- 1) Līmeņa failiem ir jābūt noteiktā direktorijā
- 2) Izsauk tikai vienu reizi pēc spēles sākšanas

Apstrāde:

- 1) Dati tiek saglabāti lokāli, lai varētu būt viegli izlasāmi līmeņa vidū.
- 2) Pārbauda vai dati ir korekti un strādā
- 3) Mazas problēmas ar nepietiekamiem datiem ir jābūt piepildītām automātiski

2.2.11. Pārskatīt visu līmeņu datus

Mērķis:

Izlasīt visu līmeņa failus, lai var parādīt līmeņu izvēlē (2.2.1.)

Ievaddati:

- 1) līmeņa faila direktorija

Nosacījumi:

- 1) Ir jābūt kaut vienam līmenim, iekšā direktorijai

Apstrāde:

- 1) Izlasa un izvada katra līmeņa nosaukums un grūtības līmeņus
- 2) Ja ir specificēts parāda PNG failu un maina līmeņa apmales krāsu specificētam līmenim.

2.2.12. Lādēšanās ekrāns

Mērķis:

Parādīt lietotājam kā programma procesē datus, lai lietotājs zina ka spēle nav apstājusies.

Ievaddati:

- 1) Apstiprinājums ka nākošā sadaļa ir gatava
- 2) Paziņojums ka jauna sadaļa tiek atvērta

Nosacījumi:

- 1) Radās tikai ja aizņem ilgāk par 0.5 sekundes, lai ielādētos.

Apstrāde:

- 1) Pēc katra procesa tiek izdots paziņojums ka progress ir pabeidzies, un tiek noņemts lādēšanās ekrāns

2.2.13. Spēles laikā nospiesta kāda no pozīcijas pogām

Mērķis:

Kad lietotājs nospiež kādu no četrām kustības pogām (w, a, d, vai space) spēlei ir jāreaģē, vai tas bija pareizā laikā, vai nē.

Ievaddati:

- 1) Spēles noteiktās bultas pozīcija
- 2) Spēles noteiktās bultas tipu
- 3) Lietotāja nospiestās pogas vērtību un nospiešanas ilgumu

Nosacījumi:

- 1) Tikai tiek rēķināta, ja lietotājs ir jebkurā līmenī
- 2) Tikai skaita punktus un dzīvību pēc sākuma 2 sekundes gaidīšanas periodu
- 3) Netiek rēķināta pēc līmeņa pabeigšanas

Apstrāde:

- 1) Kad lietotājs nospiež jebkuru no 4 pogām jebkura no šīm iespējām var notikt:

1.tabula

Funkcijas apstrāde

	Īsā klikšķa bulta gadījumā	garu spiedienu bultas gadījumā	Neviena bulta laukā
Poga tiek nospiesta tieši laikā	bonuss punktus un dzīvību iedod	pusi iespējamiem punktiem	neskaita
Poga tiek Nospiesta mazliet par ātru/lēnu	Punkti iedoti, bet ne dzīvību	ceturtdaļu iespējamiem punktiem	neskaita
Poga tiek nospiesta pārāk vēlu/ātru	Nav punkti iedoti, dzīvība samazinās	Nav punkti iedoti, dzīvība samazinās	neskaita
Poga netiek nospiesta	Nav punkti iedoti, dzīvība samazinās	Nav punkti iedoti, dzīvība samazinās	neskaita
Nepareizā poga tika nospiesta	Nav punkti iedoti, dzīvība samazinās	Nav punkti iedoti, dzīvība samazinās	neskaita
Poga nospiesta laikā un atlaista perfekti	Bonuss punktus un dzīvību iedod	Bonuss punktus un dzīvību iedod	neskaita
Poga nospiesta laikā, bet atlaista par vēlu/ātru	Bonuss punktus un dzīvību iedod	Punkti iedoti, bet ne dzīvību	neskaita

Poga tiek nospiesta par vēlu, bet atlaista perfekti	Punkti iedoti, bet ne dzīvību	Punkti iedoti, bet ne dzīvību	neskaita
--	-------------------------------	-------------------------------	----------

2.2.14. Spēles noteikto bultu ielādēšana

Mērķis:

Norādīt lietotājam kuras pogas, cik ilgi un kurā laikā ir jāspiež, lai progresētu līmenī.

Ievaddati:

- 1) Līmeņa dati
- 2) Līmeņa modifikācijas
- 3) Šobrīdējais laiks līmenī

Nosacījumi:

- 1) Bultas vienmēr ir ritmiski kopā ar mūziku

Apstrāde:

- 1) Kad līmenis ir sācies un ir līdz galam izlasīts, programma ievieto bultiņu spēles augšā, tieši kādu pozīciju, tipa un kad līmeņa fails nosaka.

2.2.15. Īsā klikšķa bultiņa

Mērķis:

Ja “Spēles noteikto bultu ielādēšana” ielādē īsa klikšķa bultiņu, šis nosaka visu kas notiks ar to.

Ievaddati:

- 1) Bultas ātrums
- 2) Vai ir lietotājs nospiedis pogu laikā

Nosacījumi:

- 1) Bultas tiek izdzēstas ja tās ir ārpus lietotāja skatam.

Apstrāde:

- 1) Pēc ielādēšanās, bultas spraits kustās uz leju noteiktā ātrum
- 2) Ja bultas sasniedz atzīmēto vietu tās dati tiek pārsūtīti uz “Spēles laikā nospiesta kāda no pozīcijas pogām” funkciju

2.2.16. Garu spiedienu bultiņa

Mērķis:

Ja kāda no spēles tekstūrām ir modificētas “Resources”

Ievaddati:

- 1) Bultas ātrums
- 2) Bultas garums
- 3) Vai ir lietotājs nospiedis pogu laikā

Nosacījumi:

- 1) Bultas tiek izdzēstas ja tās ir ārpus lietotāja skatam.

Apstrāde:

- 1) Pēc ielādēšanās, bultas spraits kustās uz leju noteiktā ātrumā
- 2) Bultai ir jābūt noteiktā garumā.
- 3) Ja bultas sasniedz atzīmēto vietu tās dati tiek pārsūtīti uz “Spēles laikā nospiesta kāda no pozīcijas pogām” funkciju.

2.2.17. Fotogrāfijas un tekstūras apstrāde mainīšanas gadījumā.

Mērķis:

Izveidot funkciju kas reaģēs ja lietotājs ir mainījis tekstūras failus, vienalga vai mainot vai izdzēšot.

Ievaddati:

- 1) Failu direktorijs/faili.

Nosacījumi:

- 1) Jaunas tekstūras tiek izmantotas spēlē.
- 2) Ja faili ir izdzēsti, tie tiek aizvietoti izmantojot nemodificējamus failus spēles datu direktoriņā.

Apstrāde:

- 1) Pēc modifikācijas tekstūras mainās.

2.2.18. Sāukmlapas kursora sekojos aizmugures lauks

Mērķis:

Izveidot interesantu izskatu sāukmlapā, kas seko peles kursoram un maina aizmugures lauka izskatu.

Ievaddati:

- 1) Peles kursora atrašanās vieta

Nosacījumi:

- 1) Peles kursors netiek ietekmēts
- 2) Aizmugures lauks kustās minimāli
- 3) Nevar redzēt fotogrāfijas galus

Apstrāde:

1) peles koordinātes nosaka kameras atrašanās vietu un rotāciju

2.3. Sistēmas nefunkcionālās prasības

2.3.1. Veikt spējas prasības

Spēlei ir jāstrādā ar jebkuru datoru ar Gtx 1050 grafikas kartes un i7-6800K CPU ekvivalentiem. Lai gan ar ir jābūt grafisku intensitātes un partikulu iestatījumiem, lai pārliecinātos ka vecāki, vai lētāki datori var spēlēt.

2.3.2. Vizuāls izskats

Visai spēlei ir jābūt krāsainai, lai uzturētu “Cyberpunk” vai “neon” mākslas stilu. Kā arī visām pogām un slaidajiem ir jābūt viegli identificējamiem. Ja pogu nevar nospiest, tam ir jābūt viegli identificējamam.

2.3.3. Skaņu dizains

Pēc katras pogas vai slīdņa kustības ir jābūt kaut kādai identificējami skaņai, tas arī ieskaita, spēles ritma gaitā.

2.3.4. Modificējama

Jābūt relatīvi vienkārši izveidot jaunu līmeni un pievienot to spēlē, pat bez Unity projekta pieejas. Kā arī vizuālie elementi, katrā līmenī ir jābūt viegli modificējamiem. Kā piemēru priekš katras dziesmas, tiek mainīts ceļu un bultiņas tekstūru.

2.4. Gala lietotāja raksturiesīmes

Veidojot ritmu spēli kura ir modificējama, veidos divu veidu lietotājus. Vienkāršus lietotājus, kuri instalēs trešās partijas veidotus līmeņus, dizaina elementus un mūziku. Kā arī modifikatori, kuri veidos trešās partijas līmeņus, dizaina elementus un mūziku, kurus varēs publicēt un pievienot pie Cybershock.

Lietotāji neizzinās neko saistītu ar iekšējo datu struktūru, vai kā šī spēle strādā. Tādēļ jāpārliecinās ka visa datu struktūra ir viegli atrodamā, labojama un viegli saprotama. Kā piemēru ja lietotājs instalē trešās partijas līmeni, kurai ir savi dizaina elementi un mūzika, ir jābūt vieglam instalācijas procesam.

Modifikatori ir lietotāji, kuri veidos jaunus līmeņus un mainīs failus, lai pārliecinātos ka modifikatori var viegli veidot savus līmeņus. Tādēļ ir jābūt vai nu dokumentācijai, vai instrumenti,

lai veidotu korespondējošos formātus. Kā arī ir jābūt log fails, vai veidu kā informēt lietotāju ja ir problēmas ar programmatūru.

3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums

Šeit tiks aprakstīti visi rīki un instrumenti kuri tiks izmantoti, lai izveidotu programmatūru pēc dokumentācijas iepriekš noteiktam prasībām. Pirmajā nodaļā tiks noteiktas visas programmatūras un līdzekļus kuri tiks izmantoti izstrādes laikā. Otrā sekcija vairāk aprakstīs visus resursus un līdzekļus kuri netiks obligāti lietoti, lai gan var būt noderīgi programmatūras izstrādes laikā.

3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts

Šī nodaļa saturēs visas programmatūras, programmēšanas valodas un līdzekļus kuri tiks lietoti izstrādes procesā. Šī nodaļa tiks sadalīta 3 daļās, līdzekļi, Dizaina rīki, datu glabāšana. Kur tālāk tiks izskaidrots katrs resurss, kā tiks izmantots un kādēļ tieši šo konkrēto līdzekli.

3.1.1 Izstrādes līdzekļi

- Unity 2019.4.f3.1

Galvenā sadaļa tiek taisīta Unity spēles dzini, kurš tiks izmantots, lai izveidotu spēles funkcionalitāti, failu glabāšanu, renderēšanu un citu programmatūru integrāciju. Lai atbalstītu iepriekšējo pieredzi tiks izmantota Unity versiju 2019.4.f3.1 ar kuru ir iepriekšējā pieredze. Var būt problēmas ar šo versiju, viss īpaši ar partikulu un efektu sadaļu. Tādēļ pāriešana uz citu versiju var notikt.

