



OpenXR-sovelluskehitys Unity-pelimoottorilla

Janne Kallio

Opinnäytetyö, AMK

Huhtikuu 2021

Tietojenkäsittely ja tietoliikenne

Insinööri (AMK), tieto- ja viestintäteknikka

Kallio, Janne

OpenXR-sovelluskehitys Unity-pelimoottorilla

Jyväskylä: Jyväskylän ammattikorkeakoulu. Huhtikuu 2021, 44 sivua.

Tietojenkäsittely ja tietoliikenne. Insinööri (AMK), tieto- ja viestintätekniikka.

Julkaisun kieli: suomi

Verkkojulkaisulupa myönnetty: kyllä

Tiivistelmä

VR-sovellukset ovat lisääntyneet huimaa vauhtia ja niiden käyttö on yleistynyt sekä liike-elämässä että viihdetuotannossa. Erilaiset teknologiat ovat samalla kehittyneet entistä käyttäjäystävällisemmiksi ja ovat myös entistä helpommin saatavilla kuluttajille. Erityisesti visuaalinen ohjelmointi pelien kehittämisessä on alkanut yleistyä samalla kun videopelaamisen suosio kasvaa.

Tehtävänä oli tuottaa pelattava demo, jonka avulla tutkittiin OpenXR-standardin ja Unity-pelimoottorin soveltuvuutta sovelluskehityksen työvälineiksi. Lisäksi tutkittiin visuaalisen ohjelmoinnin Bolt-työkalua. Näiden soveltuvuutta havainnoitiin kehitystyön aikana ja lisäksi demon testasivat testipelaajat arvioiden käyttäjäkokemusta.

Toteutustavaksi valikoitui pakohuoneteemainen peli VR-ympäristössä, jossa pelin kulku perustui pelaajan ratkaisemiin pulmiin. Peli sijoitettiin virtuaaliseen poliisiasemaympäristöön. Yksittäisissä pulmissa tutkittiin pelimoottorissa olevan fysiikkamoottorin rajoitteita sekä visuaalisen ohjelmoinnin tilakoneiden ominaisuuksia. Demon testausta varten suunniteltiin loppukäyttäjättestaus, jonka aikana pelaajien pelikertoja havainnoitiin ja aikataulutettiin. Lisäksi testipelaajat vastasivat käyttäjäkokemusta käsitteleviin kysymyksiin. Näiden pohjalta kyettiin arvioimaan demon onnistumista käyttäjäkokemuksen kannalta.

Kehitystyön tuloksena saatiin havaintoja OpenXR-standardin toimivuudesta, mikä osoittautui virheettömäksi ja työhön soveltuvaksi. Myös Unityn kyky toimia pelimoottorina osoittautui hyväksi ja käyttöliittymä koettiin selkeäksi. Visuaalisen käyttöliittymän ohjelmointityökalu Bolt poisti tarpeen perinteiselle aikaa vaativalle syntaksin opiskelulle, joten pääpaino pysyi pelin kehittämisessä. Erityisesti vikojen etsimiseen suunnitellut ominaisuudet koettiin hyödyllisiksi. Käyttäjäkokemuksen osalta pelin suunnittelu onnistui halutulla tavalla, eikä VR-laitteiden käyttö aiheuttanut ongelmia. Testipelaajilta saatu data oli arvokas tiedonlähde ja vahvisti käyttäjäkokemusta tutkivien menetelmien toimivuutta hyvän pelikokemuksen suunnittelussa.

Avainsanat (asiasanat)

Pelinkehitys, käyttäjäkokemus, pelillistäminen, virtual reality, ohjelmistokehitys

Muut tiedot (salassa pidettävät liitteet)

Kallio, Janne

OpenXR software development with Unity game engine

Jyväskylä: JAMK University of Applied Sciences, April 2021, 44 pages.

Information and Communication Technologies. Bachelor's Degree Programme in Information and Communications Technology.

Permission for web publication: Yes

Language of publication: Finnish

Abstract

VR-applications have grown at a tremendous rate and they have become more common in both business and entertainment production. At the same time, various technologies have also become more user-friendly and are more easily accessible to consumers. Especially visual programming has become more common in game development as the popularity of video gaming keeps growing.

The task was to produce a playable demo. It was used to study the suitability of the OpenXR standard and the Unity game engine as application development tools. In addition, the Bolt tool for visual programming was studied. The suitability of these was observed during the development and in addition the demo was tested by test players evaluating the user experience.

An escape room-themed game in the VR environment was chosen as the implementation method, where the course of the game was based on puzzles solved by the player. The game was placed in a virtual police station environment. In individual problems, the limitations of the physics engine in the game engine and the properties of the visual programming flow machines were studied. In order to test the demo, end-user testing was designed. During it players' game experiences were observed and scheduled. In addition, test players answered few questions about their playing experience. Based on these, it was possible to evaluate the success of the demo in terms of user experience.

As a result of the development work, observations were made about the functionality of the OpenXR standard. Standard proved to be error-free and suitable for the project. Unity's ability to act as a game engine also proved to be good and the user interface was clear. The visual interface programming tool Bolt eliminated the need for traditional time-consuming syntax learning, so the focus remained on game development. Features designed specifically for debugging were found to be useful. In terms of user experience, the design of the game succeeded as desired, and the use of VR devices did not cause any problems. The data obtained from test players was a valuable source of information and confirmed the functionality of user experience research methods in designing a good gaming experience.

Keywords/tags (subjects)

Game design, gamification, user experience, virtual reality, software development

Miscellaneous (Confidential information)

Sisältö

1	Johdanto	4
1.1	Opinnäytetyön tausta	4
1.2	Tavoitteet ja rajaukset	4
1.3	Menetelmät ja kehittämisympäristö	5
1.4	Tutkimusasetelma	5
2	Unity ja käytettävät resurssit	6
2.1	Unity	6
2.2	OpenXR.....	7
2.3	Resurssipaketit	7
2.3.1	QA InterrogationRoom ja QA Police Station.....	8
2.3.2	Flashlight PRO	8
2.3.3	Auto Hand	8
3	Bolt	8
3.1	Bolt visuaalisen ohjelmoinnin työkaluna	8
3.2	Boltin tilakaaviot	9
3.2.1	Tilakaavion lukeminen	9
3.2.2	Flow machine -tilakonetyyppi.....	11
3.2.3	Super unit tilakonetyyppien apuna	11
3.2.4	State machine -tilakonetyyppi	13
4	Kehitystyön tavoite	14
4.1	Pelin konsepti	14
4.2	Kehitystyön tavoitteet ja onnistumisen mittarit.....	14
5	Peliympäristön suunnittelu	15
5.1	Pelin teema ja aloitusvalikko.....	15
5.2	Alakerta	16
5.2.1	Selliosasto	16
5.2.2	Vankien käsittelytila.....	17
5.2.3	Portaikko.....	18
5.3	Yläkerta.....	20
5.3.1	Taukotila	20
5.3.2	Toimisto	21
5.3.3	Suunnitteluhuone	22
6	Pelillistävät pulmat	22
6.1	Pulmien pelillistävä tarkoitus	22

6.2	Pulmien esimerkkitoteutuksia	23
6.2.1	Numerolukko	23
6.2.2	Ovien avaaminen avaimella	26
6.2.3	Tikkataulu.....	27
6.2.4	Videon avaaminen televisioon kaukosäätimellä	30
7	Sovellustestausten analysointi	31
7.1	Sovellustestausten käytännön toteutus	31
7.2	Yhteenveto sovellustestausten tuloksista	32
8	Opinnäytetyön tulokset ja onnistumisen arviointi	34
	Lähteet	37
	Liitteet	39
	Liite 1. Pelitestausten tietojenkeruulomake	39
	Liite 2. Pelaajan 1 pelitestausta.....	41
	Liite 3. Pelaajan 2 pelitestausta.....	42
	Liite 4. Pelaajan 3 pelitestausta.....	43
	Liite 5. Pelaajan 4 pelitestausta.....	44
	Kuviot	
	Kuvio 1. Yksinkertainen esimerkki Flow machine -tilakoneen toiminnasta	10
	Kuvio 2. Esimerkkikuva timer -funktiosta	10
	Kuvio 3. Esimerkkitoteutus Super unit:n käytöstä	12
	Kuvio 4. Super unit:n sisältö.....	12
	Kuvio 5. Esimerkkikuva State Machinen toiminnasta.....	13
	Kuvio 6. Pelin aloitusvalikkona toimiva kuulusteluhuone	16
	Kuvio 7. Selliosastolla kahden vangin selli	16
	Kuvio 8. Selliosaston käytävä	17
	Kuvio 9. Käsittelytilan aula	18
	Kuvio 10. Portaikon alapuoli	19
	Kuvio 11. Portaikon yläpuoli	19
	Kuvio 12. Taukuhuoneen nurkkatila	20
	Kuvio 13. Yläkerrassa sijaitseva toimisto	21
	Kuvio 14. Yläkerran suunnitteluhuone	22
	Kuvio 15. Esimerkkikuva yläkerrassa sijaitsevasta numerolukosta	23
	Kuvio 16. Yleiskuva Boltin logiikasta numerolukon takana, mistä voidaan havaita tilakoneen laajuus	24

Kuvio 17. Logiikka nappien painalluksessa	25
Kuvio 18. Logiikka lukon resetoinnissa	26
Kuvio 19. Logiikka koodin tarkastamisessa	26
Kuvio 20. Logiikka oven avaamiseen avaimella	27
Kuvio 21. Tikkataulun logiikkatilakone kolmelle tikalle	28
Kuvio 22. Tilakoneen logiikka tikan asettamisessa	28
Kuvio 23. Logiikka tikkataulun kohtien arvojen vertaamisessa	29
Kuvio 24. Logiikka tikan pysymiseen tikkataulussa.....	30
Kuvio 25. Kaukosäätimen On Squeeze määrittäminen.....	30
Kuvio 26. Kuva videosta televisiossa aktivoinnin jälkeen	31

Taulukot

Taulukko 1. Pelaajien vastaukset analysoituna yhteenvetotaulukossa.....	32
Taulukko 2. Pelin kulkuun liittyvä statistiikka analysoituna yhteenvetotaulukossa.....	33

1 Johdanto

1.1 Opinnäytetyön tausta

Virtuaalista todellisuutta käyttävien sovellusten lukumäärä on ollut räjähtävässä kasvussa useiden vuosien ajan. Näitä sovelluksia on valjastettu laajasti niin kaupalliseen kuin viihteelliseen käyttöön. Liike-elämässä tällaisten sovellusten kysyntä tulee kasvamaan merkittävästi. Suosion hiipumista ei myöskään ole odotettavissa lähitulevaisuudessa.

Opinnäytetyöntekijän mielenkiinnosta johtuen valittiin aiheeksi kehitystyö Unity-pelimoottorilla pohjautuen pitkään kokemukseen pelaamisen parissa. Aihe on todella mielenkiintoinen ja koettiin, että sen tutkiminen tukee opinnäytetyön tekijän ammatillista osaamista.

Toinen kiinnostava trendi on ns. pakohuonepelaaminen, joka perustuu erilaisten loogisten ongelmien ratkaisuun. Opinnäytetyöntekijän kokemus molemmista aihepiireistä antoi ajatuksen suunnata opinnäytetyö viihdekäyttöön tarkoitettuun sovellukseen, jonka painopiste on loogisten tehtävien ratkaisemisessa.

1.2 Tavoitteet ja rajaukset

Opinnäytetyön tavoitteena on tuottaa pelattava demo. Pelin tavoitteena ovat riittävä haasteellisuus ja mielenkiinto myös aikuiselle pelaajalle. Teknisinä tavoitteina ovat pelin toimivuus OpenXR-standardin laitteilla sekä hyvä käyttäjäkokemus. Käyttäjäkokemuksen onnistumista mitataan demon avulla järjestettävissä loppukäyttäjien pelitestauksissa.

Tekninen rajaus tehtiin OpenXR-standardiin sen skaalautuvuuden vuoksi sekä aikaisemman osaamisen perusteella Unity-pelimoottoriin. Tavoitteena oli, että demo ei laajene liikaa, mutta se tarjoaa silti mielenkiintoisen pelikokemuksen. Näistä syistä demo rajattiin yhden henkilön pelattavaan versioon ja pulmien määrä kymmeneen. Laadullisin tutkimusmenetelmin saadun materiaalin määrä rajattiin neljällä eri loppukäyttäjällä suoritettavaan pelitestiin.

1.3 Menetelmät ja kehittämisympäristö

Opinnäytetyön mahdollisti VR-ympäristön ja lisenssien olemassaolo sekä niistä saatu käyttökokeemus. Opinnäytetyönä tehty demo ja sen testaus oli näin ollen mahdollista suorittaa opinnäytetyöntekijän omassa virtuaalisen todellisuuden studiossa. Myös Unity-pelimoottoriin tarvittavat lisenssit ja käyttöoikeudet resursseihin olivat olemassa jo ennen opinnäytetyön aiheen rajaamista. Käyttäjäkokemuksen mittaamisessa käytettyyn pelitestaukseen valittiin laadullinen tutkimusmenetelmä. Siinä loppukäyttäjä pelaa pelin itsenäisesti alusta loppuun. Testin aikana seurataan mahdollisia ongelmatilanteita, erilaisia loppukäyttäjän huomioita, kellotetaan pelin kulku sekä merkitään pelaajan ottamat vihjeet. Testin lopuksi testattava vastaa muutamaa kysymykseen pelikokemukseen liittyen. Täten laadullisella tutkimusmenetelmällä saadaan riittävä kokonaiskuva demon käyttäjäkokemuksesta.

