Liepājas Valsts tehnikums

**“SystemShock” videospēles**

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Personālā kvalifikācija ……………………………….

Darba autors: Kārlis Melveris

Liepāja 2024

# Saturs

[Ievads 5](#_Toc153965003)

[1. Uzdevuma formulējums 6](#_Toc153965004)

[2. Programmatūras prasību specifikācija 8](#_Toc153965005)

[2.1. Produkta perspektīva 8](#_Toc153965006)

[2.2. Sistēmas funkcionālās prasības 9](#_Toc153965007)

[2.2.1. Izvēlēties līmeni poga 9](#_Toc153965008)

[2.2.2. Iestatījuma izvēles atvēršanas poga 9](#_Toc153965009)

[2.2.3. Iziet poga 9](#_Toc153965010)

[2.2.4. Izvēlēties grūtības iestatījumu logs 10](#_Toc153965011)

[2.2.5. Izvēlēties grūtības iestatījumu 10](#_Toc153965012)

[2.2.6. Sākt spēli poga. 10](#_Toc153965013)

[2.2.7. Pauzēšana 11](#_Toc153965014)

[2.2.8. Atsākt spēli poga 11](#_Toc153965015)

[2.2.9. Iziet no līmeņa 11](#_Toc153965016)

[2.3. Sistēmas nefunkcionālās prasības 12](#_Toc153965017)

[2.3.1. Izstrādes līdzekļi 12](#_Toc153965018)

[2.3.2. Veikt spējas prasības 12](#_Toc153965019)

[2.3.3. Vizuāls izskats 12](#_Toc153965020)

[2.3.4. Skaņu dizains 12](#_Toc153965021)

[2.3.5. Modificējamība 13](#_Toc153965022)

[2.4. Gala lietotāja raksturiezīmes 13](#_Toc153965023)

[3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums 14](#_Toc153965024)

[3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts 14](#_Toc153965025)

[3.2. Iespējamo risinājuma līdzekļu un valodu apraksts 14](#_Toc153965026)

[4. Sistēmas modelēšana un projektēšana 15](#_Toc153965027)

[4.1. Sistēmas struktūras modelis 15](#_Toc153965028)

[4.2. Klašu diagramma 15](#_Toc153965029)

[4.3. ER diagramma 15](#_Toc153965030)

[4.4. Funkcionālais un dinamiskais sistēmas modelis 15](#_Toc153965031)

[4.5. Aktivitāšu diagramma 15](#_Toc153965032)

[4.6. Lietojumgadījumu diagramma 15](#_Toc153965033)

[4.7. Sistēmas moduļu apraksts un algoritmu shēmas 15](#_Toc153965034)

[5. Lietotāju ceļvedis 16](#_Toc153965035)

[6. Testēšanas dokumentācija 17](#_Toc153965036)

[6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums 17](#_Toc153965037)

[6.2. Testpiemēru kopa 17](#_Toc153965038)

[6.3. Testēšanas žurnāls 17](#_Toc153965039)

[7. Individuālais ieguldījums 18](#_Toc153965040)

[8. Secinājumi 19](#_Toc153965041)

[9. Lietoto terminu un saīsinājumu skaidrojumi 20](#_Toc153965042)

[10. Literatūras un informācijas avotu saraksts 21](#_Toc153965043)

# Ievads

Šis ievads ir domāts, lai iedotu vienkāršu aprakstu par “SystemShock” 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 ievienošanai.

Projekta veidošanas laikā tiek dokumentēti visas kļūdas un problēmas kas izveidojas projekta taisīšanas laikā, kā arī tiks dokumentēts 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.

SystemShock 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, izdzēš, 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 uzsvērta kļūdu labošana, optimizācija un stratēģiskā izlaišana ar pastāvīgu atbalstu, izmantojot atjauninājumus un kopienas iesaisti.

Spēles SystemShock 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.

# 2. Programmatūras prasību specifikācija

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, utt.

## 2.1. Produkta perspektīva

No sākuma es izskaidrošu šī projekta spēli. Šī ir ritmu spēle, tādēļ seko ierasto ritmu spēles formātu, kā spēlētājs tavs uzdevums ir nospiest 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 iespējamos 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 šis nokrīt ir jābūt muzikāla instrumenta skaņa vai vienkārši jābūt ritmā ar mūziku. Tad lietotāja objektīvs ir nospiest, 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 pauzēt un mainīt iestatījumus vai iziet no līmeņa. Lai gan tavs 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.

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 sadaļa.

Apstrāde:

Pēc nospiešanas lietotājs tiek pārnests uz Iestatījumu izvēlnes.

### 2.2.3. Iziet 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 kursors nospiež uz jebkuru līmeni.
2. Konkrētā līmeņa noteiktie dati JSON faila.

Nosacījumi:

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

Apstrāde:

Pēc nospiešanas paradās 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. Peles kursors nospiež uz kādu no grūtības iestatījumiem.

Nosacījumi:

1. Nav iespējams noklikšķināt uz grūtības līmeni kas nefunkcionē.
2. Paradās tikai spēles grūtības iestatījuma sadaļā.

Apstrāde:

Pēc nospiešanas sākt spēli poga tiek atvērta un var nospiest. 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 kursors 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 nospiest, ja lietotājs nav izvēlējiem 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 progresa.

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. Iziet no līmeņa

Mērķis:

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:

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

## 2.3. Sistēmas nefunkcionālās prasības

### 2.3.1. Izstrādes līdzekļi

Šī spēle tiks izstrādāta Unity spēles dzina, 2019.4.31f versijā, tādēļ tiks izmantota C# programmēšanas valoda. Lai saglabātu šo projektu caur vairākām ierīcēm tiks izmantots Github projekts, kurā viss saistīts ar šo projektu tiks saglabāts. Lai modificētu spēles kodu tiks izmantots “Visual Studio code”.

### 2.3.2. Veikt spējas prasības

Spēlei ir jāstrādā ar vismaz 5 gadu veciem datoru standartiem, vai jebkuru datoru ar Gtx 1050 grafikas kartes un i7-6800K CPU ekvivalentiem. Lai gan ar ir jābūt grafisku intensitātes un efektu iestatījumi, lai pārliecinātos ka vecāki, vai lētāki datori var spēlēt.

### 2.3.3. 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.

### 2.3.4. Skaņu dizains

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

### 2.3.5. 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 tekstūra.

## 2.4. Gala lietotāja raksturiezīmes

# 3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums

## 3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts

## 3.2. Iespējamo risinājuma līdzekļu un valodu apraksts

# 4. Sistēmas modelēšana un projektēšana

## 4.1. Sistēmas struktūras modelis

## 4.2. Klašu diagramma

## 4.3. ER diagramma

## 4.4. Funkcionālais un dinamiskais sistēmas modelis

## 4.5. Aktivitāšu diagramma

## 4.6. Lietojumgadījumu diagramma

## 4.7. Sistēmas moduļu apraksts un algoritmu shēmas

# 5. Lietotāju ceļvedis

# 6. Testēšanas dokumentācija

## 6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums

## 6.2. Testpiemēru kopa

## 6.3. Testēšanas žurnāls

# 7. Individuālais ieguldījums

# 8. Secinājumi

# 9. Lietoto terminu un saīsinājumu skaidrojumi

# 10. Literatūras un informācijas avotu saraksts