- Studio Code Editor

Lai izmainītu Unity programmatūru ir jāizmanto C# programmēšanas valodu, lai to valodu rediģētu es izmantošu rekomendēto IDE. Pie Studio Code Editor ir iespēja pievienot

3.1.2 Dizaina rīki

- Photoshop

Lai veidotu lielāko daļu 2D dizainus tiks izmantota jaunākā versija ar Photoshop. Photoshop ir noderīgs dēļ krāsas regulācijas iestatījumiem kas atļauj izveidot detalizētus emisijas un svara kartes.

- Blender

Blender programma tiks izmantota lai veidotu 3D objektus un UV kartes. Šī programma tika izvēlēta, dēļ tās vairākām iebūvētām funkcijām kas atļauj izmantot tikai vienu programmu veidojot visas daļas kas ir saistītas ar 3D modeļiem.

- Inkscape

Šī ir vēl vienā programma kuru var izmantot, lai veidotu 2D dizainus, viņa ir atvērta programma ar vairākām funkcijām kas ir vairāk noderīga precizitātei un izmēriem.

3.1.3 Datu glabāšanas formatēšanā

- Json

Json faila formāts ir ļoti viegli modificējams un pieejams caur lielāka daļa valodām tas izveido daudz vieglāku datu modifikāciju un veidošanu. Kā arī tas atļauj pievienot jaunus pievienojumus un modifikācijas nemainot kopējo datu struktūru.

3.2. Iespējamo risinājuma līdzekļu un valodu apraksts

Iespējamo risinājumu līdzekļu nodaļa ir domāta, lai rādītu alternatīvas risinājumus visām iepriekš pieminētām programmatūrām un risinājumiem. Lai gan šie risinājumi nav plānoti, tie ir noderīgi gadījumos kad ir kādas problēmas ar iepriekš minētām programmatūrām.

3.2.1 Izstrādes līdzekļi

- Unity 2022.3.6f1

Pat ja šī versija ir jaunāka un modernizētāka un ar ļoti līdzīgām un vairākām jaunām funkcijām. Tas iemesls kādēļ šī versija netiek izmantota ir dēļ tā ka šī versija nav ierasta un nav pieredze izmantot. Viena no lietām kas bija izmainīta ir GUI un ģenerālu dizainu, tas var ļoti ietekmēt projekta taisīšanas procesu.

- Unreal Engine

Šis ir cits dzinis kas ir populāri lietots. Tam ir vairāki uzlabojumi kas nav unity, kā piemēram labāka ēnotāja kontrole, apgaismojuma sistēma un ģenerāli labāk optimizēta. Lai gan es regulāri to neizmantoju un nav daudz pieredze. Unreal ir arī vienkāršojusi vairākas sadaļas kas nozīmē būs mazāk jāraksta skripti.

- Godot

Godot ir alternatīvs Unity spēles dzinis, šis ir daudz mazāk populārs nekā Unity vai Unreal. Godot izmanto 4 valodas, GDScript, C#, C++ un C++. No tām man ir tikai pieredze ar C# un C++. Godot, ir iepriekš lietots, bet nav pietiekami daudz pieredze, vai mentori.

- Visual studio

Visual studio ir parasti rekomendēta priekš Unity C# galvenokārt dēļ visām trešās partijas bibliotēkām un modifikācijām kuras ir domātas C# un konkrētu Unity. Pat ja lielākā daļa Unity programmētāji izmanto šo IDE tai ir problēmas ar optimizāciju, un parasti aizņem pārāk daudz RAM kas var ļoti ietekmēt testēšanu.

3.2.2 Dizaina rīki

- Inkscape

Šī programma ir domāta, lai veidotu vektora grafikus kuri ir daudz labāki priekš videospēles grafikiem, lai gan Inkscape ir krāsas imitācijas kas izveido problēmas ar modifikāciju, svāra kartes veidošanu un emisijas kartes veidošanu.

3.2.3 Datu glabāšanas formatēšanā

- Īpašumtiesības formatēts fails

Alternatīvi JSON formātam ir iespēja veido īpašumtiesības formātu. Paštaisīts formāts var aizņemt mazāk atmiņu, ātrāk lasāms un viss svarīgāk var aizsargāt pret pirātismu vai trešās partijas modifikāciju. Vairākas kompānijas izmanto paštaisītu, vai īpašumtiesības formātu dēļ iepriekš minētiem iemesliem. Lai gan, priekš indi ražošanas šī drošība un minimāli uzlabojumi nav pietiekami lieli, lai veidotu pilnībā jaunu formatējumu. Jauns formatējums nozīmē ka jāveido jauns lasītājs, re-formatētājs un rakstītājs, visi var aizņemt pārāk daudz laiku šādam projektam.

3.2.4 Datu glabāšanas modifikācija

- Paštaisīta JSON kartes veidošanas aplikācija

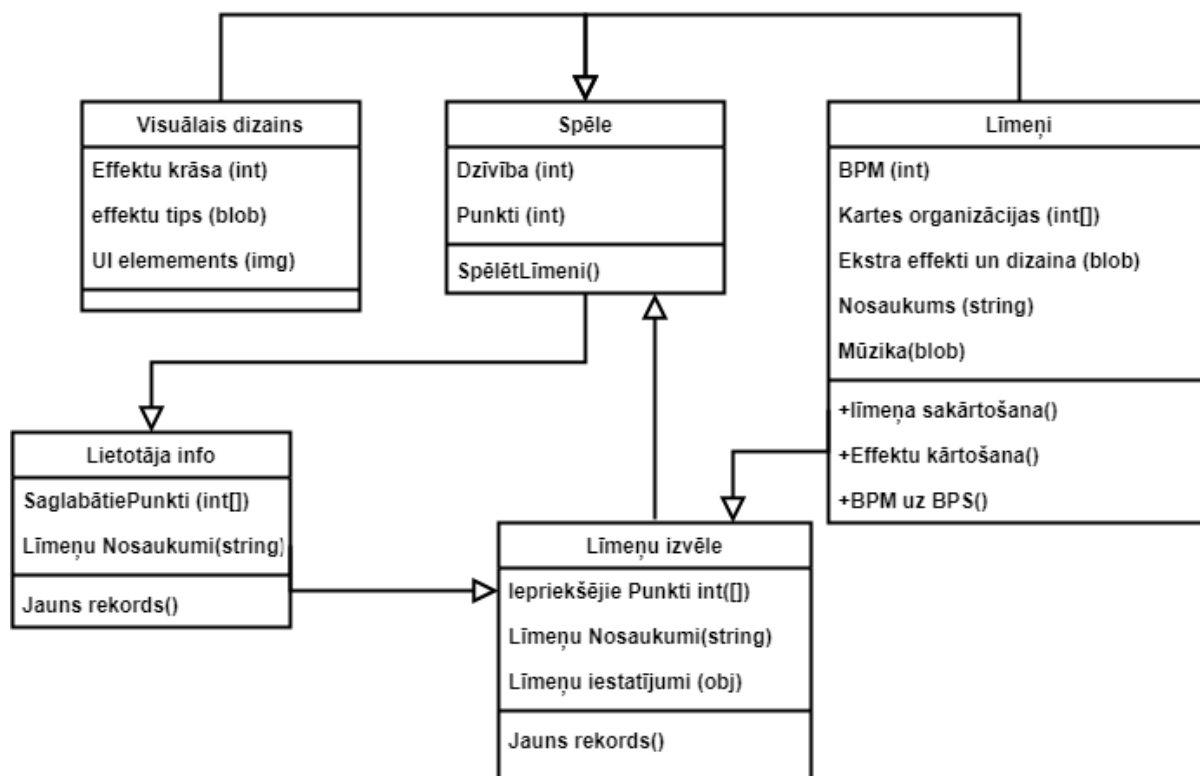
Lai veidotu līmeņus ātrāk ir iespēja veidot alternatīvu programmu kas izmantojot JSON formātu atļaus spēlēt korespondējošo mūziku un demonstrācijas skatu no videospēles skatupunkta. Šādi veidota programmatūra ir specifiska un tikai izmantojama ar cybershock JSON kārtējuma formātu.

4.1. Sistēmas struktūras modelis



4.2. Klašu diagramma

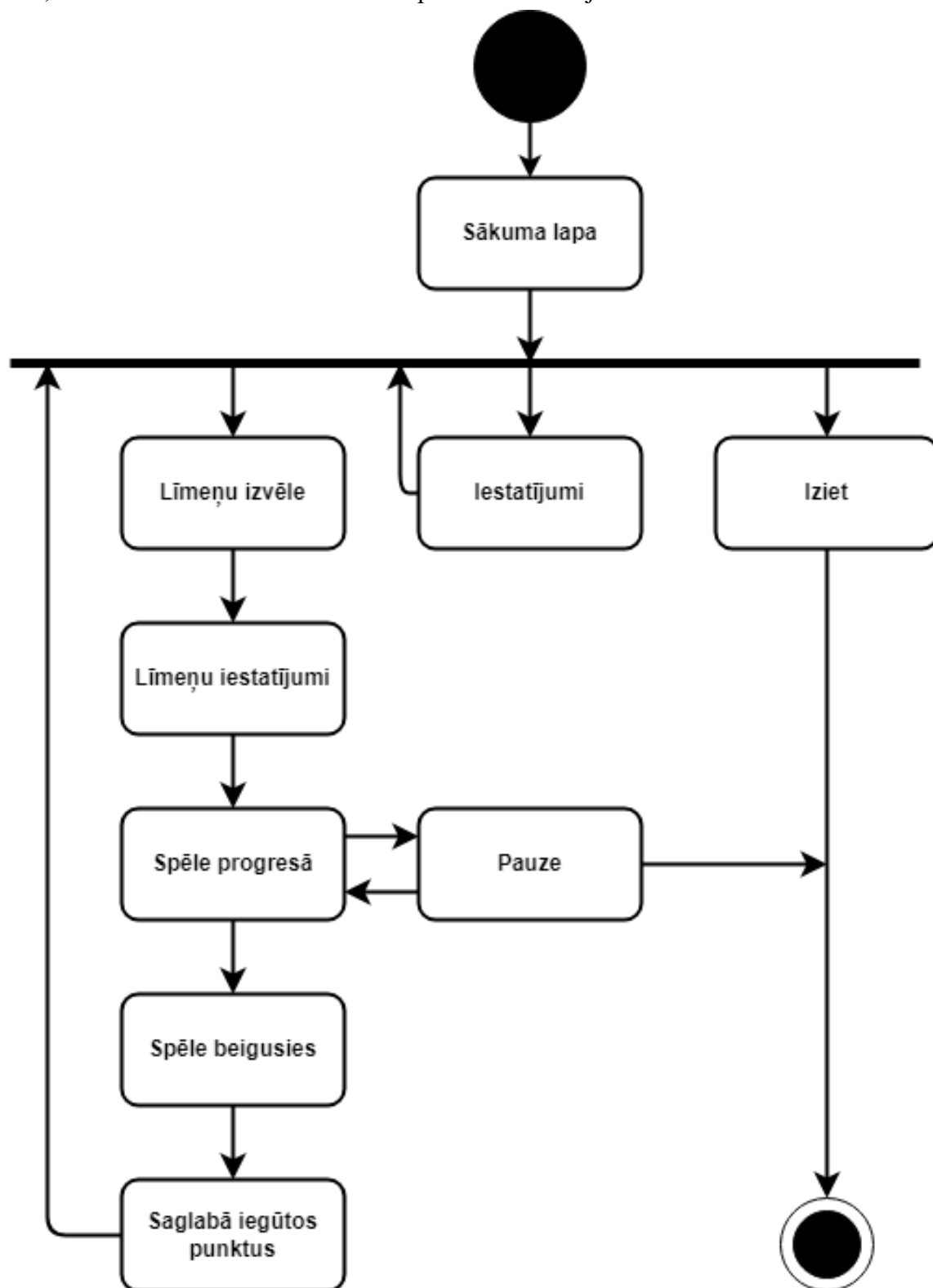
Šī diagramma reprezentē datus kuri tiks izmantoti projektā. Katra sadaļa ir individuālā failā vai faila direktorijā.