1.4 Tutkimusasetelma

Opinnäytetyön tutkimuksellisenä ongelmana on halu ymmärtää OpenXR-standardin sekä Boltin soveltuvuutta Unity-pelimoottorilla tuotettuun VR-sovellukseen. Toisena tutkimuskohteena on tutkia, saadaanko näillä teknologioilla tuotettua viihdekäytössä toimiva pelidemo, jonka yksityiskohdat mahdollistavat hienomotoriikkaa käyttäjäkokemuksen siitä huonontumatta. Opinnäytetyössä pyritään vastaamaan seuraaviin kysymyksiin:

- * Miten OpenXR-standardi sekä Bolt soveltuvat Unity-pelimoottorilla tuotettuun VR-sovellukseen?
- * Millainen käyttäjäkokeemus rakentuu opinnäytetyön VR-pelidemosta?

Ensimmäiseen tutkimuskysymykseen vastaus saadaan suunnittelemalla ja toteuttamalla VR-pelidemo valituilla komponenteilla ja halutuilla toiminnallisuuksilla. Opinnäytetyöntekijän kokeilee, suunnittelee ja arvioi valittujen komponenttien käytännöllisyyttä, toimivuutta ja soveltuvuutta ja analysoi havaintojaan opinnäytetyön lopputuloksissa. Toiseen tutkimuskysymykseen vastamista varten toteutetaan edellä kuvattu laadulliseen tutkimusmenetelmään perustuva sovellustestaus. Valitut tutkimusmenetelmät antavat hyvää kokemusperäistä tutkimustietoa, mutta eivät ole tilastollisesti todennettavia.

2 Unity ja käytettävät resurssit

2.1 Unity

Unity on pelimoottori, jonka tavoitteena on tuoda käytännöllinen ja tehokas sovelluskehitys jokaisen aiheesta kiinnostuneen sovelluskehittäjän ulottuville. Unity on yksittäiselle käyttäjälle ilmainen, mutta mikäli kehittäjän tuottama sovellus tuottaa rahaa yli tietyn määrän, myös Unityn lisenssi muuttuu maksulliseksi. Immateriaalioikeudet kuuluvat kuitenkin aina kehittäjälle itselleen, ellei sopimuksella erikseen päätetä muuta.

Unitylla on mahdollista kehittää sovelluksia laajasti eri alustoille sekä tuottaa mm. pelejä. Sen avulla pystyy kehittämään pelejä perinteisesti tietokoneelle mm. Windows-, Mac- sekä Linux-käyttöjärjestelmille sekä pelien kehittäminen onnistuu myös perinteisille konsoleille kuten mm. Playstation 4 sekä Xbox One. Lisäksi pelejä on mahdollista kehittää myös eri web-selaimissa toimiviksi. Perinteisten pelikoneiden lisäksi Unitylla on mahdollista kehittää sovelluksia myös muille laitteille kuten puhelimille tai televisioille. Puhelimen käyttöjärjestelmistä soveltuvia kehittämiselle ovat Android- sekä iOS-pohjaiset laitteet.

Unityn omien tilastojen mukaan vuonna 2020 ladattiin viisi miljardia sillä tehtyä sovellusta kuukaudessa. Tuhannesta suosituimmasta mobiilipelistä 71 prosenttia oli tuotettu Unityn avulla ja kaikista mobiili-, tietokone- sekä konsolipeleistä Unitylla oli tehty 50 prosenttia. Unity Solutions -palvelussa oli kaksi ja puoli miljardia aktiivista käyttäjää kuukausittain. He kuluttivat, loivat tai operoivat sisältöä. Vuonna 2020 Unitylla tuotettuja sovelluksia tuki yli 20 erilaista alustaa ja sovelluksia kehitettiin sen avulla yli 190 maassa. Suosittuja Unitylla kehitettyjä pelejä ovat mm. Kerbal Space Program, Hearthstone: Heroes of Warcraft, Escape from Tarkov, Cuphead sekä Rust. (Unity Technologies 2021.)

Vaikka Unity mielletään pelimoottoriksi, on sillä myös monia muita sovelluskehittämisen käyttökohteita. Sen avulla on mahdollista tuottaa mm. liike-elämää palvelevia tuotantokäyttöön suunniteltuja sovelluksia, erilaisia animaatioita, luovien alojen mainos- ja ilmoitusratkaisuja sekä viihdemaailman rahapelien sovelluksia.

Lisäksi Unity tarjoaa käyttäjilleen myös muita palveluita. Merkittävimpiä niistä ovat mm. Unity Asset Store sekä Unity Learn. Unity Asset Store -palvelu mahdollistaa kehittäjälle muiden kehittäjien luomusten lataamisen. Näitä voivat olla vaikka erilaiset skriptit tai 3D-mallit. Palvelussa voi myös jakaa omia tuotoksiaan ilmaiseksi muille kehittäjille tai myydä palvelun sisäisesti käyttöoikeuden omiin tuotoksiin rahaa vastaan. Unity Learn -palvelussa on Unityn henkilökunnan sekä muiden käyttäjien luomia oppimiskokonaisuuksia, joiden tasot vaihtelevat laajasti aloittelijasta aina ammattilaistasoon saakka.

2.2 OpenXR

OpenXR on Unitysta erillinen rojalteista vapaa avoin standardi. OpenXR:n päätehtävänä on toimia rajapintana, jonka avulla on mahdollista kehittää virtuaalisen todellisuuden sekä lisätyn todellisuuden sovelluksia. OpenXR sijoittuu siis työkaluna tarkemmin valmiin loppusovelluksen ja pelimoottorin väliin.

OpenXR on alun perin Khronos Group Inc:n kehittämä ja ylläpitämä työkalu. Sitä on mahdollista hyödyntää kuitenkin Unitylla sovelluskehityksessä valmiiksi tehdyn "Unity OpenXR"-liitännäisen avulla, joka on ladattavissa Unity-pelimoottoriin. OpenXR-standardin mittava hyöty saadaan sen soveltuvuudesta monen eri laitevalmistajan VR-laitteille. OpenXR on yhteensopiva mm. HTC, Microsoft, Oculus sekä SteamVR-tuotteiden kanssa. Tästä syystä OpenXR helpottaa oikein implementoituna sovelluskehittäjän työtä merkittävässä määrin, jos tavoite on sovelluksen toimiminen useiden eri laitevalmistajien laitteilla.

2.3 Resurssipaketit

Unityn Asset Store -palvelu tarjoaa useita erilaisia resursseja. Tässä kappaleessa esitellään tarkemmin jokaista opinnäytetyössä kehitetyn pelin ympäristön ja pelaajan luontiin käytettyä resurssia.

2.3.1 QA InterrogationRoom ja QA Police Station

Resurssien etuliite QA viittaa niiden alkuperäiseen tekijään, jonka käyttäjänimi on "QAtmo". Nämä resurssipaketit sisältävät useita uniikkeja esineitä, erilaisia ovia sekä useita rakenteellisia malleja, kuten seiniä ja kattoja. Nämä resurssipaketit valittiin tähän projektiin yleisen visuaalisen ilmeen soveltuvuuden vuoksi.

2.3.2 Flashlight PRO

Flashlight PRO -niminen resurssipaketti on lisenssiltään ilmainen ja ladattavissa Unity Asset Store -palvelusta. Tämä resurssipaketti sisältää yhden taskulampun ja pariston 3D-mallit sekä skriptin, joka liittyy taskulampun toimintaan. Kyseinen resurssipaketti valittiin visuaalisten ominaisuuksien vuoksi käytettäväksi kehitystyössä.

2.3.3 Auto Hand

Auto Hand on Unityn Asset Store -palvelusta ostettu resurssipaketti. Se tarjoaa valmiin pohjan, joka on suunniteltu toimimaan mm. OpenXR-liitännäisen kanssa. Auto Hand-resurssipaketilla on siis mahdollista luoda pelaajan realistinen liikkuminen sekä implementoida hänen vuorovaikutuksensa ympäristön kanssa. Tämä resurssipaketti tarjoaa myös laajan dokumentaation sekä lukuisia valmiita esimerkkejä. Nämä esimerkit sisältävät mm. toimivia ovia, vipuja, nappuloita sekä ohjainpyöriä.

3 Bolt

3.1 Bolt visuaalisen ohjelmoinnin työkaluna

Bolt on visuaalinen ohjelmoinnin käyttöliittymä. Se on lisätty kaikkiin Unityn versioihin vuonna 2020. Boltin perimmäinen tarkoitus on mahdollistaa skriptien luonti kehittäjille, joilla ei ole ohjelmointitaitoa. Boltin tuottama skripti ei kuitenkaan logiikaltaan eroa merkittävästi perinteisestä Unityn käyttämästä C#-ohjelmointikielestä. Niiden työkalut ja logiikka ovat hyvin lähellä toisiaan,

mutta kehittäjän ei tarvitse osata syntaksin kirjoittamista perinteisellä tavalla. Visuaalisen käyttöliittymän on todettu olevan myös hyvä työkalu ohjelmoinnin oppimiseen. Osa kehittäjistä on ker-tonut logiikan hahmottamisen olevan helpompaa tilakoneen muodossa. Esimerkkejä tilakoneista löytyy tämän opinnäytetyön kuvioista 16–24.

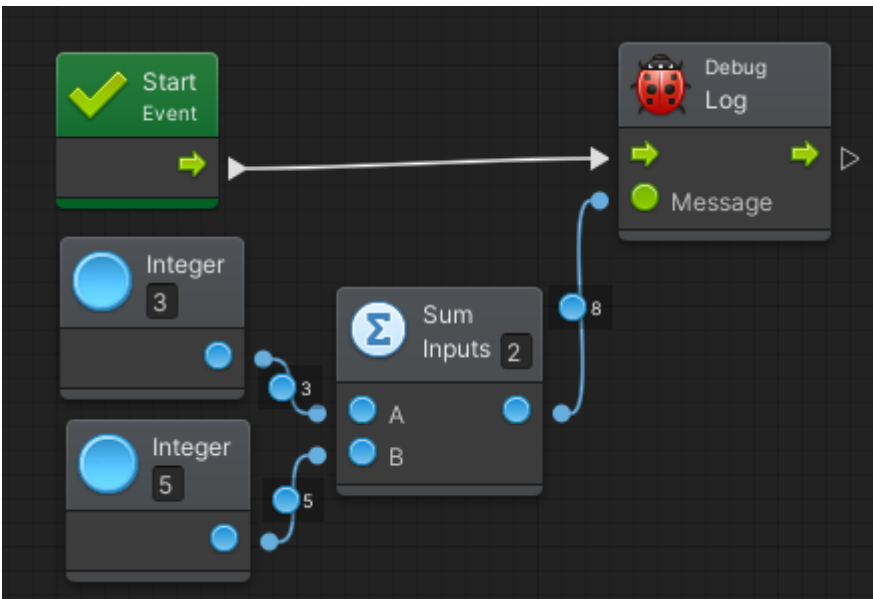
Bolt tarjoaa kehittäjille myös mahdollisuuden jakaa ja keskustella sen avulla tehdyistä sovelluksista tai komponenteista. Kehittäjän tukena on lisäksi massiivinen Boltin henkilökunnan laatima dokumentaatio liittyen sen toimintaan ja käyttöön. Lisäksi tarjolla on useita oppimiskokonaisuuksia eri-laisten käytännön projektien parissa.

3.2 Boltin tilakaaviot

Bolt ohjaa pelissä tapahtuvaa logiikkaa tilakoneiden avulla. Tässä kappaleessa perehdytään tarkemmin tilakoneiden teoriaan ja toimintaan käytännön esimerkkien kautta. Kappaleen tavoite on helpottaa ymmärtämään opinnäytetyössä myöhemmin esiteltyjä tilakoneita. Tilakoneita suunnitellessa on tärkeää hahmottaa kaksi erilaista tilakoneen tyyppiä, joita ovat Flow machine sekä State machine.

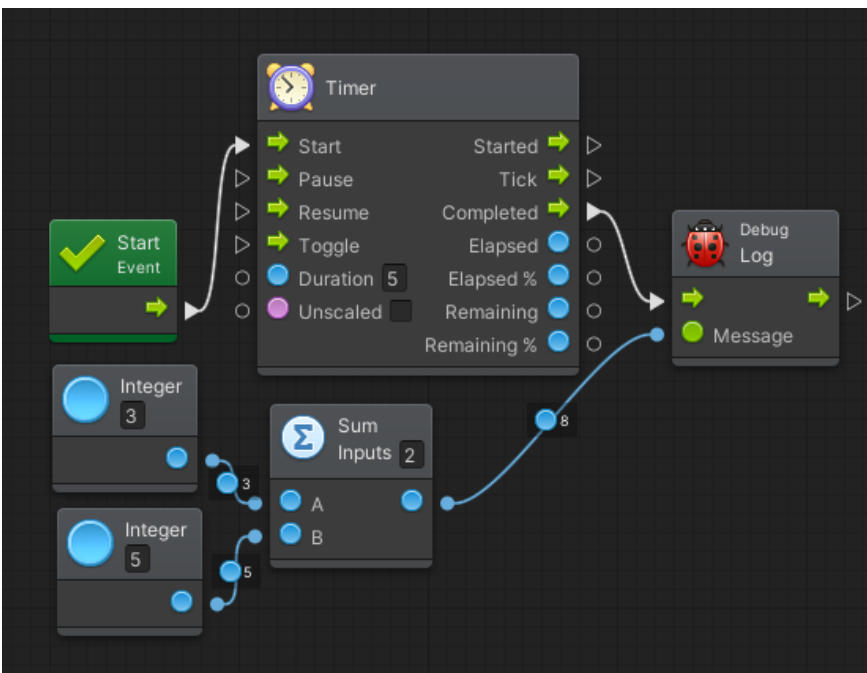
3.2.1 Tilakaavion lukeminen

Tilakaavioissa esiintyvä logiikka on verrattavissa vahvasti esim. perinteiseen lukemiseen tai yleisimpien koodikielten kirjoittamiseen. Siinä missä perinteisessä lukemisessa edetään oikealta vasemmalle ja koodatessa ylhäältä alas, niin Boltin tilakaavioissa mennään tilakaavion nuolten mukaisesti funktiosta toiseen. Kuviossa 1 on esitelty yksinkertainen tilakaavio, jonka tehtävänä on laskea kaksi ennalta annettua numeroa yhteen. Start event -funktio etenee nuolta pitkin ensimmäisen ruudun piirityessä pelaajan näytölle Debug log -funktioon, jonka tehtävä on printata Unityn konsoliin message-kenttään tuleva arvo. Start event -funktion alapuolella on kaksi muuttujaa, jotka antavat sum-funktioon kaksi arvoa. Sum-funktio lähettää arvot viivaa pitkin eteenpäin, jolloin yhteenlaskettu arvo tulostuu konsolille.



Kuvio 1. Yksinkertainen esimerkki Flow machine -tilakoneen toiminnasta

Kuviossa 1 esitetty toiminta tapahtuu siis välittömästi, kun ruudulle piirretään ensimmäinen kuva. Mikäli tilakaaviossa on useampia funktioita, voidaan ne asettaa tapahtumaan samanaikaisesti suorituskäynnin puitteissa, mutta vastaavasti suoritusta voidaan viivästyttää esim. timer -funktiolla. Useamman funktion tilakaaviosta on esimerkki kuviossa 2.



Kuvio 2. Esimerkkikuva timer -funktiosta

Mikäli edellä esitetyissä kuvioissa tapahtuisi virheitä olisi Boltin avulla suhteellisen helppoa myös löytää ne tilakoneesta, sillä konsoliin tulostuu perinteinen virheilmoitus sekä lisäksi samalla tilakaaviossa virheen tuottava funktio muuttaa värin punaiseksi.

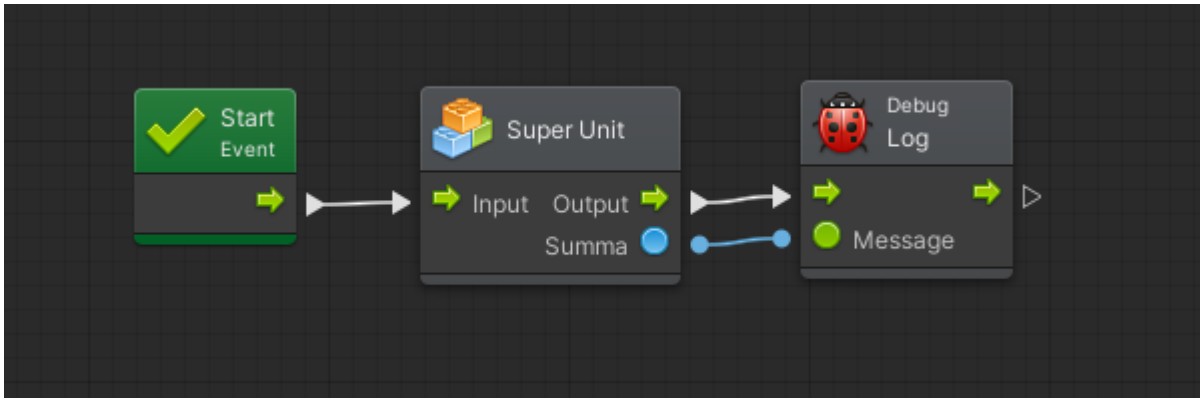
3.2.2 Flow machine -tilakonetyyppi

Flow machine -tilakonetyyppi on tämän opinnäytetyön kehitysprojektissa ehdottomasti käytetyin tilakonetyyppi. Se on muodoltaan yksinkertaisin ja myös havainnollisin verrattuna muihin tilakonetyyppeihin, koska siinä muuttujat ja jokainen funktio määritellään kehittäjän toimesta.

Flow machine on mahdollista tuottaa joko makrona tai pelikomponentin sisäisenä tilakaaviona. Merkittävä ero näiden välillä on Flow machine:n uudelleenkäytettävyys. Makrona tuotettu tilakaavio on mahdollista liittää muihin objekteihin uudelleen omana kokonaisuutenaan. Pelissä toteutetut myöhemmin esiteltävät numerolukot toimivat makroina. Tämä valittiin toteutustavaksi, koska pelissä on useampi lukko, joiden toiminta on riippumaton muista lukoista. Flow machine:n pelikomponentin sisäinen tilakaavio taas toimii muuten identtisesti, mutta ei ole siirrettävissä muihin pelissä oleviin komponentteihin.

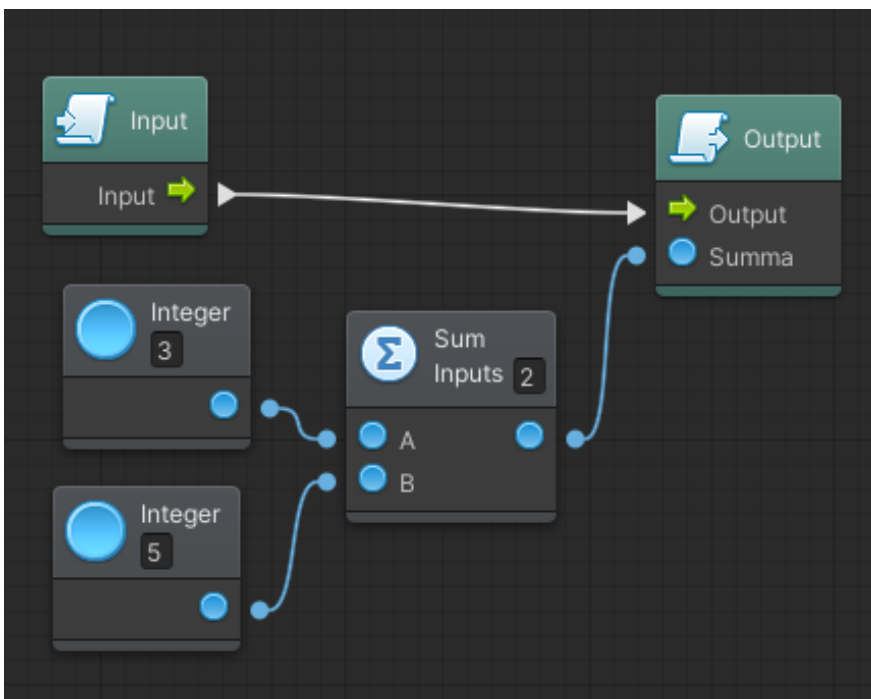
3.2.3 Super unit tilakonetyyppien apuna

Super unit on kokoelma, joka esiintyy yksittäisenä funktiona. Sille on määritetty sisääntuloon vaaditut muuttujat sekä ulostulosta saatavat muuttujat. Käyttämällä Super unit:ia voidaan selkeyttää tilakaavioita merkittävästi. Yhden Super unit:n alle on mahdollista liittää käytännössä loputon määrä funktioita. Tällöin Super unit esiintyy luodussa Flow machine -tilakoneessa ainoastaan yhtenä funktiona. Kuviossa 3 esitetään sama summafunktio kuin kuviossa 1 Super unit:n avulla toteutettuna.



Kuvio 3. Esimerkkiteotus Super unit:n käytöstä

Kuviossa 4 esitetään kuvion 3 Super unit:n sisältö, jonka perusteella se laskee kuvion 3 summa ulostulon.



Kuvio 4. Super unit:n sisältö

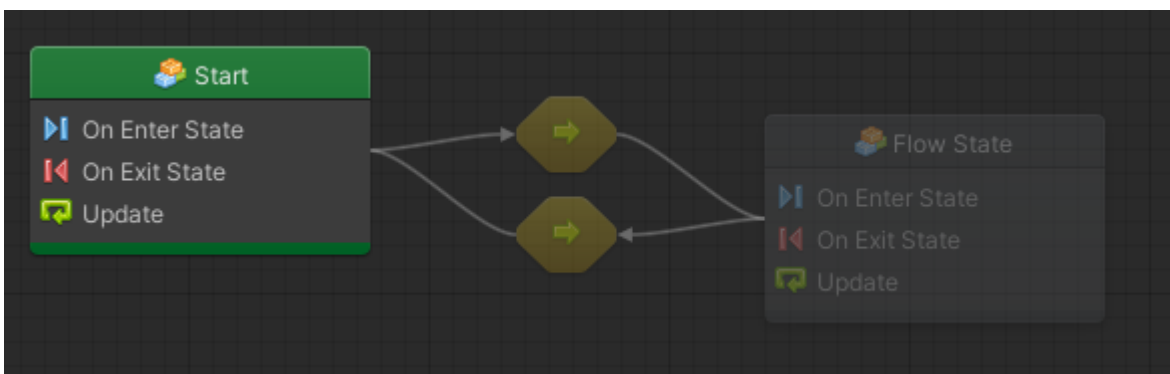
Kuvion 3 ja 4 esitellyssä Super unit:ssä etu on vielä suhteellisen pieni, koska sum-funktio on toimintona hyvin yksinkertainen. Monimutkaisissa tilakoneissa voi olla satoja funktioita, jolloin Super unit:n tuoma etu on merkittävä tilakaavion selkeyden kannalta.

On myös mahdollista jakaa Super unit:a eri projektien ja eri kehittäjien kesken. Super unit on huomattavasti monikäyttöisempi hyödyntää uudessa projektissa sellaisenaan kuin kokonaisena Flow machine:na.

3.2.4 State machine -tilakonetyyppi

State machine -tilakonetyyppi ohjaa ennalta määritettyjen ehtojen avulla tietyn komponentin toimintaa. Käytännössä State machine:n on mahdollista syöttää useampi Flow machine:n kaltainen tilakone, joiden aktiivisuutta vaihdetaan tiettyjen ehtojen muuttuessa. Yleisin esimerkki on hahmon liike. State machine:n avulla voidaan esimerkiksi vaihtaa hahmon tila tuottamaan liikettä sekä laukaisemaan kävelyanimaatio pelaajan painaessa W-näppäintä. Vastaavasti State machine huolehtii, että kun W-näppäintä ei enää paineta, hahmo pysähtyy ja animaatio vaihtuu. State machine:n logiikka ja toiminnallisuus ovat muuten hyvin lähellä aikaisemmin esiteltyjä Flow machine -tilakoneita, mutta pääasiallinen eroavaisuus on automaattinen tilojen vaihto ehtojen täytyessä.

State machineen tulee määrittää aloitustila, josta tiettyjen ehtojen täytyttyä voidaan siirtyä haluttuun toiseen tilaan. Esimerkki hyvin yksinkertaisesta State machine -tilakoneesta on löydettävissä kuviosta 5.



Kuvio 5. Esimerkkikuva State Machinen toiminnasta

4 Kehitystyön tavoite

4.1 Pelin konsepti

Pelin tavoite on antaa pelaajalle avaimet soveltavaan päättelyyn, jotta pelaaja pystyy progressiivisesti etenemään tasojen läpi ja ratkaisemaan erilaisia pulmia yhdistellen useita vihjeitä. Peli toteutetaan kokonaan Unity-pelimoottorilla ja on pelattavissa ainoastaan VR-peleihin soveltuvalla laitteistolla. VR-pelimaailmaa käytetään hyödyksi siten, että pelaaja pystyy paremmin havainnoimaan pulmien mahdollisia vihjeitä sekä poikkeuksellisia tapoja piilottaa vihjeitä pelin sisälle. Myöskin tietyissä tilanteissa pelaajan fyysinen liike on suunniteltu parantamaan pelaajan vireystilaa pulmia ratkoessa. Pelin genre ja ideologia ottavat useita vaikutteita oikean elämän pakohuonepeleistä. Vastaavia lineaarisia pakohuone-elementin toteutuksia on nähty useita oikeassa elämässä sekä videopeleissä. Opinnäytetyön kehitystyönä syntyvässä pelissä on kuitenkin poikkeuksellisesti painopiste virtuaalitodellisuudessa ja erityisesti fysiikan todentamisessa mahdollisimman realistisesti. Hyviä esimerkkejä realistisen fysiikan läsnäolosta pelissä on mm. sellin ovi, jota pelaajan täytyy työntää kaksin käsin oikeassa elämässä, jotta raskas ovi siirtyy. Toisena esimerkkinä toimii hyvin tikan heittäminen tikkatauluun, jotta pelaaja pääsee etenemään pulman ratkomisen parissa.