2. attēls. Klašu diagramma

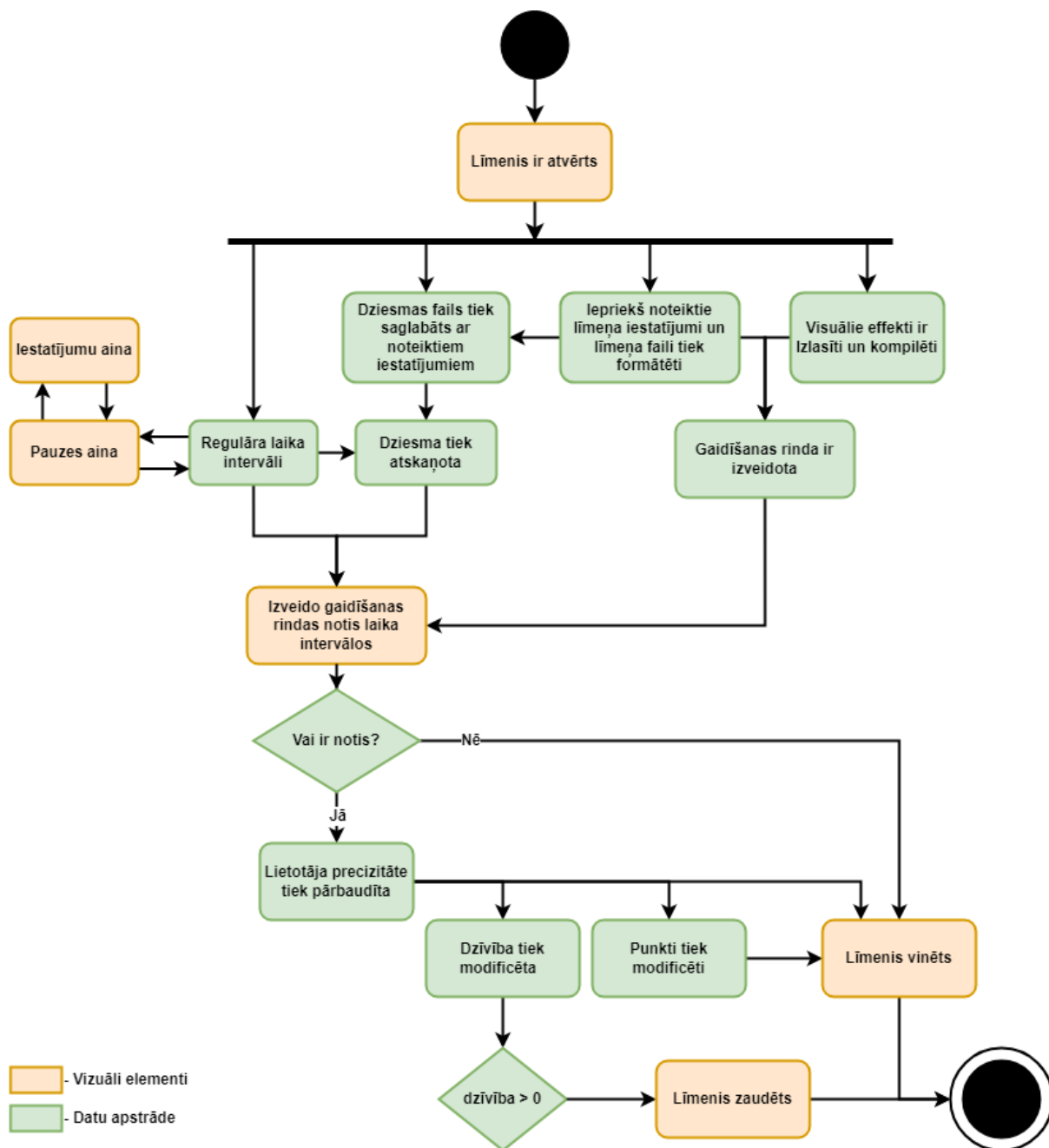
4.3. State diagramma

Šī diagramma reprezentē tikai navigācijas un loģistikas funkcijas. Šī diagramma nav veidota, lai demonstrētu internālas un datu pārraides funkcijas.



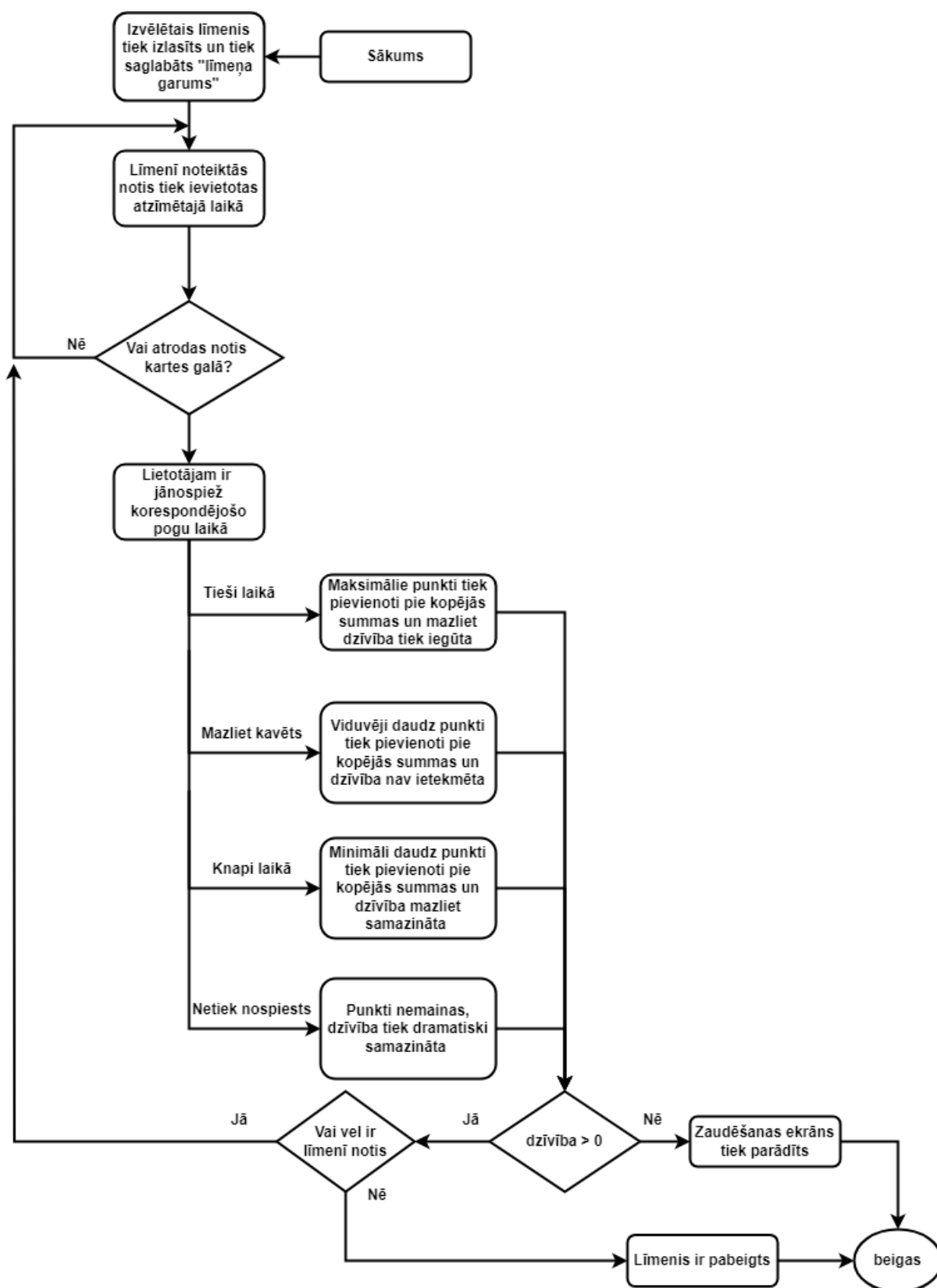
3. attēls State diagramma

Šī diagramma ir domāta, lai reprezentētu “spēle progresā” nodaļu un labāk izskaidrotu kā lietotāji pārvietosies cauri spēles elementiem un kā tie mainīsies relatīvi padotajiem datiem un lietotāju reakciju.