4.2 Kehitystyön tavoitteet ja onnistumisen mittarit

Tavoitteeksi on kirjattu tuottaa pelattava demo. Pelattavan demon laadun mittaukseen käytetään usealla henkilöllä suoritettavaa pelitestausta. Käyttäjillä suoritettavat pelitestaukset ovat merkittävässä osassa määrittämässä demon käyttäjäkokemuksen onnistumista. Pelitestauksessa käytetään laadullista tutkimusmenetelmää, sillä otanta tulee olemaan vain muutamia loppukäyttäjiä. Loppukäyttäjät vastaavat muutaman kysymyksen kyselyyn pelin jälkeen ja heidän pelaamistaan seurataan jatkuvasti, sekä eri pulmiin menevä aika kellotetaan. Kaikesta aineistosta luodaan tiivistelmä, joka tulee määrittämään millä tasolla peli on onnistunut vastaamaan sille asetettuja tavoitteita.

VR-sovellusta suunnitellessa ja kehittäessä tulee käyttäjäkokemuksen huomioiminen ottaa suurella vakavuudella. Hyvälläkin idealla varustettu sovellus muuttuu helposti epämieluisaksi kokeemukseksi, jos toteutus on kömpelö, pahoinvointia aiheuttava tai siinä on muuten liikaa visuaalisia

ärsykeitä. Kulmakiviä VR -sovelluskehityksessä ovat esimerkiksi kirkkauden ja valojen huomioiminen, vuorovaikutteisten nappien ja valikoiden sijoitus sekä pahoinvoinnin tunteen ehkäiseminen. Esimerkiksi kirkkauden äkilliset muutokset saattavat osalla pelaajista aiheuttaa päänsärkyä tai silmien arkuutta. Huonosti suunnitellut valikot ja pelin sisäiset nappulat aiheuttavat helposti turhautumisen tunnetta tai muita negatiivisia tunnetiloja, jos ne on suunniteltu VR-maailmaan sopimattomiksi. Nopeat ja suuret liikkeet, joissa pelaaja ei itse liiku saattavat aiheuttaa yleisesti pahoinvointia osalla pelaajista. Ilmiönä tämä on hyvin samankaltainen kuin esimerkiksi linja-autossa matkustamisessa, jolloin matkustajan aistien ristiriitaisuus liikkeen tunteesta aiheuttaa joillakin ihmisistä pahoinvointia. Englanniksi ilmiö on tunnettu ”motion sickness” termillä.

5 Peliympäristön suunnittelu

5.1 Pelin teema ja aloitusvalikko

Pelin ympäristöksi valittiin tarinan ja pulmien perusteella poliisiasema. Pelin aloitusvalikkona toimii kuulusteluhuone, jonka visuaalista ilmettä on esitelty kuviossa 6. Itse pelin alkaessa käyttäjä siirtyy sellitiloihin, josta pelaaja lineaarisesti etenee poliisilaitoksen sisällä. Peliympäristön suunnittelussa tulee ottaa huomioon hyvä valaistus koko pelin ajan, sillä pelaajan tulee erityisesti pulmia ratkaistaessa pystyä näkemään jatkuvasti mahdolliset vihjeet. Valaistuksen poistaminen tai himmentäminen ei palvele pelin alkuperäistä tarkoitusta pulmapelinä. Peliympäristöä rakennettaessa teemana pidettiin realistista grafiikkaa ja yhdenmukaista tunnelmaa kolkosta ja metallisesta poliisilaitoksesta.



Kuvio 6. Pelin aloitusvalikkona toimiva kuulusteluhuone

5.2 Alakerta

5.2.1 Selliosasto

Pelin aloituspisteenä toimiva selliosasto on rakennettu tuntumaan mahdollisimman realistiselta. Siellä on esimerkiksi huomattavan vähän irtoesineitä, joita pelaaja voi koskettaa tai leikkiä niillä. Enemmistö selliosaston irtaimistosta on seinään tai lattiaan pultattuja staattisia esineitä, joiden avulla selliosasto saadaan tuntumaan kolkolta ja ikävältä paikalta pelaajan näkökulmasta. Selliosaston kahden vangin selli on esitelty kuviossa 7.



Kuvio 7. Selliosastolla kahden vangin selli

Selliosastolle on myös lisätty vilkkuvia punaisia hälytysvaloja. Niiden tarkoitus on luoda pelaajalle stressiä ja lisätä immersion tuntua sekä toimia tarvittavana vihjeenä pelin pulman ratkaisussa. Selliosasto noudattaa hyvin pitkälle samaa tyyliä kuulusteluhuoneen kanssa. Tilasta on luotu mahdollisimman kolkko, metallinen ja laitosmainen. Selliosasto on suunniteltu edustamaan tätä ääripäätä, jotta pelin tunnelma ensin säilyy ja muuttuu vasta, kun pelaaja pääsee etenemään muille alueille. Kuviossa 8 esitellään selliosaston käytävä ja punaiset hälytysvalot.



Kuvio 8. Selliosaston käytävä

5.2.2 Vankien käsittelytila

Selliosaston jälkeinen vankien käsittelytila alkaa tyyliään muuttua huomattavasti miellyttävämmäksi verrattuna selliosastoon. Käsittelytilassa pelaajan on ratkaistava kaksi pulmaa, jotta hän pääsee jatkamaan seuraavaan tilaan. Seinillä olevat taulut ja muut koristeet ovat vapaasti pelaajan käytössä, jotta hänen vaikutelmansa eri tasoista voimistuu merkittävästi. Käsittelytilan tavaroitten lisääntyne määrä mahdollistaa myös erilaisten vihjeiden piilottamisen niiden sekaan. Tavoitteena on säilyttää edelleen laitosmainen tunnelma, mutta tasosuunnittelun kautta luodaan vaikutelma pääsystä alueelle, johon pelaajalla ei pitäisi olla pääsyä vangin roolissa. Kuvio 9 kuvaa käsittelytilan aulaa.



Kuvio 9. Käsittelytilan aula

5.2.3 Portaikko

Portaikin päätehtävä on toimia rajana kolkkojen ja metallisten vankien käsittelyyn tarkoitettujen alueiden ja poliisilaitoksen muun henkilöstön tilojen välillä. Portaikin alaosa on edelleen metallinen ja kolkko, mutta portaiden yläosasta alkaa selkeämmin täysin eri tyyliuunnan huoneisto. Kuvio 10 esittelee portaikin alaosa, mistä on nähtävissä karu tyyliuunta.



Kuvio 10. Portaikon alapuoli

Portaikossa pelaajan on ratkottava vain yksi käytännön ongelma, jonka jälkeen hän pääsee jatkaamaan portaikon yläosassa olevasta ovesta seuraavalle alueelle. Kuviossa 11 on jo hieman havaittavissa värimaailman muutosta pelin edetessä kohti henkilökunnan tiloja.



Kuvio 11. Portaikon yläpuoli

5.3 Yläkerta

5.3.1 Taukotila

Taukotila on yläkerrassa sijaitseva tila, joka aukeaa pelaajalle noin puolivaiheessa peliä. Tästä huolimatta taukotilassa sijaitsee koko pelin viimeinen pulma, joka pelaajan on palattava ratkomaan kerättyään tarvittavat vihjeet muista huoneista. Se sisältää myös vihjeitä, joilla pelaaja pääsee etenemään muihin huoneisiin yläkerran alueella. Taukuhuone on poliisiaseman henkilöstölle suunniteltu taukotila, joten edellisten tilojen kolkkaus ja metallinen tyyliä ei enää yläkerran peliympäristön suunnitteluun. Sen sijaan tyyliä on otettu enemmän käyttöön pehmeät materiaalit ja neutraali värimaailma. Kuvio 12 esittelee taukotilan miellyttävää peliympäristöä.



Kuvio 12. Taukuhuoneen nurkkatila

5.3.2 Toimisto

Toimistotilan tasosuunnittelussa on pyritty luomaan mielikuva kiireisestä ja sekaisesta toimistosta. Tästä syystä huone sisältää ylivoimaisesti eniten yksittäisiä esineitä, joista kaikille on luotu oma fyysikkansa, jotta ne käyttäytyvät VR-pelaajalle aidosti sekä jokaista esinettä voi liikuttaa ja koskettaa.

Toimisto pitää sisällään kaksi erilaista ratkottavaa pulmaa. Toimiston laatikot on tehty avattaviksi, joten toimiston tarkkaan tutkimiseen kuluu pelaajalta aikaa. Yläkerran toimistotila on esitelty kuviossa 13.



Kuvio 13. Yläkerrassa sijaitseva toimisto

5.3.3 Suunnitteluhuone

Suunnitteluhuone on yläkerran perimmäinen huone. Se on tarkoitettu poliisilaitoksen henkilöstön kokouksille, joten tyyliisuunta tavoittelee perinteistä neuvottelutilaa. Tilasta löytyy paljon mm. piir-
totauluja, kelloja sekä televisio, joten visuaalisia ärsykykeitä on pelaajalle paljon. Suunnitteluhuo-
neessa sijaitsee kolme eri pulmaa pelaajan ratkottavaksi. Kuvio 14 esittelee suunnitteluhuoneen
pelisuunnittelun.



Kuvio 14. Yläkerran suunnitteluhuone

6 Pelillistävät pulmat

6.1 Pulmien pelillistävä tarkoitus

Pulmien pelillistävässä suunnittelussa on otettu merkittävästi vaikutteita oikean elämän pakohuonepeleistä. Peli sisältää noin kymmenen erilaista pulmaa, jotka käyttäjän tulee ratkaista edetäkseen pelissä. Pelin tyyliisuunta on vahvasti pakohuonepelien hengessä luotu. Varmistaakseen pelillisyyden luoman mielenkiinnon laajalle kohdeyleisölle ovat pulmat merkittävässä roolissa.

6.2 Pulmien esimerkkitoteutuksia

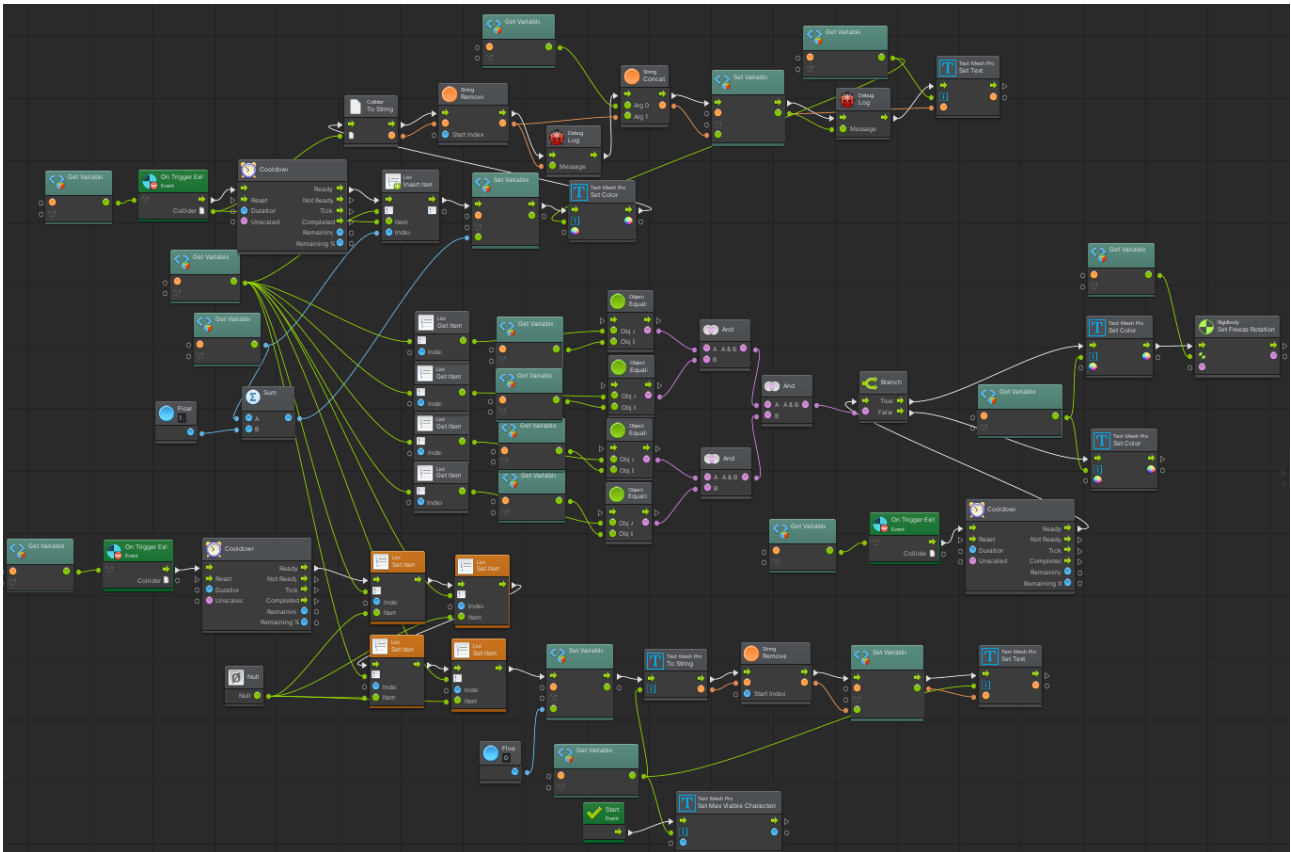
6.2.1 Numerolukko

Pelissä toistuu useassa pulmassa numerolukko, johon pelaaja voi syöttää neljä numeroisen koodin. Numerolukko on tehty Unityn perusmateriaaleilla itse, joten valmista 3D-mallia ei ole käytetty. Lukon tarkempi visuaalinen toteutus selviää kuviosta 15. Jokaisen nappulan taakse on tehty oma näkymätön nivel, jonka tehtävänä on vastustaa kevyitä voimia ja taipua ainoastaan käyttäjän painaluksesta kohti lukon takaseinää. Kaikki suunnat, lukuun ottamatta lukon takaseinää kohti, on rajoitettu pois pelimoottorin toimesta. Numerolukon logiikka on rakennettu kokonaan Boltin avulla.



Kuvio 15. Esimerkkikuva yläkerrassa sijaitsevasta numerolukosta

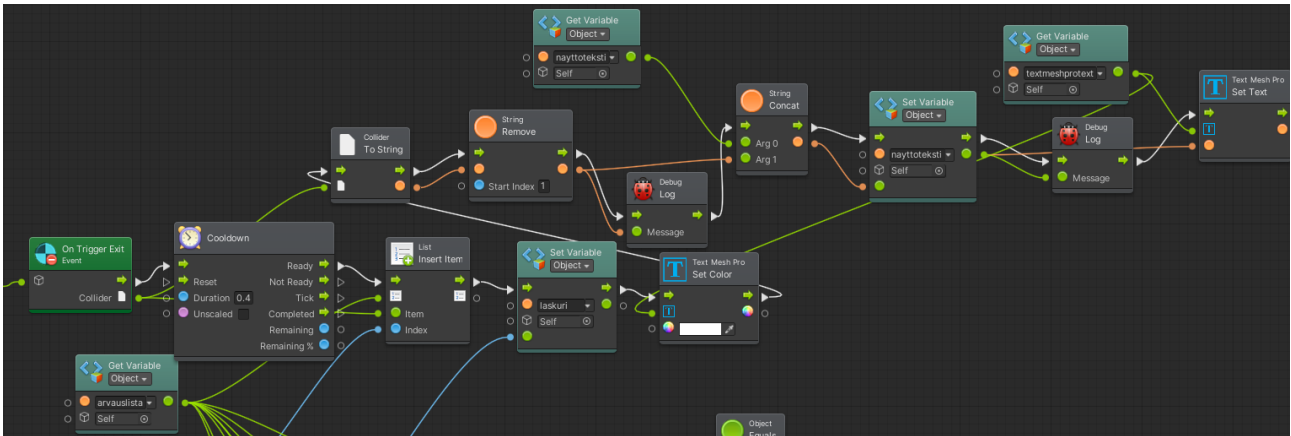
Numerolukon toiminnallisuudelle asetettiin useita vaatimuksia. Sen haluttiin tulostavan ruudulle pelaajan syöttämät numerot sekä vaihtavan väriä, jos pelaajan koodi on oikein. Numerolukkoon piti saada myös mahdollisuus havaita käyttäjän painallukset ja ehkäistä virhepainalluksia sekä tyhjentää pelaajan lukkoon syöttämät numerot. Nämä ominaisuudet saatiin luotua kuviossa 16 esitellyyn Boltin tilakoneeseen.



Kuvio 16. Yleiskuva Boltin logiikasta numerolukon takana, mistä voidaan havaita tilakoneen laajuus

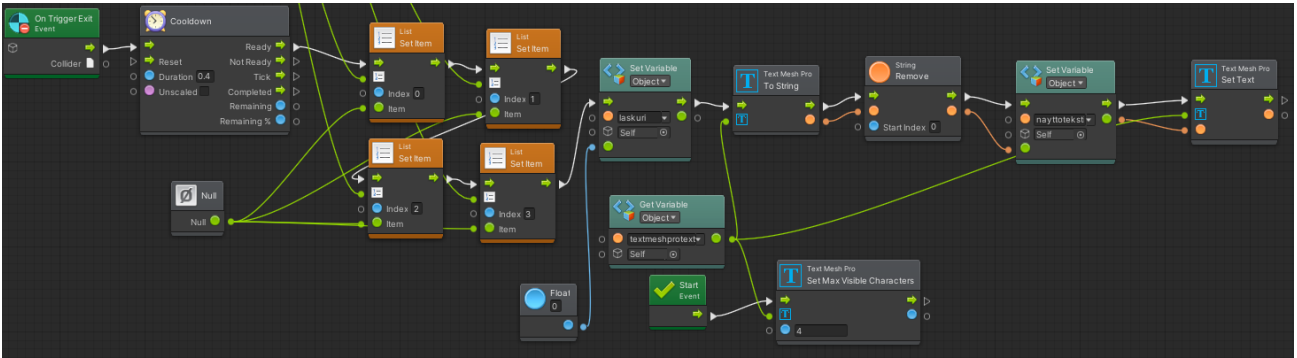
Koska tavoitteena oli tunnistaa käyttäjän painallus ja pyrkiä estämään mahdollisia virhepainalluksia, huomioitiin se tilakoneen suunnittelussa jo alkuvaiheessa. Numerolukon takana sijaitsee lukon takaseinä, joka toimii anturina. Kaikki takaseinästä irtoavat komponentit aiheuttavat hälytyksen anturissa ja pelimoottori kertoo, mikä esine irtosi takaseinästä. Käytännössä pelaajan painalluksella ja napin irrottamisella saadaan aikaan hälytys anturissa, joka kertoo mitä nappulaa käyttäjä painoi kohti lukon takaseinää. Tähän toimintoon lisättiin 0.4 sekunnin palautumisaika, jotta mikäli pelaaja osuisi useaan nappiin samanaikaisesti, ainoastaan ensimmäinen painallus rekisteröidään

lukkoon. Jokainen onnistunut painallus kirjataan listaan indeksin osoittamalle paikalle. Ensimmäinen painallus kasvattaa indeksia, jotta toinen painallus sijoittuu listassa seuraavalle indeksille. Samasta anturista otetaan myös painetun napin nimi, josta saadaan numero printtavaksi lukon näytölle. Numero voi olla ainoastaan yhden merkin pituinen, joten printattava merkkijono lyhennetään automaattisesti yhden merkin pituiseksi String Remove -toiminnon avulla. Painalluksen tunnistaminen, virhepainallusten estäminen sekä numeron tulostamiseen suunniteltu tilakone on esitelty tarkemmin kuviossa 17.



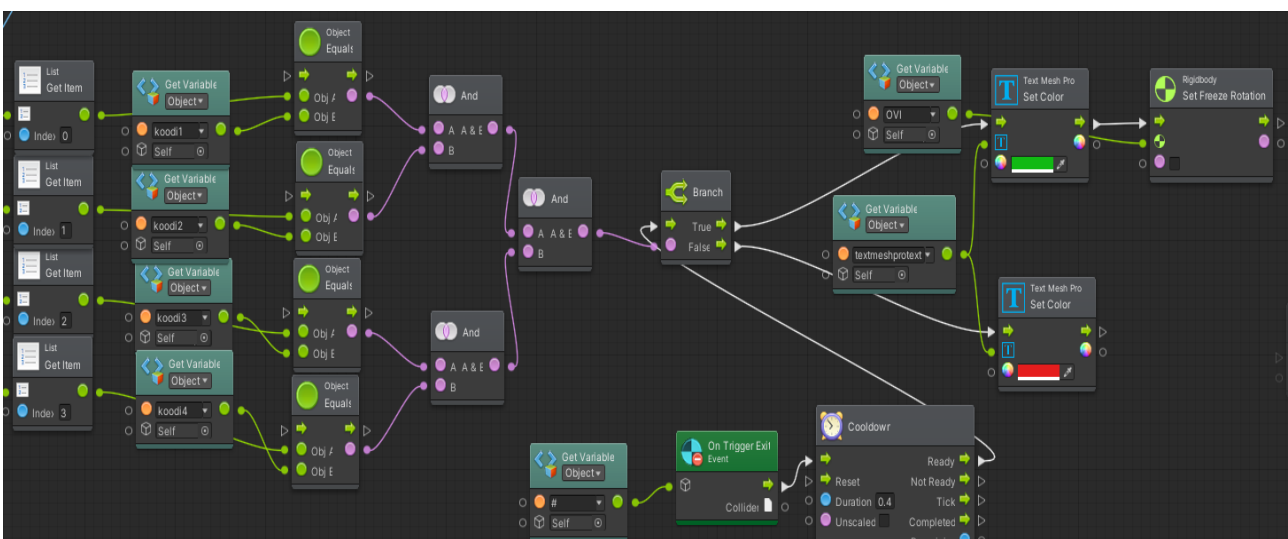
Kuvio 17. Logiikka nappien painalluksessa

Numerolukossa on mahdollisuus tyhjentää jo syötetty koodi, mikäli koodi on virheellinen tai pelaaja painaa väärää nappulaa. Tämä tapahtuu painamalla näppäintä, jossa on * kuva. Tässäkin painalluksessa on käytössä sama 0.4 sekunnin palautumisaika virhepainallusten välttämiseksi. Ainoastaan neljällä ensimmäisellä indeksillä on merkitystä siinä listassa, johon tallennetaan käyttäjän syöttämät numerot, sillä jokainen koodi pelissä on maksimissaan neljän numeron pituinen. Pelaajan painettua lukon resetointinäppäintä asetetaan listan indeksit 0, 1, 2 ja 3 tyhjiksi arvoiksi. Laskuri, jonka tehtävänä on asettaa seuraava painallus taas oikean indeksin kohdalle listassa, asetetaan arvoon 0. Lopuksi poistetaan näytöllä olevat vanhat pelaajan painamat numerot. Listan tyhjentäminen ja laskurin asettaminen on esitelty kuviossa 18.



Kuvio 18. Logiikka lukun resetoinnissa

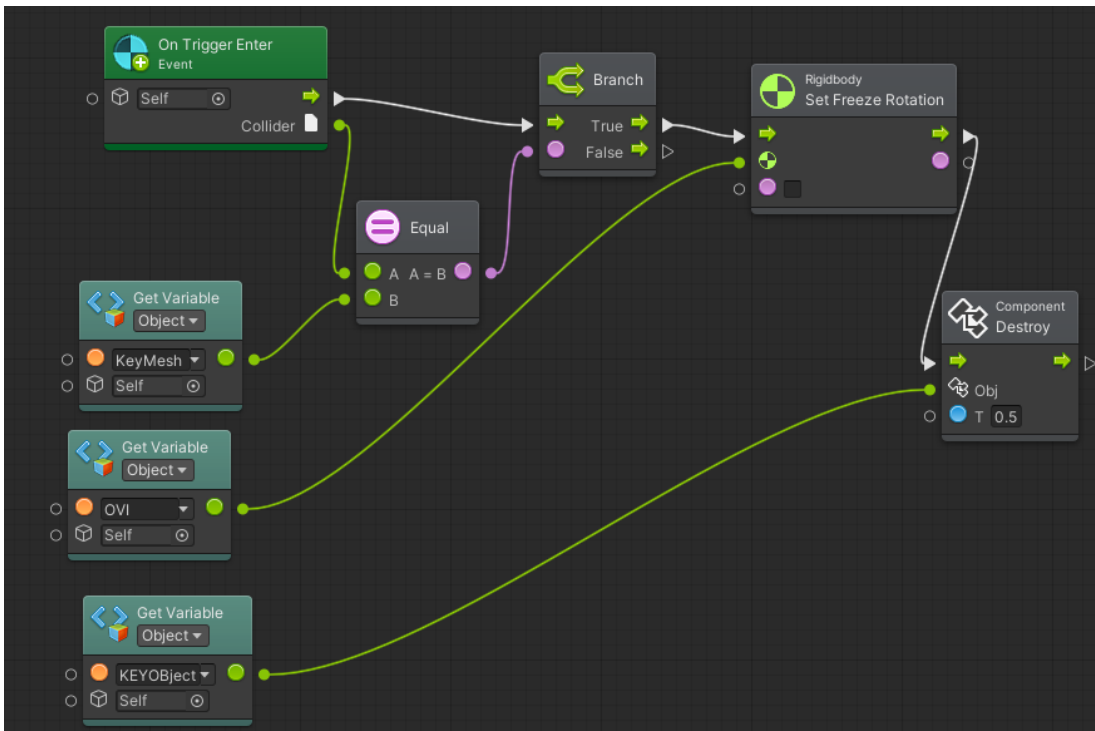
Numerolukun tulee myös tarkistaa pelaajan syöttämä koodi ja verrata sitä oikeaan koodiin. Jos koodi täsmää, halutun oven tulee avautua. Vertailua varten listasta otetaan indeksiltä 0, 1, 2 ja 3 arvot ja verrataan niitä muuttujiin koodi1, koodi2, koodi3 ja koodi4. Kaikkien indeksien täsmätessä kaikkiin muuttujiin voidaan todeta koodin olevan oikein. Tämän ansiosta lukkoa ei ole mahdollista murtaa, vaikka pelaaja painaisi useampaa kuin neljää numeroa. Pelaaja voi aktivoida edellä mainitun tarkistusprosessin painamalla lukosta # -näppäimen kuvaa. Pelaajan painaessa tarkistusnäppäintä ja koodin ollessa oikein teksti muuttuu vihreäksi ja poistaa määritellyn oven rajoitetun sijainnin. Kun taas pelaajan painaessa tarkistusnäppäintä ja koodin ollessa väärä, teksti muuttuu punaiseksi ja ovi pysyy suljettuna. Kuviossa 19 esitellään tilakone, jossa verrataan listan indeksejä oikeisiin muuttujiin sekä koodin ollessa oikein avataan ovi.