4. attēls State diagramma

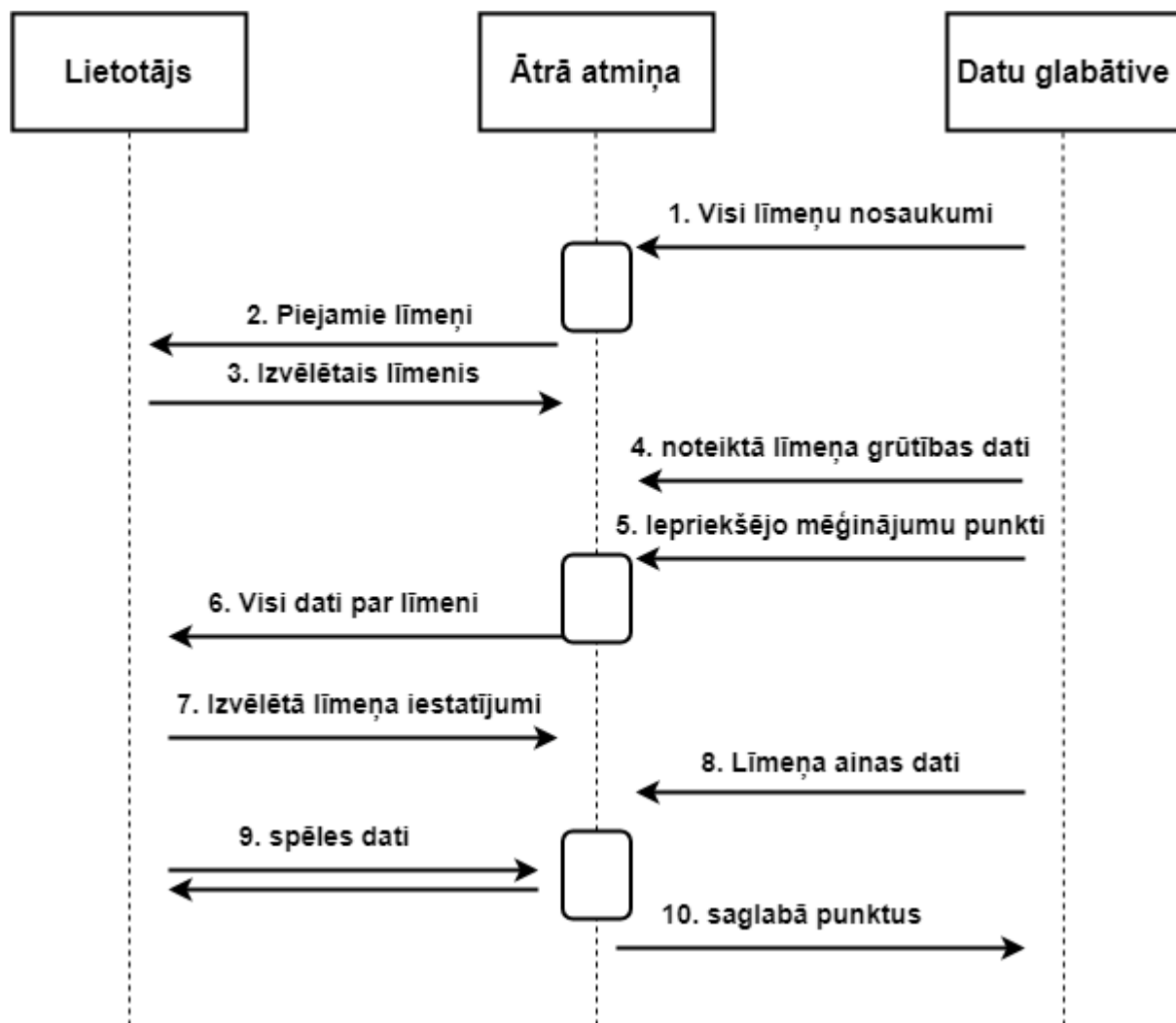
4.4. Algoritma diagramma



5. attēls Algoritma diagramma

4.5. Scenāriju diagramma

Lai reprezentētu datu plūsmas laikus un ieguves vietu šis dinamiskais sistēmas modelis ir domāts, lai atrastu iespējamus konfliktus, vai neloģiskas informācijas pārraides gadījumus.

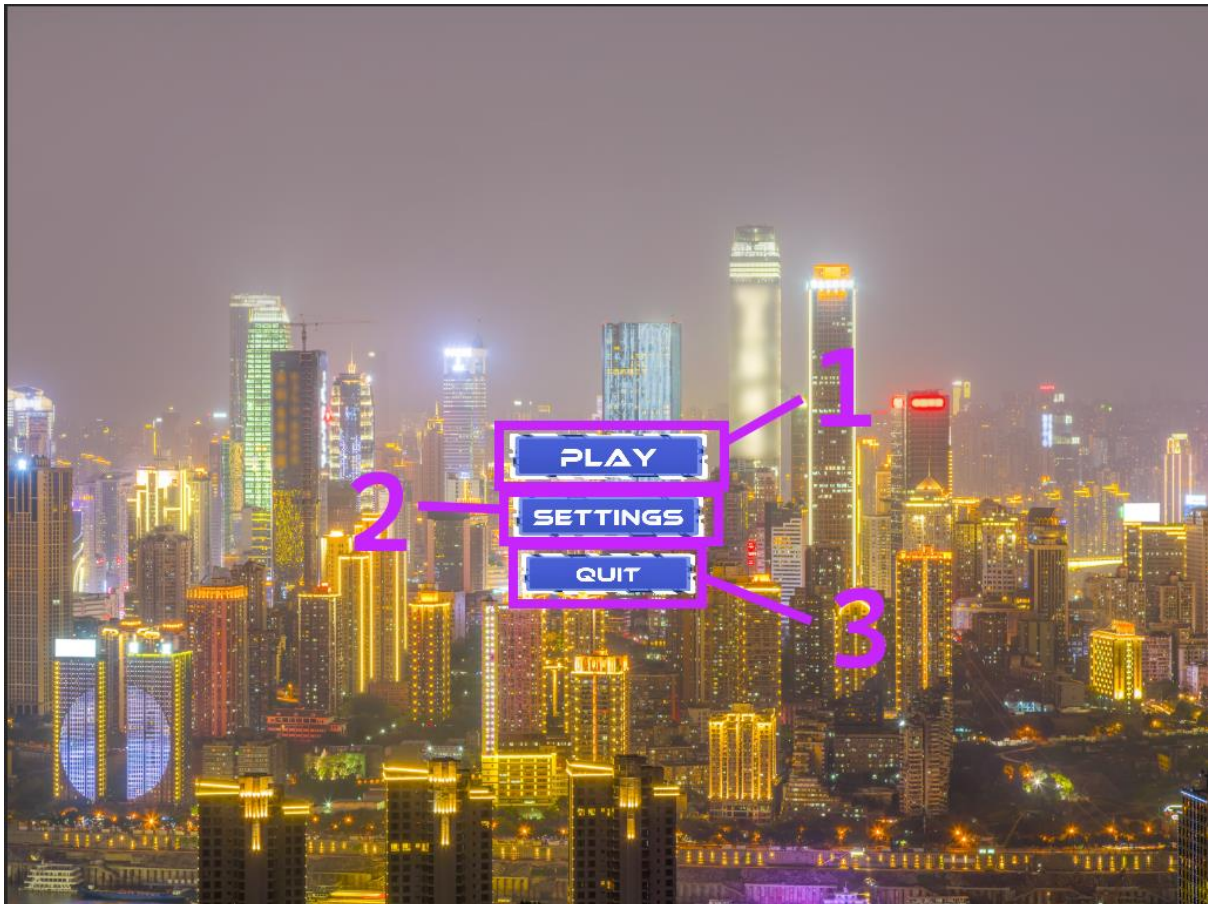


6. attēls Scenāriju diagramma

5. Lietotāju ceļvedis

Šī nodaļa ir domāta, lai izskaidrotu lietotāju navigāciju un kā spēlēt spēli. Pogu atrašanās vieta paliks vienāda, bet ir iespējams ka grafiski dizaina elementi var mainīties, vai dēļ jaunas versijas, vai pēc modifikatorā izvēles.

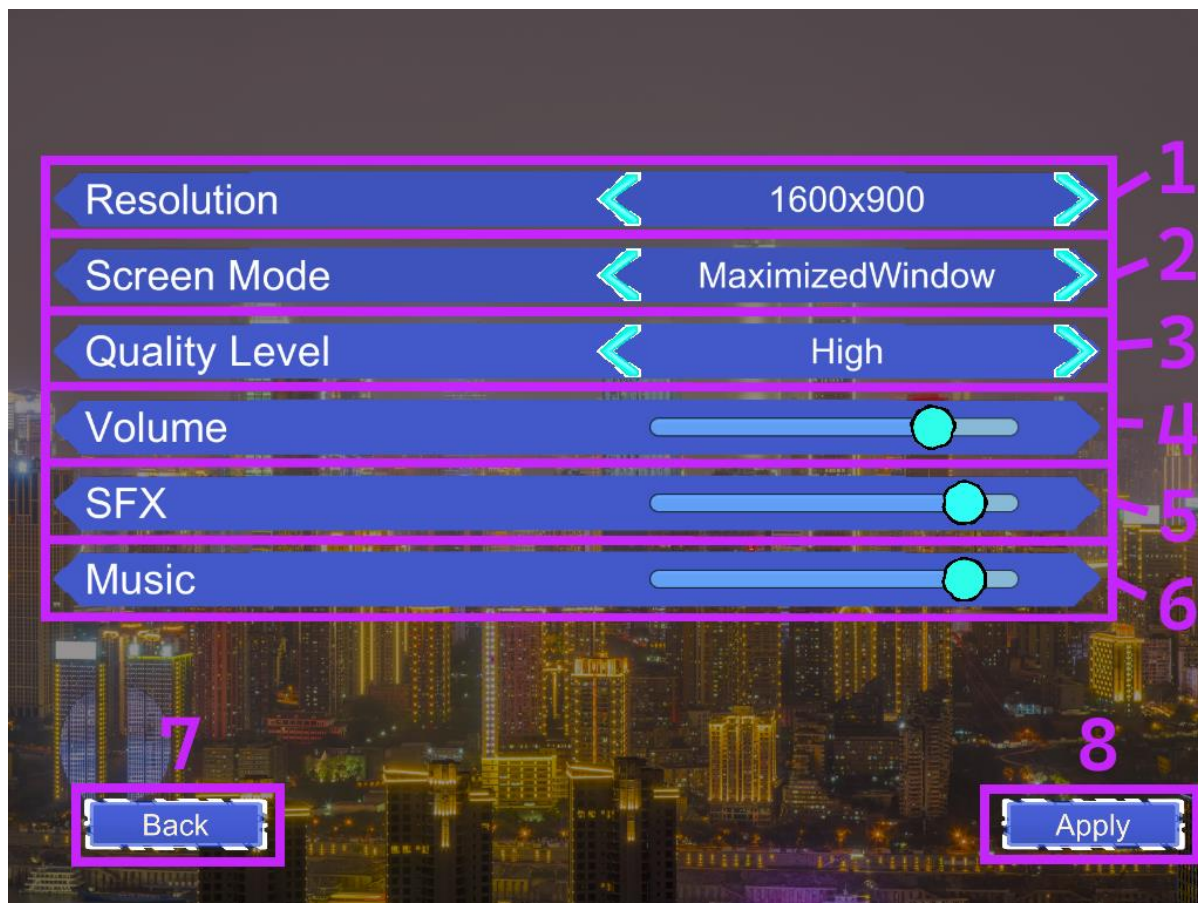
5.1 Sāukmlapa



7. attēls Sāukmlapa

1. Play pārvietos tevi uz 5.3 nodaļu kurā varēs izvēlēties kuru līmeni vēlies spēlēt.
2. Settings pārvietos tevi uz 5.2 nodaļu kurā varēs mainīt iestatījumus.
3. Quit aizver aplikāciju.

5.2 Iestatījumi



8. attēls Iestatījumi

1. Resolution izmaina izšķirtspēju kurā spēle tiks vadīta, pa kreisi ir mazāka un pa labi palielina izšķirtspēju.
2. Screen mode maina kā spēle tiks vadīta caur windows, pa kreisi ir “Windowd” kas nozīmē ka izskatīsies tā kā jebkura cita aplikācija, ja ir “Borderless window” tad tas aizņems visu ekrānu, bet varēs iziet ārā no programmas un izmantot citu ekrānu neaizverot programmu. “Full screen” aizņems visu ekrānu un ja nospiedīs uz jebkuru citu aplikāciju programma minimizēs sevi.
3. Quality iestatījumi maina cik detalizēti izskatīsies spēle. Konkrētas iespējas ir izskaidrotas tabulā, kur kreisā puse norāda zemākas kvalitātes iestatījumus kas ir pieejami un pa labi rāda visas augstākas kvalitātes iestatījumus.