Kuvio 19. Logiikka koodin tarkastamisessa

6.2.2 Ovien avaaminen avaimella

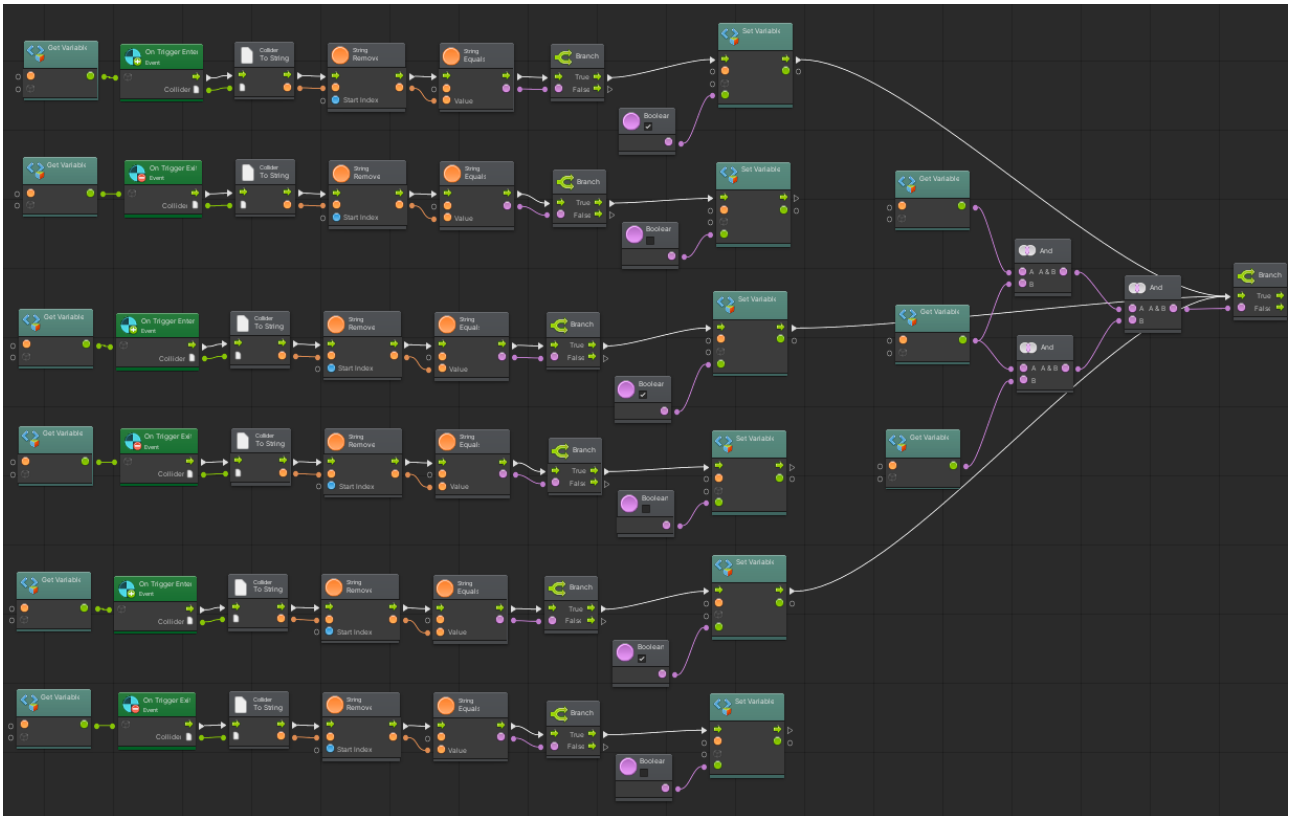
Kaikkiin oviin on määritelty lukkopesän kohdalle alue, joka toimii anturina. Pelimoottori ilmoittaa sille alueelle osuvat esineet. Ensin halutaan määrittellä, ettei mikä tahansa esine avaa ovea vaan uniikki avain on ainoa esine, jonka seurauksena ovi aukeaa. Esinettä verrataan siis ennalta määritettyyn KeyMesh-muuttujaan. Jos voidaan todeta, että pelaaja laitto avaimen oven lukkopesään, ovi aukeaa ja avain tuhotaan pois pelimaailmasta, jottei pelaaja sekoita identtisen näköisiä avaimia keskenään. Avaimen toiminta oven lukkopesässä on esitelty kuviossa 20.



Kuvio 20. Logiikka oven avaamiseen avaimella

6.2.3 Tikkataulu

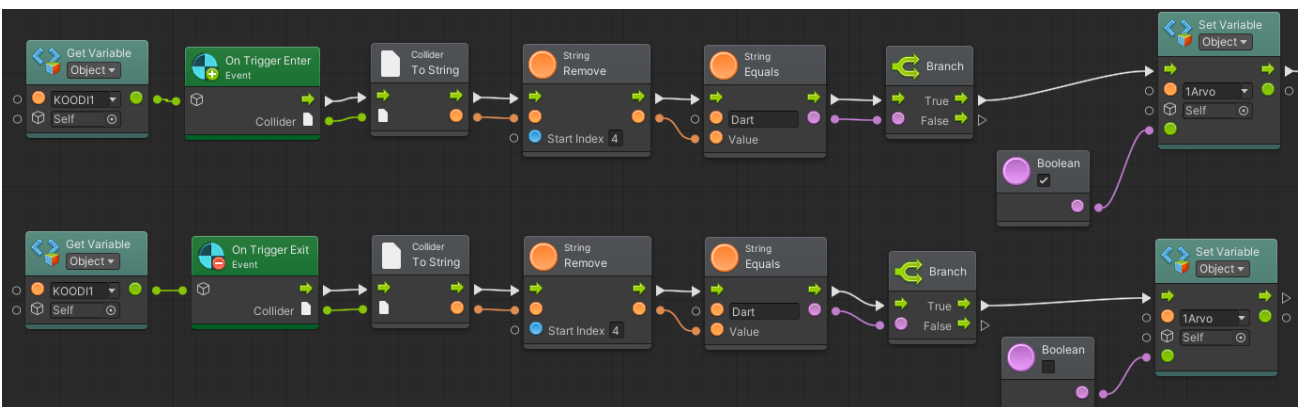
Pelissä yhtenä ratkaistavana pulmana on tikkataulu, jossa käyttäjän tulee laittaa tikat oikeisiin kohtiin. Tälle pulmalle vaadittu määritelmä on se, että kaikkien tikkojen tulee olla yhtä aikaa taulussa



Kuvio 21. Tikkataulun logiikkatilakone kolmelle tikalle

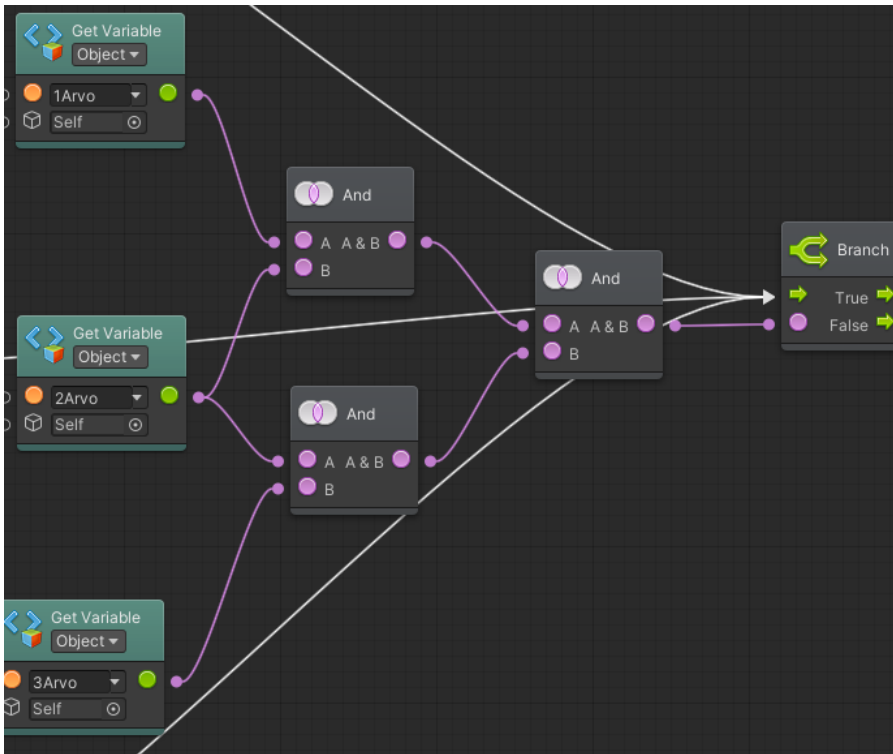
pulman ratkeamiseksi. Tikkojen järjestyksellä ei ole väliä, mutta tikkojen sijaintien on oltava tismalleen oikein. Nämä ominaisuudet saatiin luotua kuviossa 21 esitelyyn Boltin tilakoneeseen.

Tikkataulussa verrataan sitä, onko pelaaja laittanut oikeaan kohtaan esineen ja onko kyseinen esine tikka. Mikäli se on tikka, asetetaan tämä kohta tikkataulusta arvolle Tosi. Jos tikka syystä tai toisesta irtoaa tikkataulusta, asetetaan sama kohta tikkataulusta arvolle Epätosi. Yhden tikkataulun kohdan tilakone on esitelty kuviossa 22.



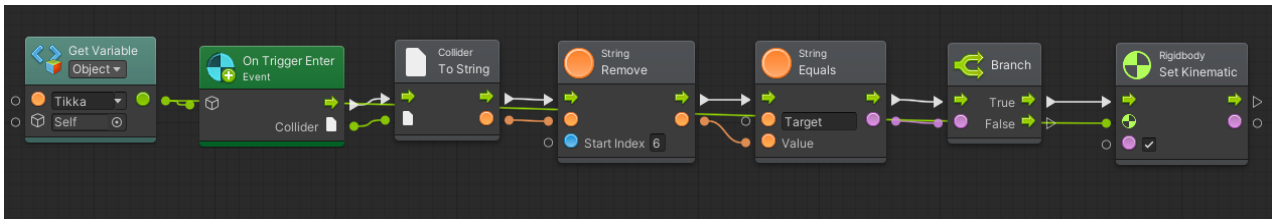
Kuvio 22. Tilakoneen logiikka tikan asettamisessa

Tämä logiikka toistetaan identtisenä kolmelle eri kohdalle, sillä pulman suorittaminen vaatii kolmen tikan olevan oikeassa kohdassa samanaikaisesti. Lopuksi kuviossa 23 verrataan kohtien arvoja keskenään. Mikäli kaikki 3 arvoa ovat Tosi samanaikaisesti, pulma merkitään ratkenneeksi.



Kuvio 23. Logiikka tikkataulun kohtien arvojen vertaamisessa

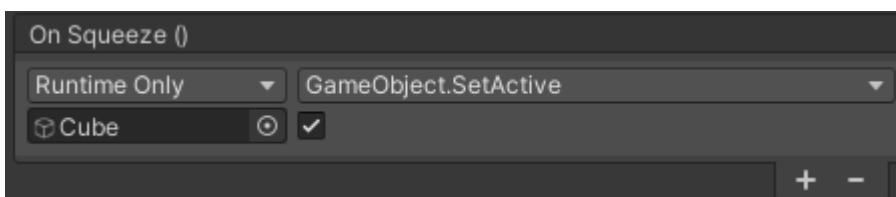
Jokaisessa tikassa on määritelty oma tilakoneensa. Sen ansiosta ne pysyvät kiinni tikkataulussa. Tikkan teräosaan on asetettu anturi, joka tarkistaa jatkuvasti osuuko teräosa toiseen esineeseen. Osuessaan tikkatauluun terä edellä jähmettyy tikka kinemaattiseen tilaan, jolloin saadaan luotua efekti tikan jäämisestä kiinni tikkatauluun. Pelaajan tarttuessa valmiiksi kinemaattisessa tilassa olevaan tikkaan muuttuu sen tila automaattisesti takaisin ei-kinemaattiseksi, jolloin käyttäjä voi vetää tikan irti taulusta. Jokaisessa tikan teräosassa olevaa tilakonetta on esitelty kuviossa 24.