Kvalitātes iestatījumi

“Very low”	“Low”	“Medium”	“High”	“Very high”
Daļiņas ir ļoti samazinātas, fotogrāfijas rezolūcija ir samazināta un gaismas efekti it izslēgti.	Daļiņas ir limitētas bet ir vairāk nekā “Very low”. Tekstūrām izšķirtspēja ir mazliet samazināta. Daži gaismas efekti ir redzami	Daļiņas nav limitētas un foto izšķirtspēja ir oriģinālā lielumā. Gaismas efekti ir mazliet limitēti un vienkārši aprēķināti	Daļiņu efekti ir ieslēgti, apgaismojumi ir optimāli veidoti. Apgaismojuma efekti ir vairāk un atspoguļojas pret citiem objektiem	Ir pieliktas vel daļiņas, efekti un foto izšķirtspēja tiek palielināta.

4. Volume vada kopējo aplikācijas skaļumu, kā piemēru ja mūzikas skaļums ir 50% un volume ir arī 50% tad kopējais mūzikas skaļums būs 25%.
5. SFX ir efektu skaļums, tās ir skaņas kas tiek atskaņotas ja nospiež pogas, vai spēlēšanas laikā efektu skaņas.
6. Music maina mūzikas skaļumu, tas ir gan spēles laikā, gan ceļvedī.
7. Back pārvieto tevi atpakaļ uz 5.1 nodaļu.
8. Apply saglabā iestatītos iestatījumus.

5.3 līmeņu izvēle



9. attēls līmeņu izvēle

1. Līmeņa izvēles nodaļa – šis saraksts satur visas spēles līmeņus, ja pāri tiem pāriet ar pelīti tā palielinās un ja tu uzklikšķini tās fotogrāfija un nosaukums parādās kreisajā pusē un maza daļa no mūzikas tiek atskaņota. Lai ietu cauri sarakstam var izmantot peles ritenīti vai turot kreiso taustiņu lejā un kustinot peli, līdzīgi kā saraksti telefonos.
2. Kreisā nodaļa sastāv ar izvēlēto dziesmas foto un nosaukumu. Ja neviena nav izvēlēta, kā piemēram ja ir tikko atvērts, spēle izvēlas kādu nejauši.
3. Zem tā ir grūtības iestatījumu saraksts. Katrai mūzikai var būt citi nosaukumi, krāsas un grūtības iestatījumu skaits. Kad kāda grūtības izvēle ir nospiesta tā tiek atzīmēta un “play” poga ir pieejamā.
4. Back poga ignorēs kas tika nospiests un pāries uz sāukmlapu.
5. Play poga ir tikai pieejama, ja ir izvēlēta grūtības izvēle. Šī poga pārvietos tevi uz pašu spēli ar iestatīto mūziku un grūtību.

6. Testēšanas dokumentācija

6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums

Galvenokārt tiks izmantots “black box” metodi, kura nozīmē ka testēšanas laikā nav pieja pie programmas failiem. Šī metode tiek izmantota, jo to var vieglāk izdalīt starp citiem testētājiem. Lai gan dažos momentos tiks izmantots “white box” ar šo metodi ir pieeja pie programmas failiem. Dažos gadījumos ir vajadzīgi, piemēram līmeņu lasīšanā.

6.2. Testpiemēru kopa

CyberShock Tests			
<u>Identifikatoru atšifrējums:</u>			
Piemērs: TP.IEV.IZV.01			
PR	Prasība		
TP	Testpiemērs		
Nosaka piederību moduļim:			
SĀK	Sākums		
IZV	Izvērne		
UEST	Iestatījumi		
PA	Pauze		
SP	Spēle		
VA	Visuālie aspekti		
FA	Funkcionālie aspekti		

10. attēls Testēšanas apraksts

Prasības ID	Prasība
Sākums	
PR.SĀK.01	Nospiežot "Quit" pogu spēle tiek aizvērta
PR.SĀK.02	Nospiežot "Options" pogu Izvēlnes lapa tiek atvērta
PR.SĀK.03	Nospiežot "Play" pogu iestatījumu lapa tiek atvērta
Izvēlne	
PR.IZV.01	Atverot lapu tiek izlasīti dati no Resorces foldera
PR.IZV.02	Tiek parādīti top 5 mēģinājuma punkti
PR.IZV.03	Izvēlējoties izvēli, tiek parādīta vairāk informācija kreisajā pusē
PR.IZV.04	Nospiežot "Back" pogu Sākuma lapa tiek atvērta
PR.IZV.05	Grūtības līmeņi tiek izlasīti no Resorces foldera
PR.IZV.06	"Play" poga ir tikai pieejama, ja ir izvēlēta grūtība
PR.IZV.07	Nospiežot "Play" pogu atverās spēles lapa
Iestatījumi	
PR.IEST.01	Var mainīt ekrāna rezolūciju
PR.IEST.02	Var mainīt ekrāna tipu
PR.IEST.03	Var mainīt spēles kvalitāti
PR.IEST.04	Var mainīt skaļumu
PR.IEST.05	Var individuāli mainīt efektu skaņu
PR.IEST.06	Var individuāli mainīt mūzikas skaņu
PR.IEST.07	Iestatīt izmainītās vērtības
Spēle	
PR.SP.01	Jebkurā momentā var pauzēt spēli, lai mainītu iestatījumus vai iziet no līmeņa
PR.SP.02	Izlasa dziesmas datus
PR.SP.03	Lādēšanās ekrāns kurš parādās ja līmeņa ielādēšana aizņem pārāk ilgu laiku
PR.SP.04	Viena spiediena pozīcijas pogas
PR.SP.05	Turot pozīcijas pogu
PR.SP.06	Notis iziet no ekrāna un samazina dzīvību
PR.SP.07	Punkti tiek iedoti pēc prasībām
PR.SP.08	Līmeņa datus var izlasīt
PR.SP.09	Var ielādēt viena spiediena noti/būltu
PR.SP.10	Var ielādēt turēšanas noti/būltu
PR.SP.11	var notiekt notes ātrumu
PR.SP.12	var notiekt notes atkārtojamību

11. attēls Testēšanas prasība

6.3. Testēšanas žurnāls

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra apraksts	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra sagaidāmais rezultāts	Prasības ID
Black Box							
TP.FA.01	Programmas startēšana	Nospiežot uz spēles ikonu spēle atverās, un aizver pati sevi	Programmas atvēršana	1) Programma ir atvērta 2) gaidot 20 sekundes neviens brīdinājums nerādas	Peles klikšķis	Tiek atvērts jauns logs	PR.01
TP.FA.02	Programmas apturēšana	Aizverot programmu nerādas brīdinājumi, vai ilgi lādējas	Pēc programmas startēšanas ir iespējams apstādināt programmu	1) Programma tiek aizvērta 2) gaidot 20 sekundes neviens brīdinājums nerādas	Peles klikšķis	Apturēta programma	PR.02
TP.SĀK.FA.01	Programmas apturēšanas poga	Aizverot programmu nerādas brīdinājumi, vai ilgi lādējas	Pēc programmas startēšanas ir iespējams apstādināt programmu	1) Kad ir atvērta sākuma lapa tiek nospiesta "quit" poga	Peles kreisā taustiņa klikšķis	Programma tiek aizvērta	PR.SĀK.01
TP.SĀK.FA.02	Sākulapā iestatījuma poga	Pēc nospiešanas atverās logs kurā atrodas iestatījumu sadaļa	Sākulapā nospiežot iestatījumu pogu, tiek atvērti iestatījumi	1) Kad ir atvērta sākuma lapa tiek nospiesta "Settings" poga	Peles kreisā taustiņa klikšķis	Iestatījumi ir atvērti	PR.SĀK.02
TP.SĀK.FA.03	Sākulapā spēlēt poga	Pēc nospiešanas atverās logs kurā atrodas izvēlnes sadaļa	Sākulapā nospiežot spēlēt pogu, tiek atvērta izvēlnes	1) Kad ir atvērta sākuma lapa tiek nospiesta "Play" poga	Peles kreisā taustiņa klikšķis	Izvēlne ir atvērta	PR.SĀK.03
TP.IZV.VA.01	Izvēles izskata ģenerācija	Ja ir faili Resorces/Map-data folderi, tiem ir jābūt izlasītiem un pievienotiem izvēlnē	Atverot izvēni labajā pusē ir jābūt redzamai dziesmas nosaukumam un foto	1) Ievietot dziesmas datus iekšā Resorces/map-data folderi 2) Atver programmu 3) Atver izvēlnes sadaļu	1) Dziesmas dati iekšā Resorces/map-data folderī 2) kreisais klikšķis, lai nokļūtu Izvēlnē	Izvēlnes labajā pusē ir redzama dziesmas foto un dziesmas nosaukums	PR.IZV.01
TP.IZV.VA.02	Izvēles dziesmas datu izskata ģenerācija	Ja ir faili Resorces/Map-data folderi, tiem ir jābūt izvēlamiem un ir iespējams izvēlēties grūtības	nospiežot izvēnes lapas labajā pusē dziesmu ir jābūt redzamai tie dati kreisajā pusē	1) Ievietot dziesmas datus iekšā Resorces/map-data folderi 2) Atver programmu 3) Atver izvēlnes sadaļu 4) Izvēlēta dziesma	1) Dziesmas dati iekšā Resorces/map-data folderī 2) kreisais klikšķis, lai nokļūtu Izvēlnē 3) Kāda no dziesmām ir izvēlēta	Kreisajā pusē var redzēt dziesmas foto, nosaukumu un grūtības iestatījumi	PR.IZV.03

12. attēls Testēšanas piemēri

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra apraksts	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra sagaidāmais rezultāts	Prasības ID
TP.IZV.FA.01	Izvēles dziesmas datu izskata ģenerācija	Ja ir izvēlēta dziesma, tās top 5 augstākie punkti ir redzami	Ja ir izvēlēta dziesma tās "High score" ir jābūt redzamiem	1) Ievietot dziesmas datus iekšā Resorces/map-data folderī 2) Atver programmu 3) Atver izvēlnes sadaļu 4) Izvēlēta dziesma	1) Dziesmas dati iekšā Resorces/map-data folderī 2) kreisais klikšķis, lai nokļūtu Izvēlnē 3) Kāda no dziesmām ir izvēlēta	Kreisajā pusē ir redzami 5 augstāk iegūtie punkti dīlstoši uz leju	PR.IZV.02
TP.IZV.FA.02	Izvēlnē atpakaļ poga	Pēc nospiešanas atverās sākumlapas logs	Izvēlnē pēc pogas "back" nospiešanas tiek nomainīts logs uz sākumlapu	1) Kad ir atvērta sākuma lapa tiek nospiesta "Back" poga	1) Navigēt uz izvēlni 2) Peles kreisā taustiņa klikšķis uz "back" pogu	Sākumlapa ir atvērta	PR.IZV.04
TP.IZV.FA.03	Grūtības datu lasīšana	Izvēlnē grūtības iestaiījumu pogas satur datus kuri ir saistīti ar grūtības līmeņiem, kurus var izvēlē un vēlāk lietot citām funkcijām	Izvēlētās dziesmas grūtības līmeņi ir redzami un satur datus, kā piemēram nosaukumu un krāsu	1) Izvēlnē tiek izvēlēta dziesma 2) Tiek salīdzināti dati kas ir rakstīti dziesmas datos	1) Dziesmas dati ir ievietoti iekšā Resorces/map-data folderī 2) Izvēlnē izvēlēta dziesma	Ja dziesmas datos ir noteikts ka grūtības līmeņa krāsai ir jābūt zaļai un nosaukumam "easy", tad pogai kas ir domāta tai izvēlei ir arī jābūt zaļai	PR.IZV.05
TP.IZV.FA.04	Grūtības izvēles Imitācija	Pirms lietotājs var spēlēt līmeni ir obligāti jāizvēlas grūtība pirms ir pieja	Pirms nospiež "Play" ir jāizvēles līmeņa grūtība	1) Izvēlnē tiek izvēlēta dziesma 2) Tiek nospiesta play pirms un pēc grūtības izvēles	1) Dziesmas dati ir ievietoti iekšā Resorces/map-data folderī 2) kreisais klikšķis uz pogām	Ja nav izvēlēta grūtība tad "play" poga neko nedara, līdz ir izvēlēta grūtība	PR.IZV.06
TP.IZV.FA.05	Atvērt dziesmas datus spēles logā	Nospiežot play pogu tiek izlasīti kas ir saistīti ar tās dziesmas līmeni un tiek atvērta spēles sadaļa ar dziesmas līmeņa datiem	"Play" poga atver spēles logu ar līmeņa datiem	1) Ievietot dziesmas datus iekšā Resorces/map-data folderī 2) Atver programmu 3) Atver izvēlnes sadaļu 4) Izvēlēta dziesma 5) izvēlēta grūtība 6) nospiesta "Play poga"	1) Dziesmas līmeņa dati 2) Kreisais peles klikšķis	Atverās dziesmas līmenis	PR.IZV.07

13. attēls Testēšanas piemēri

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra apraksts	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra sagaidāmais rezultāts	Prasības ID
TP.IEST.VA.01	Mainīt ekrāna rezolūciju	Spiežot pa labi vai kreisi izmainīs skaitli kas reprezentē rezolūciju un saglabā šo skaitli xml failā. Mainot šo skaitli nemainīs ekrāna rezolūciju	Maina rezolūcijas iestatījumus	1) Atver iestatījumu lapu 2) Nospiež pa labi vai kreisi 3) Vērtība palielinās, vai samazinās	Kreisais peles klikšķis	Skaitli var mainīt, bet nekas cits nemainas	PR.IEST.01
TP.IEST.VA.02	Mainīt ekrāna tipu	Spiežot pa labi vai kreisi izmainīs nosaukumi kas reprezentē ekrāna tipu (windowed, borderless window, fullscreen) un saglabā šo vērtību xml failā	Maina ekrāna tipu iestatījumu	1) Atver iestatījumu lapu 2) Nospiež pa labi vai kreisi 3) Vērtība mainās	Kreisais peles klikšķis	Vērtību var mainīt, bet nekas cits nemainas	PR.IEST.02
TP.IEST.VA.03	Mainīt spēles kvalitāti	Spiežot pa labi vai kreisi izmainīs spēles kvalitāti no viss vienkāršākā, līdz viss sarežģītākajam un saglabā šo vērtību xml failā	Maina spēles kvalitātes iestatījumu	1) Atver iestatījumu lapu 2) Nospiež pa labi vai kreisi 3) Vērtība palielinās, vai samazinās	Kreisais peles klikšķis	Vērtība var mainīt, bet nekas cits nemainas	PR.IEST.03
TP.IEST.VA.04	Mainīt spēles skaļumu	Slīdni var palielināt vai samazināt kopējo spēles skaļumu, no 0% līdz 100%. Šis iestatījums mainās uzreiz pēc izmaiņas	Maina spēles kopējo skaļumu	1) Atver iestatījumu lapu 2) Mainot slīdni pa labi vai kreisi mainīs skaļumu	Kreisais peles klikšķis un turēšana	Mainot slīdņa vērtību mainīs visas spēles skaļumu	PR.IEST.04
TP.IEST.VA.05	Mainīt efektu skaļumu	Slīdni var palielināt vai samazināt efektu skaļumu, no 0% līdz 100%. Šis iestatījums mainās uzreiz pēc izmaiņas	Maina efektu skaļumu	1) Atver iestatījumu lapu 2) Mainot slīdni pa labi vai kreisi mainīs efektu skaļumu	Kreisais peles klikšķis un turēšana	Mainot slīdņa vērtību mainīs visas spēles skaļumu	PR.IEST.05
TP.IEST.VA.06	Mainīt mūzikas skaļumu	Slīdni var palielināt vai samazināt mūzikas skaļumu, no 0% līdz 100%. Šis iestatījums mainās uzreiz pēc izmaiņas	Maina mūzikas skaļumu	1) Atver iestatījumu lapu 2) Mainot slīdni pa labi vai kreisi mainīs mūzikas skaļumu	Kreisais peles klikšķis un turēšana	Mainot slīdņa vērtību mainīs visas spēles skaļumu	PR.IEST.06

14. attēls Testēšanas piemēri

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra apraksts	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra sagaidāmais rezultāts	Prasības ID
TP.IEST.FA.01	Akseptēt iestatījumus	Visi iestatījumi kuri bija tikai mainīti tikai vizuāli, pēc "Apply" nospiešanas tiek saglabāti un iestatīti kā ir noteikts iepriekšējos iestatījuma testa piemēros	Pēc nospiešanas visas iepriekš noteiktās vērtības tiek saglabātas un lietotas	1) Atver iestatījumus 2) Nomaina jebkuru iestatījumu kas nav skaļums. 3) Nospiež "Apply"	Kreisais peles klikšķi	Iepriekš noteiktās vērtības tiek iestatītas	PR.IEST.07
TP.SP.FA.01	Iespēja pauzēt spēli	Jebkurā momentā nospeiest "esc", lai pauzētu spēli un atvērt jaunu skatu	Jebkurā momentā var pauzēt spēli, lai mainītu iestatījumus vai iziet no līmeņa	1) Atvērt spēles sadaļu 2) nospiež "esc"	"esc" poga	Atver pauzēšanas ekrānu	PR.SP.01
TP.SP.FA.02	Izlasa dziesmas datus	Izlasa dziesmu kas ir Map-data folderī	Ielādējot jaunu līmeni tiek atskaņota dziesma	Atvērt spēles sadaļu ar dziesmas failu	Kreisais peles klikšķi	Mūzika tiek atskaņota	PR.SP.02
TP.SP.VA.01	Lādēšanās ekrāns	Ja faili aizņem pārāk ilgu laiku lai ielādētos ir jābūt redzamam ka spēle lādējas	Spēle rāda lādēšanās ekrānu	Pievienojot vairākus milzīgus failus "map-data" folderī	Kreisais peles klikšķi	Lādēšanās ekrāns parādas	PR.SP.03
TP.SP.FA.03.1	Nots vienklikšķa gandrīz netrāpīšanas gadījumā	Ja lietotāja spēles gaitā ir gandrīz pārgājusi nots pāri limitam un ir nospiesta pareizā poga tad jābūt samazinātiem punktiem (100p)	Ja lietotājs knapi trāpija notī tad iedod mazāk punktus	1) Atvērt līmeni 2) Izmantojot paredzamu laika klikšķi vai palēninātu spēli 3) Jānospiež pareizā poga	"Right", "Left", "up", "d", "a", "w", vai "space"	Parādas partikuls ar 100p simbolu, tiek iedoti 100p kopēja punktu skaitā, samazināta dzīvība, atskaņota efekta skaņa	PR.SP.04
TP.SP.FA.03.2	Gandrīz perfektā nots, vienklikšķa gadījumā	Ja lietotāja spēles gaitā ir gandrīz perfekti nospiesta pareizā poga tad jābūt iedotiem 200 punktiem	Ja lietotājs gandrīz dabuja perfektu notī, tiek dots viduvēji daudz punktu	1) Atvērt līmeni 2) Izmantojot paredzamu laika klikšķi vai palēninātu spēli 3) Jānospiež pareizā poga	"Right", "Left", "up", "d", "a", "w", vai "space"	Parādas partikuls ar 200p simbolu, tiek iedoti 200p kopēja punktu skaitā, atskaņota efekta skaņa	PR.SP.04
TP.SP.FA.03.3	Perfektā nots, vienklikšķa gadījumā	Ja lietotāja spēles gaitā ir perfekti nospieda pareizo pogu tad jābūt iedotiem 300 punktiem un mazliet dzīvību	Ja lietotājs dabuja perfektu notī, tiek dots maksimāli daudz punktu	1) Atvērt līmeni 2) Izmantojot paredzamu laika klikšķi vai palēninātu spēli 3) Jānospiež pareizā poga	"Right", "Left", "up", "d", "a", "w", vai "space"	Parādas partikuls ar 300p simbolu, tiek iedoti 300p kopēja punktu skaitā, 5% dzīvības ir atdots, atskaņota efekta skaņa	PR.SP.04