Kuvio 24. Logiikka tikan pysymiseen tikkataulussa

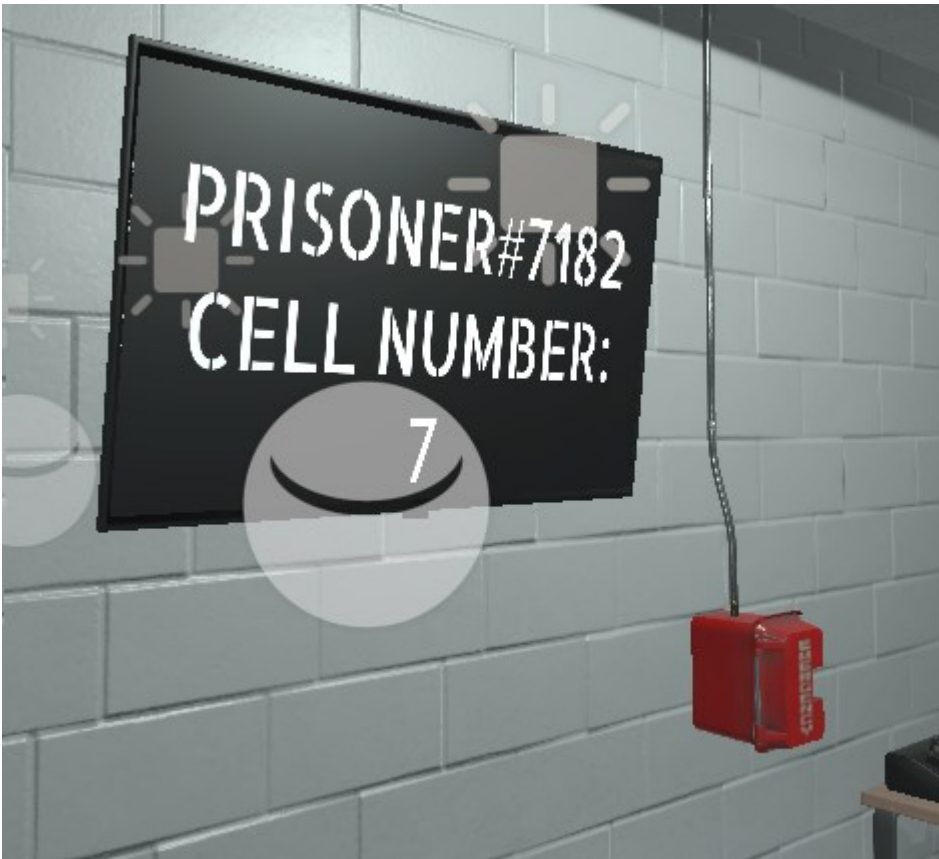
6.2.4 Videon avaaminen televisioon kaukosäätimellä

Pelaajan löydettyä kaukosäätimen voi hän puristaa ohjainta ja aktivoida siten On Squeeze -funktion. Tälle funktiolle on määritelty se, että pelaajan puristaessa esinettä voidaan aktivoida haluttu tapahtuma. Kaukosäädin on määritelty aktivoimaan puristettaessa toinen peliesine. Se on näkymätön kuutio, joka on sijoitettu televisioruudun eteen ja aloittaa videon toistamisen välittömästi aktiivoinnin yhteydessä. Kuviossa 25 on esitelty kaukosäätimen määrittely Unity-pelimoottorin sisällä.



Kuvio 25. Kaukosäätimen On Squeeze määrittely

Kuviossa 26 näkyy visuaalinen näkymä pelaajalle television näkymättömän kuution aktivoitua.



Kuvio 26. Kuva videosta televisiossa aktivoinnin jälkeen

7 Sovellustestausten analysointi

7.1 Sovellustestauksen käytännön toteutus

Pelitestauksia järjestettiin neljä kappaletta. Kaikki ne järjestettiin etänä verkon yli siten, että testaaaja pelaa pelin läpi opinnäytetyön tekijän valvoessa ja dokumentoidessa suorituksen. Sovellustestaukseen valittiin loppukäyttäjää, jotka omistavat valmiiksi tarvittavan välineistön sekä joilla on mielenkiintoa pakohuoneteemaan. Testauksen aikana mitattiin mm. jokaisen pulman ratkaisemiseen kulunut aika ja otettujen vihjeiden määrä. Välittömästi pelin suorittamisen jälkeen testattavalta kysyttiin viisi kysymystä käyttäen laadullista tutkimusmenetelmää. Kyselyn kysymykset pyrkivät selvittämään käyttäjäkokemuksen onnistumista. Niiden avulla kartoitettiin mm. mahdollinen pahoinvointi VR-kokemuksen johdosta sekä pulmien sopiva vaikeustaso ja määrä.

7.2 Yhteenveto sovellustestauksen tuloksista

Tarkemmat testitulokset sekä tietojenkeruulomake on esitetty opinnäytetyön liitteissä 1–5. Niistä voidaan tutustua tarkemmin yksittäisen pelaajan vastauksiin. Testituloksien keskeisimmät havainnot on koottu alla oleviin taulukoihin 1 ja 2.

	Kysymys 1: Koitko pahoinvointia pelin aikana tai välittömästi pelin jälkeen?	Kysymys 2: Mikä oli vaikein pulma?	Kysymys 3: Mikä oli helpoin pulma?	Kysymys 4: Oliko pulmia sopiva määrä?	Kysymys 5: Häiritsikö yleisen teleportaatio- ominaisuuden puute?
Pelaaja 1	Ei	Pulma 9	Pulma 5	Kyllä	Ei
Pelaaja 2	Lievästi	Pulma 9	Pulma 5	Kyllä	Ei
Pelaaja 3	Ei	Pulma 9	Pulma 5	Kyllä	Ei
Pelaaja 4	Ei	Pulma 9	Pulma 1	Kyllä	Ei

Taulukko 1. Pelaajien vastaukset analysoituna yhteenvetotaulukossa

Yhdelläkään neljästä testattavasta ei esiintynyt mittavassa määrin pahoinvointia pelin aikana tai välittömästi sen jälkeen. Ainoastaan yksi testattava kertoi havainnon pienestä pahoinvoinnista pelin alussa, jonka selitti vähäinen VR-laitteiden käyttökokemus. Havainnon mukaan pelaajan totuttua liikkeeseen VR-sovelluksessa pahoinvointi lieveni tai poistui kokonaan.

Pulmien määrä oli yleisesti pelaajien mielestä sopiva. Pelaajien tekemät huomiot vahvistivat sopivaa pulmamäärää, varsinkin koska pelistä puuttuu tallennusominaisuus ja moninpelimahdollisuus. Täten pelattavaa on sopiva määrä yhdelle pelaajalle yhdellä pelikerralla. Keskiarvo pelaajilla pelin suorittamisessa oli yksi tunti 15 minuuttia ja 30 sekuntia. Yleisesti pakohuoneiden pelaaminen on rajoitettu noin tunnin pituisiksi pelikerroiksi, joten tämä keskiarvo tukee testattavien mielipidettä sopivasta vaikeustasosta ja pulmien määrästä.

Selkeästi eniten hankaluuksia aiheutti pulma numero 9, jonka jokainen testattava nimesi vaikeimmaksi. Suunnitteluvaiheessa ajatuksena oli rakentaa pelin loppuvaiheeseen yksi selkeästi haastavampi pulma, joten pelaajien havaintojen mukaan tämä onnistui. Pelikokemuksen kannalta kyseinen pulma sai myös kiitosta sen haasteellisuuden ja täten mielenkiintoa lisäävän vaikutuksen vuoksi. Koska kaikki testaajat saivat pulman lopulta ratkaistua, tuottaa se onnistumisen tunteen pelikokemuksen aikana. Helpoimmaksi pulmaksi suurin osa nimesi pulman numero 5, mutta myös pulma numero 1 luokiteltiin helpoksi.

Taulukossa numero 2 on esitetty opinnäytetyöntekijän keräämään statistiikkaan liittyvät havainnot.

	Pelin kokonaiskesto: h.min.sek	Otettujen vinkkien määrä:	Ajallisesti pisin pulma:
Pelaaja 1	1.19.50	9	Pulma 2
Pelaaja 2	1.40.45	4	Pulma 2
Pelaaja 3	0.55.46	0	Pulma 9
Pelaaja 4	1.05.38	0	Pulma 9
Keskiarvo:	1.15.30	3,25	-

Taulukko 2. Pelin kulkuun liittyvä statistiikka analysoituna yhteenvetotaulukossa.

Käyttäjät ottivat keskiarvona 3,25 vihjettä pelikertaa kohden. Otannasta löytyi pelaajia, jotka läpäisivät pelin ottamatta ollenkaan vihjettä, mutta vastaavasti jotkut pelaajat tarvitsivat useampia vihjeitä. Ajallisesti pisimpään pelaajilla kesti ratkaista pulmat 2 ja 9. Kaikki pelaajat kuitenkin nimesivät pulman 9 haastavimmaksi. Pulma 2 on heti pelikokemuksen alussa ja siinä vaiheessa pelaaja vasta totuttelee pakohuone -ajatusmaailmaan, joten on luonnollista, että alussa ratkaistavat pulmat vievät aikaa. Toisena selittävänä syynä voidaan pitää totuttelua VR-elementteihin ja laitteiden käyttöön. Pulma 9 vaatii haastavaa ajattelua ja oivallusta vaikeutensa vuoksi, joten pelaajat käyttivät luonnollisesti sen parissa paljon aikaa. Peliajat vaihtelivat suhteellisen merkittävässä määrin eri testikerroilla.

Yleisesti käyttäjäkokemus vaikutti hyvin onnistuneelta. Monet testaajat nauttivat ylimääräisistä esineistä, jotka loivat sopivaa haastetta ja aitoa sekavuuden tuntua. He antoivat vapaa kommenttikenttä -vastauksissa hyviä ideoita jatkokehitykselle. Pulmien vaikeustaso nousi monen pelaajan mielestä loppua kohti. He pitivät tästä suunnittelusta.

8 Opinnäytetyön tulokset ja onnistumisen arviointi

Kehitystyönä tuotettiin toimiva ja pelattava OpenXR-pohjainen pakohuonepeli-sovellus Unity-pelimoottorilla. Sovellusta on testattu useassa eri järjestelmässä kehitysympäristön ulkopuolella, joten sen toimivuus vaikuttaa varmalta. Kehitystyön aikana luotiin useita tilakaavioita, jotka on esitelty osana opinnäytetyötä. Näiden kaavioiden tehtävänä on ohjata sovelluksessa toimivaa logiikkaa.

Kehitystyöhön käytetystä ajasta ei ole tarkkaa mittausta, mutta työtuntien määrä on todellisuudessa noin 200–300 tunnin välissä. Kehitystyölle asetetut vaatimukset täytyivät. Opinnäytetyön alussa määritellyt tavoitteet: sopiva vaikeustaso ja mielenkiintoinen pelikokemus myös aikuiselle ihmiselle, vaikuttaisivat havaintojen perusteella täytyneen hyvin.

Kehitystyöstä saatu pelaajien yleinen palaute oli positiivista. Sovellus on teknisesti toimiva. Yhdenkään testauksen aikana ei tapahtunut pelin pelaamiseen vaikuttavia virheitä, vaan ainoat havainnot olivat lähinnä vähäisiä visuaalisia virheitä. Havaitut viat ja muut kehityskohteet olivat kokonaisuutena nähden pieniä ja helposti korjattavissa. Käyttäjäkokemus miellettiin monilla aspekteilla mukavaksi. Saatujen havaintojen perusteella voidaan tehdä johtopäätös, että kehitystyö on onnistunut sille asetetuissa tavoitteissa.

Kehitystyön aikana opittiin merkittävästi Unityn toiminnasta ja erityisesti sovelluskehittämisestä VR-laitteille. Ensimmäisenä tutkimustavoitteena oli analysoida valittujen Unityn komponenttien soveltuvuutta pakohuonepelin suunnitteluun ja toteutukseen. Tältä osin opinnäytetyöntekijän havainnot ovat seuraavia.

OpenXR-standardi oli lähes huomaamaton koko kehitystyön demopelin toteutusprosessin ajan. Standardin alkuasetusten jälkeen niihin ei ollut tarvetta suunnittelun edetessä palata, vaan OpenXR toimi virheettömästi ja huomaamattomasti koko projektin pohjana ja antoi täten aikaa keskittyä muihin kehitysongelmiin. Tästä johtuen OpenXR-standardi toimi projektin tarkoitukseen erinomaisesti ja täytti asetetut yleiset vaatimukset laitteiden seurannan ja monimuotoisuuden puolelta. Voidaankin sanoa, että OpenXR on erittäin soveltuva yksinkertaisiin VR-sovelluksiin, joita on tarkoitus käyttää usealla eri laitevalmistajan laitteella.