15. attēls Testēšanas piemēri

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra apraksts	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra sagaidāmais rezultāts	Prasības ID
TP.SP.FA.04	Nots turēšanas, gandrīz zaudēšanas gadījumā	Sākumā ir vienkārša, vienklikšķa nots, bet aiz tās ir redzama gara līnija.	Ja lietotājs perfekti nospiež vienklikšķa pogu, bet netur pogu nospiestu, tad ir iedoti minimālie garās notis punktus (300+0)	1) Atvērt līmeni 2) Izmantojot paredzamu laika klikšķi vai palēninātu spēli 3) Jānospiež pareizā poga 4) Uzreiz atlaist	"Right", "Left", "up", "d", "a", "w", vai "space"	Parādas vienklikšķa punktu partikuls un samazina dzīvību mazliet	PR.SP.05
TP.SP.FA.05	Nots turēšanas, gadījumā	Sākumā ir vienkārša, vienklikšķa nots, bet aiz tās ir redzama gara līnija.	Ja lietotājs perfekti nospiež vienklikšķa pogu un tad tur pog, katru "beat" tiks iedoti 50 ekstra punkti.	1) Atvērt līmeni 2) Izmantojot paredzamu laika klikšķi vai palēninātu spēli 3) Jānospiež pareizā poga 4) Turēt klikšķi nospiestu	"Right", "Left", "up", "d", "a", "w", vai "space"	Parādas vienklikšķa punktu partikuls un netiekmē dzīvību	PR.SP.05
TP.SP.FA.06	Notis iziet no ekrāna	Ja jebkura nots iziet no ekrāna tā tiek izdzēsta un 15% dzīvības tiek atņemti	Lietotājs nenospiež uz ekrānu	1) Ater līmeni 2) Skatās uz dzīvību	Šis ir piemērs kas notiek ja nespiež nevienu pogu spēles sākumā	Nots tiek izdzēsta un 15% dzīvības tiek atņemta	PR.SP.06
TP.SP.FA.07	Punktu palielināšana	Pēc punktu iegūšanas ir jāpārliciecinās ka tie tiek saglabāti	Punkti tiek vizuāli palielināti un ir redzami sēles beigās	1) Atver līmeni 2) Pēc līmeņa pabeigšanas tiek parādīts gala ekrāns kas rāda cik punkti tika iegūti	"Right", "Left", "up", "d", "a", "w", vai "space"	Punkti vienmēr ir vienādi sarp visiem gadījumiem kur ir redzami	PR.SP.07
TP.SP.FA.08	Pēc līmeņa beigām punkti ir redzami	Pēc līmeņa pabeigšanas tiek saglabāti visi dati par spēles gaitu un ir redzami izvēlnē	Punkti un reitings ir redams pēc spēles pabeigšanas	1) Atver līmeni 2) Pēc līmeņa pabeigšanas tiek parādīts gala ekrāns kas rāda cik punkti tika iegūti	"Right", "Left", "up", "d", "a", "w", vai "space"	Punkti un reitngs vienmēr ir vienādi sarp visiem gadījumiem kur ir redzami	PR.SP.08
TP.SP.FA.09	vienklikšķa būltas ievietošana	Līmeņa datos ir jābūt vienklikšķa nots pozīcija un tips noteikts, lai līmeņa gaitā var izlasīt un ievietot vienklikšķa būltu	Vienklikšķa būltas parādas līmeņos	1) Līmeņa datos ir noteikti vienklikšķi 2) Līmenī ir redzami vienklikšķi	Līmeņa dati	Līmeni ir ievietotas vienklikšķa būltas	PR.SP.09
TP.SP.FA.10	Turēšanas būltas ievietošana	Līmeņa datos ir jābūt vien nots pozīcija un tips noteikts, lai līmeņa gaitā var izlasīt un ievietot turēšanas būltu	Turēšanas būltas parādas līmeņos	1) Līmeņa datos ir noteikti turēšanas 2) Līmenī ir redzami turēšanas notis	Līmeņa dati	Līmeni ir ievietotas turēšanas būltas	PR.SP.10

16. attēls Testēšanas piemēri

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra apraksts	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra sagaidāmais rezultāts	Prasības ID
		būltu		notis			
TP.SP.FA.11	Nots ātruma noteikšana	Līmeņa datos var noteikt cik ātri nāk notis no aukšas	Nots ātrumu var noteiks un redzams spēlējot līmeņus	1)Līmeņa datos ir noteikts ātrums 2) notes ātrums ir redzams līmeņos	Līmeņa dati	Ja līmeņu datos ir lielāks ātrums noteikts tad būltas iet no aukšas uz leju ātrāk	PR.SP.11
TP.SP.FA.12	Nots BPM noteikšana	Līmeņa datos var noteikt cik daudz notis nāk sekundē	Nots BPM ir noteiks un redzams spēlējot līmeņus	1)Līmeņa datos ir noteikts ātrums 2) notes BPM ir redzams līmeņos	Līmeņa dati	Ja līmeņu datos ir BPM noteikts tad būltas iet no aukšas uz leju vairāk	PR.SP.12

17. attēls Testēšanas piemēri

Testēšanas ID	Datums	Testpiemērs ID	Testpeimēra nosaukums	Testētājs	Statuss	Kļūdas ziņojums	Kļūdas ziņojuma Nr.
Black box							
T.1	16.02.2024	TP.FA.01	Programmas startēšana	Kārlis Melveris	Veiksmīgs		
T.2	16.02.2024	TP.FA.02	Programmas apturēšana	Kārlis Melveris	Veiksmīgs		
T.sak.1	20.04.2024	TP.SĀK.FA.01	Programmas apturēšanas poga	Kārlis Melveris	Neveiksmīgs	Programma netiek apturēta	KZ.1
T.sak.2	20.04.2024	TP.SĀK.FA.02	Sākulapā lestatījuma poga	Kārlis Melveris	Veiksmīgs		
T.sak.3	20.04.2024	TP.SĀK.FA.03	Sākulapā spēlēt poga	Kārlis Melveris	Veiksmīgs		
T.IZV.1	10.05.2024	TP.IZV.VA.01	Izvēles izskata ģenerācija	Kārlis Melveris	Veiksmīgs		
T.IZV.2	10.05.2024	TP.IZV.VA.02	Izvēles dziesmas datu izskata ģenerācija	Kārlis Melveris	Veiksmīgs		
T.IZV.3	10.05.2024	TP.IZV.FA.01	Izvēles dziesmas datu izskata ģenerācija	Kārlis Melveris	Veiksmīgs		
T.IZV.4	11.05.2024	TP.IZV.FA.02	Izvēlnē atpakaļ poga	Kārlis Melveris	Veiksmīgs		
T.IZV.5	14.05.2024	TP.IZV.FA.03	Grūtības datu lasīšana	Kārlis Melveris	Veiksmīgs		
T.IZV.6	12.05.2024	TP.IZV.FA.04	Grūtības izvēles lmitācija	Kārlis Melveris	Veiksmīgs		
T.IZV.7	12.05.2024	TP.IZV.FA.05	Atvērt dziesmas datus spēles logā	Kārlis Melveris	Veiksmīgs		

18. attēls Testēšanas žurnāls

Testēšanas ID	Datums	Testpiemērs ID	Testpeimēra nosaukums	Testētājs	Statuss	Kļūdas ziņojums	Kļūdas ziņojuma Nr.
T.IEST.1	10.05.2024	TP.IEST.VA.01	Mainīt ekrēna rezolūciju	Kārlis Melderis	Veiksmīgs		
T.IEST.2	10.05.2024	TP.IEST.VA.02	Mainīt ekrēna tipu	Kārlis Melderis	Veiksmīgs		
T.IEST.3	10.05.2024	TP.IEST.VA.03	Mainīt spēles kvalitāti	Kārlis Melderis	Veiksmīgs		
T.IEST.4	10.05.2024	TP.IEST.VA.04	Mainīt spēles skaļumu	Kārlis Melderis	Veiksmīgs		
T.IEST.5	12.05.2024	TP.IEST.VA.05	Mainīt efektu skaļumu	Kārlis Melderis	Neveiksmīgs	netiek saglabāti dati un skaļums ir iestatīts vienmēr uz maksimālo skaļumu	KZ.2
T.IEST.6	12.05.2024	TP.IEST.VA.06	Mainīt mūzikas skaļumu	Kārlis Melderis	Neveiksmīgs	netiek saglabāti dati un skaļums ir iestatīts vienmēr uz maksimālo skaļumu	KZ.3
T.IEST.7	12.05.2024	TP.IEST.FA.01	Akseptēt iestatījumus	Kārlis Melderis	Neveiksmīgs	netiek saglabāti dati un skaļums ir iestatīts vienmēr uz maksimālo skaļumu	KZ.4
T.IEST.5	31.05.2024	TP.IEST.VA.05	Mainīt efektu skaļumu	Kārlis Melderis	Veiksmīgs		
T.IEST.6	31.05.2024	TP.IEST.VA.06	Mainīt mūzikas skaļumu	Kārlis Melderis	Veiksmīgs		
T.IEST.7	31.05.2024	TP.IEST.FA.01	Akseptēt iestatījumus	Kārlis Melderis	Veiksmīgs		
T.SP.1	04.06.2024	TP.SP.FA.01	Iespēja pauzēt spēli	Mats Dalehamn Sletvold	Veiksmīgs		
T.SP.2	16.03.2024	TP.SP.FA.02	Izlasa dziesmas datus	PNTA	Veiksmīgs		
T.SP.3	16.03.2024	TP.SP.VA.01	Lādēšanās ekrāns	PNTA	Veiksmīgs		
T.SP.4	12.05.2024	TP.SP.FA.03.1	Nots vienklikšķa gandrīz netrāpīšanas gadījumā	PNTA	Veiksmīgs		

19. attēls Testēšanas žurnāls

Testēšanas ID	Datums	Testpiemērs ID	Testpeimēra nosaukums	Testētājs	Statuss	Kļūdas ziņojums	Kļūdas ziņojuma Nr.
T.SP.5	12.05.2024	TP.SP.FA.03.2	Gandrīz perfektā nots, vienklikšķa gadījumā	PNTA	Veiksmīgs		
T.SP.6	12.05.2024	TP.SP.FA.03.3	Perfektā nots, vienklikšķa gadījumā	PNTA	Veiksmīgs		
T.SP.7	06.06.2024	TP.SP.FA.04	Nots turēšanas, gandrīz zaudēšanas gadījumā	Kārlis Meveris	Neveiksmīgs	Garās notis neparādas	KZ.5
T.SP.8	06.06.2024	TP.SP.FA.05	Nots turēšanas, gadījumā	Kārlis Meveris	Neveiksmīgs	Garās notis neparādas	KZ.6
T.SP.9	07.06.2024	TP.SP.FA.06	Notis iziet no ekrāna	Kārlis Meveris	Veiksmīgs		
T.SP.10	04.06.2024	TP.SP.FA.07	Punktu palielināšana	Mats Dalehamn Sletvold	Veiksmīgs		
T.SP.11	04.06.2024	TP.SP.FA.08	Pēc līmeņa beigām punkti ir redzami	Mats Dalehamn Sletvold	Neveiksmīgs	Netiek parāditi	KZ.7
White box							
T.SP.12	16.03.2024	TP.SP.FA.09	vienklikšķa bultas ievietošana	Kārlis Meveris	Veiksmīgs		
T.SP.13	16.03.2024	TP.SP.FA.10	Turēšanas bultas ievietošana	Thomas Hawke	Neveiksmīgs	Bultas nevar pievienot	KZ.8
T.SP.14	20.04.2024	TP.SP.FA.11	Nots ātruma noteikšana	Thomas Hawke	Veiksmīgs		
T.SP.15	20.04.2024	TP.SP.FA.12	Nots BPM noteikšana	Thomas Hawke	Veiksmīgs		

20. attēls Testēšanas žurnāls

Testēšanas ID	Kļūdas ziņojuma Nr.	Testpiemērs ID	Testpeimēra nosaukums	Testētājs	Kļūdas ziņojums
Black box					
T.sak.1	KZ.1	TP.SĀK.FA.01	Programmas apturēšana	Kārlis Meveris	Nekas nenotiek
T.IEST.5	KZ.2	TP.IEST.VA.05	Mainīt efektu skaļumu	Kārlis Meveris	netiek saglabāti dati un skaļums ir iestatīts vienmēr uz maksimālo skaļumu
T.IEST.6	KZ.3	TP.IEST.VA.06	Mainīt mūzikas skaļumu	Kārlis Meveris	netiek saglabāti dati un skaļums ir iestatīts vienmēr uz maksimālo skaļumu
T.IEST.7	KZ.4	TP.IEST.FA.01	Akseptēt iestatījumus	Kārlis Meveris	netiek saglabāti dati un skaļums ir iestatīts vienmēr uz maksimālo skaļumu
T.SP.7	KZ.5	TP.SP.FA.04	Nots turēšanas, gandrīz zaudēšanas gadījumā	Kārlis Meveris	Garās notis neparādas
T.SP.8	KZ.6	TP.SP.FA.05	Nots turēšanas, gadījumā	Kārlis Meveris	Garās notis neparādas
T.SP.11	KZ.7	TP.SP.FA.08	Pēc līmeņa beigām punkti ir redzami	Mats Dalehamn Sletvold	Netiek parādīti
T.SP.13	KZ.8	TP.SP.FA.10	Turēšanas būltas ievietošana	Thomas Hawke	Būltas nevar pievienot

21. attēls kļūdas ziņojumi

7. Individuālais ieguldījums

Šī nodaļa ir domāta, lai demonstrētu iegūto pieredzi taisot šo programmu. Tā būs sadalīta vairākās daļās, katra daļa ir domāta citam aspektam kurā es uzlaboju savas zināšanas. Kā piemēram 3D un 2D māksla, Unity un C# pieredze.

7.1. 2D mākslas pieredze

Lielāka daļa māksla nāca no komisijām, viņu info ir redzami 10 sekcijā (Literatūras un informācijas avotu saraksts). Es ieguvu pieredzi pareizi prasīt, izskaidrot un iedot references foto priekš jauniem mākslas darbiem. Lai izmantotu šos foto vajadzēja pārveidot, lai to darītu man vajadzēja izmantot Photoshop un Inkscape.

7.2 3D mākslas pieredze

3D objekti ir ļoti vienkārši šajā programma, bet es joprojām ieguvu pieredzi definēt un sakārtot UV mapes, materiālu sakārtošanu un “mirrored editing” kas ļauj man veidot simetriskus objektus vienkāršāk.

7.3 Unity pieredze

Mani galvenie ieguldījumi ir izmantojot Unity, UI taisīšana, trigeru izmantošana 3D vidē, laika apstrāde, ainas apstrāde un skriptu sadalīšana. Konkrēti saistībā ar Unity dzini, es vairāk iemācījos kā izmantot “Project Settings”, lai modificētu spēles sajūtu un kopējo izskatu.

7.4 C# pieredze

Pirms taisot šo projektu man bija ļoti minimāla pieredze ar jebkuru C versiju, tādēļ es ieguvu ļoti daudz pieredzi kā izmantot C# sintaksi un ģenerālu lietošanu Unity ekosistēmā.

7.5 Vispārējā pieredze

Manuprāt, es ieguvu pieredzi strādāt videospēles vidē un kā video spēļu industrija organizē projektus. Tas ir kopumā, gan kā māksla tiek organizēta, dokumentācijā tiek rakstīta izstrādes laikā un spēles dizains. Šīs prasmes ļoti palīdzēja manā praksē, kā arī varēju izmantot pieredzi no prakses lai izmantotu uz šī projekta.

8. Secinājumi

Galvenokārt, es Izvēlējos labu projekta ideju kurai man bija motivācija piestrādāt un uzlabot. Šī “rave” interese nav samazinājusies un vel joprojām mani interesē šī stila mūzika. Pie tam es vel joprojām izmantoju Unity, lai gan jaunāku versiju. Es izvēlējos 2.5D spēli kas ļoti izaicināja manu spēju integrēt abu veidu grafikus, kā arī kā tos modificēt.

Viss nebija perfekti, Unity versijā kuru es izmantoju ir ar dažām problēmām, vis īpaši ar daļiņu ģenerācijas sadaļu un UI izkārtojumu. Tie abi veicināja spēles izstrādi daudz grūtāku nekā bija plānots. Pat ja taisot 2.5D iedeva man pieredzi strādāt ar jaunām funkcijām un iedeva man daudz pieredzi, bet es nevarēju izmantot manu specializāciju, 3D mākslu. Ja es taisītu 3D spēli es arī varētu izrādīt vairāk savas prasmes rakstīt skriptus.

Kopumā es varēju izplānot labāku projekta ideju un vairāk ieskatīties maniem izstrādes instrumentiem, bet taisot šo projektu es iemācījos daudz vairāk kā izmantot Unity rīkus un 2D mākslas apstrādi. Kā arī man vel ir motivācija turpināt izstrādāt šo spēli ar vairāk funkcijām un jaunām tekstūrām.

9. Lietoto terminu un saīsinājumu skaidrojumi

1. 2.5D – Šis termins nozīmē ka spēle tiek rēķināta un taisīta kā 3d spēle, bet izmanto vairākus 2d elementus un visuāli izskatās gan 3D un kā 2D vienlaicīgi
2. Aizmugur lauks - Fotogrāfija vai grafiks kas ir aiz UI
3. Beat – Mūzikas termins kas norāda kad viena nots, vai pauze tiek spēlēta, tā kā šī ir ritma spēle, es izmantoju to pašu terminu, lai apzinātu laika fragmentu.
4. BPM – “Beats Per Minute” ir kā tek saglabāts mūzikas ātrums, jeb cik beat ir vienā minūtē, šajā dziesmā.
5. Bulta/nots – bulta vai nots tiks izmantoti kā sinonīmi. Tie būs objekti kuri reprezentēs katru beat.
6. Cyberpunk – māksliniecisks stils kas savieno kiber tehnoloģijas dizainus ar punk stiliem, cyberpunk ir vairākas subkultūras un stili.
7. Daļiņas – Daļiņas, jeb partikui ir vizuāli objekti, kā piemēram putekļi, uguns, dūmi vai efekti. Jebkurš objekts kas ir 3 dimensijās, bez 3D modeļiem.
8. Rave – Mūzikas koncerts/stils kas spēlē ātru elektronisko mūziku, pasti pārtaisītas populāras dziesmas kuras ir jauktas kopā un ar palielinātiem BPM
9. Spēles dzinis – Programma domāta lai veidotu videospēles
10. Spraits – Objekts ar foto kurš tiks izmantots spēles gaitā.
11. UI – “User Interface” ir interakcija ar 2d interaktīvām daļām kas izmanto peli.
12. PNTA – Saīsinājums “Prefer Not To Answer” vai “negrib rādīt identitāti” būs rakstīts jebkurā gadījumā kāds negrib rādīt identitāti.

10. Literatūras un informācijas avotu saraksts

- <https://docs.unity3d.com/Packages/com.unity.textmeshpro@2.2/manual/index.html>
- <https://docs.unity3d.com/2019.4/Documentation/Manual/>
- <https://docs.unity3d.com/2019.4/Documentation/ScriptReference/index.html>

Mākslinieki

- Raine, izveidoja nots dizainu, pogas un logo:
<https://www.instagram.com/togetherlast?igsh=MTRmNmKxMDFzbnYyMA==>
- Alexie, izveidoja mapes dizainu un dzīvību:
https://youtube.com/@alaxiety?si=NMLhPJn8c_cnDkHW

Pielikumi

1. Pielikums

```
public void Update()
{
    tempCheckChaiGe = GetComponent<TMP_Text>().text;
    string nameOfFolder = GetComponent<TMP_Text>().text;
    string filepath = Application.dataPath + "/Resources/map-data/" + nameOfFolder + "/scores.json";
    //Remove all the inicial menu items:
    foreach (Transform child in ContentOBJ.transform)
    {
        Destroy(child.gameObject);
    }
    //mkae the new list, if ther is any data
    if (File.Exists(filepath))
    {
        //get data out of the file
        ScoreData dataWrapp = JsonUtility.FromJson<ScoreData>(ReadAsString(filepath));
        int[] sorted = dataWrapp.Scores;
        Array.Sort(sorted);
        Array.Reverse(sorted);

        foreach (var item in sorted)
        {
            //add the new scores
            GameObject newscoreObj = Instantiate(presetForScore, ContentOBJ.transform);
            newscoreObj.GetComponent<TMP_Text>().text = "- " + item.ToString();
        }
        List<int> addingAValueList = new List<int>(dataWrapp.Scores);
        Debug.Log(tempCheckChaiGe);
    }
    else
    {
        GameObject newscoreObj = Instantiate(presetForScore, ContentOBJ.transform);
        newscoreObj.GetComponent<TMP_Text>().text = "- No Scores... Yet";
    }
}
```

1. attēls punktu lasīšana

2. Pielikums

```
void ArrowLaunch()
{
    if(k<=levelLength-2){
        k++;
        curStage = level[k];
        for (int i = 0; i <= 3; i++)
        {
            switch (Mathf.Floor(level[k] / Mathf.Pow(10, i) % 10))
            {
                case 1:
                    spriteMask.spawnArrows(i+1);
                    break;
                case 2:
                    spriteMask.spawnArrows(4);
                    i = 3;
                    break;
                default:
                    break;
            }
        }
    }
}
```

2. attēls līmeņa lasītājs

3. Pielikums

```
public class ChaingeScene : MonoBehaviour
{
    1 reference
    public string sceneName;
    1 reference
    public string sceneName1;
    1 reference
    public string sceneName2;
    0 references
    public void ChaingeScenes()
    {
        SceneManager.LoadScene(sceneName);
    }
    0 references
    public void ChaingeScenes1()
    {
        SceneManager.LoadScene(sceneName1);
    }
    0 references
    public void ChaingeScenes2()
    {
        SceneManager.LoadScene(sceneName2);
    }
}
```

3. attēls skates mainītājs

4. Pielikums

```
private string ReadAsString(string filePath)
{
    string returnString = null;
    try
    {
        if (File.Exists(filePath))
        {
            using (FileStream stream = new FileStream(filePath, FileMode.Open))
            {
                using (StreamReader reader = new StreamReader(stream))
                {
                    returnString = reader.ReadToEnd();
                }
            }
        }
        else
        {
            Debug.Log("NoScores");
        }
    }
    catch (System.Exception e)
    {
        Debug.LogError(e);
        throw;
    }
    return returnString;
}
```

4. attēls json lasītājs

5. Pielikums

```
void Update()
{
    if(Input.GetKeyUp(KeyCode.Escape) && MapValues.damageTaken<=1){
        isPaused = !isPaused;
        optionsMenu.SetActive(isPaused);
        if(!isPaused){
            settingsMenu.SetActive(false);
        }
    }
    if(isPaused){
        PlayerPrefs.SetInt("isPaused", 1);
        AudioSource.Pause();
        Time.timeScale = 0;
    }
    if(MapValues.damageTaken<=1 && !isPaused){
        PlayerPrefs.SetInt("isPaused", 0);
        Time.timeScale = 1;
        AudioSource.UnPause();
    }
}
```

5. attēls atvērt pauzes ekrānu

6. Pielikums

Pilnā projekta GitHub saite:

<https://github.com/kaaaaaaaaaaarl/CyberShock>

7. Pielikums

Testēšanas žurnāls:

<https://github.com/kaaaaaaaaaaarl/CyberShock/blob/main/Documentation/Test%C4%93%C5%A1anas%20dokuiment%C4%81cija.xlsx>