Unity oli hyvä valinta tähän projektiin ja toimi pelimoottorina erinomaisesti. Prosessin aikana opinnäytetyöntekijällä nousi esiin kysymys pelin suorituskyvystä pienitehoisilla päätelaitteilla, koska pelissä olevien yksittäisten fysiikkaan reagoivien esineiden määrä nousi radikaalisti. Unity selviytyi kuitenkin optimoinnista ja pelin rakentamisesta hyvin ja sovellus toimi myös useilla pienitehoisilla päätelaitteilla opinnäytetyöntekijän demostudiossa. Unityn käyttöliittymä koettiin selkeäksi ja käytännölliseksi mukautumisominaisuuksiensa ja hyvän informatiivisuuden vuoksi. Unityn sisäänrakennettu fysiikkamoottori tarjosi hyvän pohjan demon vaatimalle realistiselle VR-simulaatiolle, johon oli mahdollista tuottaa pelaajan selkeästi havaitsemia eripainoisia esineitä, mikä oli yksi tutkimusaihe. Tärkeäksi osoittautui myös oikeiden resurssipakettien valinta Unityn kauppapaikalta. Niiden avulla oli mahdollista lisätä realismia hyvin mallinnettujen 3D-mallien avulla.

Opinnäytetyön keskeisin työväline oli visuaalisen käyttöliittymän ohjelmointityökalu Bolt. Opinnäytetyön tekijä edustaa sellaista kehittäjää, jolla on ohjelmoinnin perustason osaamista muutamista ohjelmointikielistä, mutta ei syvää käytännön kokemusta. Kehitystyö olikin ensimmäinen käyttökokemus visuaalisen ohjelmoinnin käyttöliittymästä. Hyödylliseksi osoittautui helppo ja visuaalinen virheiden löytäminen. Boltin sisäänrakennettu ominaisuus näyttää virheitä merkitsemällä viallinen funktio visuaalisesti punaiseksi oli äärimmäisen hyödyllinen. Bolt poisti myös tarpeen perinteiselle aikaa vaativalle syntaksin opiskelulle, joten pääpaino pysyi logiikan ja ongelmien ratkomisessa. Bolt tarjosi mahdollisuuden muokata visuaalisia tilakaavioita myös pelin ollessa käynnissä. Täten virheiden etsiminen simuloitun pelikokemuksen kautta helpotti merkittävästi vianselvitystä. Boltin kyky tuottaa ja liikutella muuttujia eri peliesineiden välillä muodosti huomattavan hyödyn esimer-

kiksi useiden identtisten fysiikkaesineiden määrittelyssä. Boltin avulla opinnäytetyön kehitystyöhön pystyttiin toteuttamaan kaikki halutut ominaisuudet. Sen avulla olisi voinut toteuttaa vielä huomattavasti enemmän ominaisuuksia sekä tuottaa monimutkaisempia toteutuksia.

Kehitystyötä oli mielenkiintoista tehdä ja pelidemon tuottaminen tarjosi samalla hyvän mahdollisuuden syventää ymmärrystä ja dokumentoida prosessia tarkemmin. Mitä pidemmälle työ eteni, sitä selkeämmin tapahtui oppimista. Alussa tehtyjä asioita olisi haluttu opitun valossa tehdä uudelleen paremmin. Tämä olisi erityisesti näkynyt kartan suunnittelussa ja toteutuksessa sekä tiettyjen resurssipakettien valinnassa.

Toinen tutkimusteema oli käyttäjäkokemuksen onnistuminen. Tehokkain tapa selvittää käyttäjäkokemuksen tasoa on kuulla loppukäyttäjää valmiiksi suunnitellun kysymyslomakkeen kanssa sekä tehdä havaintoja heidän pelikerrastaan ja analysoida niitä. Pelin suunnittelu onnistui halutulla tavalla koska pelaajat pitivät pelin kulusta ja ratkottavista pulmista. VR-laitteiden käyttö ei aiheuttanut ongelmia vaan demon toiminnot ja liike koettiin miellyttäväksi. Pelin haastavuus osoittautui sopivaksi myös aikuisille ja sen kesto oli suunnitellun kaltainen. Testaajat eivät kiinnittäneet huomiota pelin toimivuuden epäkohtiin, joten voidaan olettaa pelikokemuksen olleen ongelmaton. Pienten yksityiskohtien runsaus sai kiitosta ja se monipuolisti pelikokemusta. Hieman haastavaksi osoittautui sellaisten testaajien löytäminen, joilla oli käytössään tarvittavat VR-laitteistot.

Kehitystyön prosessi herätti mielenkiinnon tuottaa mahdollisesti jopa kaupallisia versioita pelin teemasta. Pelin testauksen järjestäminen oli opettava kokemus, koska eri ihmiset ajattelivat asioista eri tavalla kuin peliä suunnitellessa oli tarkoitus. Pelaajat löysivät pulmiin ratkaisuja, joita pelin tekijä ei ollut tullut edes ajatelleeksi. Testaukset antoivat hyvän ymmärryksen niiden merkityksestä käyttäjäkokemuksen ymmärtämisessä sovelluskehittämisen maailmassa.

Lähteet

Auto Hand V1.5 Documentation N.d. Auto Hand resurssipaketin dokumentaatio Google drivessä. Viitattu: 21.3.2021. <https://drive.google.com/file/d/1jmSryTKb3P4NADaRau4Tbp7lLzZ5XUXd/view>

Gur Dotan 2015. Top 10 Unity Games Ever Made. Viitattu: 6.3.2021. <https://blog.soomla.com/2015/01/top-10-unity-games-ever-made.html>

Khronos Group Inc 2021. The OpenXR Specification. Viitattu: 6.3.2021. <https://www.khronos.org/registry/OpenXR/specs/1.0/html/xrspec.html>

Khronos Group Inc N.d. OpenXR. Viitattu: 6.3.2021. <https://www.khronos.org/openxr/>

Unity Technologies 2020. Bolt visual scripting is now included in all Unity plans. Viitattu: 6.3.2021. <https://blogs.unity3d.com/2020/07/22/bolt-visual-scripting-is-now-included-in-all-unity-plans/>

Unity Technologies 2021. Asset store. Viitattu: 6.3.2021. <https://assetstore.unity.com/>

Unity Technologies 2021. Build Settings. Viitattu: 6.3.2021. <https://docs.unity3d.com/Manual/BuildSettings.html>

Unity Technologies 2021. Learn Unity. Viitattu: 6.3.2021. <https://learn.unity.com/>

Unity Technologies 2021. OpenXR Plugin. Viitattu: 6.3.2021. <https://docs.unity3d.com/Packages/com.unity.xr.openxr@0.1/manual/index.html>

Unity Technologies N.d. Auto Hand - VR Physics Interaction. Viitattu: 6.3.2021. <https://assetstore.unity.com/packages/tools/modeling/auto-hand-vr-165323>

Unity Technologies N.d. Flashlight PRO. Viitattu: 6.3.2021. <https://assetstore.unity.com/packages/3d/props/tools/flashlight-pro-53053>

Unity Technologies N.d. Made with. Viitattu: 6.3.2021. <https://unity.com/madewith>

Unity Technologies N.d. Our Company. Viitattu: 6.3.2021. <https://unity.com/our-company>

Unity Technologies N.d. Products. Viitattu: 6.3.2021. <https://unity.com/products>

Unity Technologies N.d. QA InterrogationRoom. Viitattu: 6.3.2021. <https://assetstore.unity.com/packages/3d/environments/urban/qa-interrogationroom-131783>

Unity Technologies N.d. QA Police Station. Viitattu: 6.3.2021. <https://assetstore.unity.com/packages/3d/environments/urban/qa-police-station-162515>

Unity Technologies N.d. Visual Scripting. Viitattu: 6.3.2021. <https://unity.com/products/unity-visual-scripting>

UX Planet 2019. Designing User Experience for Virtual Reality (VR) applications. Viitattu: 11.4.2021. <https://uxplanet.org/designing-user-experience-for-virtual-reality-vr-applications-fc8e4faadd96>

Liitteet

Liite 1. Pelitestausten tietojenkeruulomake

Pelitestaus

TIETOJENKERUULOMAKE

Pelaajan __ vastaukset:

Kysymys: Koitko pahoinvointia pelin aikana tai välittömästi pelin jälkeen?

Kysymys: Mikä oli vaikein pulma?

Kysymys: Mikä oli helpoin pulma?

Kysymys: Oliko pulmia sopiva määrä?

Kysymys: Häiritsikö yleisen teleportaatio-ominaisuuden puute?

Otettujen vinkkien määrä: ____

Vapaa kommenttikenttä:

Pelaajan __ vastaukset:

Pelaajan tausta:

	Pelin kesto tähän saakka: h.min.sek
Pulma 1	
Pulma 2	
Pulma 3	
Pulma 4	
Pulma 5	
Pulma 6	
Pulma 7	
Pulma 8	
Pulma 9	
Pulma 10	

Liite 2. Pelaajan 1 pelitestausta

Pelaajan 1 vastaukset:

Kysymys: Koitko pahoinvointia pelin aikana tai välittömästi pelin jälkeen?

-En kokenut.

Kysymys: Mikä oli vaikein pulma?

-Pulma numero 9

Kysymys: Mikä oli helpoin pulma?

-Pulma numero 5

Kysymys: Oliko pulmia sopiva määrä?

-Yhteen pelikertaan sopiva määrä.

Kysymys: Häiritsikö yleisen teleportaatio-ominaisuuden puute?

-En kaivannut

Pelaaja 1	Aika: H/Min/Sek
Pulma 1	0.02.09
Pulma 2	0.23.20
Pulma 3	0.27.40
Pulma 4	0.31.25
Pulma 5	0.35.01
Pulma 6	0.49.20
Pulma 7	0.54.15
Pulma 8	1.01.24
Pulma 9	1.15.54
Pulma 10	1.19.50

Otettujen vinkkien määrä:

9

Vapaa kommenttikenttä:

Snap turning -liikkumisvaihtoehto asetuksen taakse. Hämäyspulmat saattavat viedä iloa kokemattomalta pakohuonepelaajalta.

Pelaajan tausta:

Useita VR-pakohuonepelejä, mutta ei kokemusta oikean elämän pakohuonepeleistä.

Liite 3. Pelaajan 2 pelitestausta

Pelaajan 2 vastaukset:

Kysymys: Koitko pahoinvointia pelin aikana tai välittömästi pelin jälkeen?

-Alkuun lievää pahoinvointia, kunnes ehdin tottua pelissä liikkumiseen.

Kysymys: Mikä oli vaikein pulma?

-Pulma numero 9

Kysymys: Mikä oli helpoin pulma?

-Pulma numero 5

Kysymys: Oliko pulmia sopiva määrä?

-Sopiva määrä. Pelissä ei ole tallennusominaisuutta, joten pulmia on mielestäni sopiva määrä yhteen pelikertaan.

Kysymys: Häiritkö yleisen teleportaatio-ominaisuuden puute?

-Ei niinkään, mutta voisi olla hyvä lisä.

Pelaaja 2	Aika: H/Min/Sek
Pulma 1	0.02.10
Pulma 2	0.24.50
Pulma 3	0.30.20
Pulma 4	0.46.45
Pulma 5	0.49.37
Pulma 6	0.53.50
Pulma 7	1.02.05
Pulma 8	1.12.01
Pulma 9	1.24.12
Pulma 10	1.40.45

Otettujen vinkkien määrä:

4

Vapaa kommenttikenttä:

Iso plussa ylimääräisestä krääsystä, joka häiritsee loogista ajattelua ja tuo sopivasti haastetta. Yhteenvetona hyvä toteutus. Pulmien ajatus ja graafinen maailma sopivat toisiinsa hyvin. Yksityiskohtiin toivoisin lisää toiminnallisuutta.

Pelaajan tausta:

Ei kokemusta VR-pakohuonepeleistä, mutta pelannut yhden oikean elämän pakohuoneen.

Liite 4. Pelaajan 3 pelitestausta

Pelaajan 3 vastaukset:

Kysymys: Koitko pahoinvointia pelin aikana tai välittömästi pelin jälkeen?

-En ollenkaan.

Kysymys: Mikä oli vaikein pulma?

-Pulma numero 9

Kysymys: Mikä oli helpoin pulma?

-Pulma numero 5

Kysymys: Oliko pulmia sopiva määrä?

-Yksin pelattaessa sopiva määrä, mutta esim moninpelaajan kokemuksena pulmien määrä voisi jäädä vajaaksi.

Kysymys: Häiritsikö yleisen teleportaatio-ominaisuuden puute?

-Ei

Pelaaja 3	Aika: H/Min/Sek
Pulma 1	0.01.54
Pulma 2	0.07.45
Pulma 3	0.15.06
Pulma 4	0.23.25
Pulma 5	0.27.09
Pulma 6	0.30.40
Pulma 7	0.37.20
Pulma 8	0.43.22
Pulma 9	0.55.17
Pulma 10	0.55.46

Otettujen vinkkien määrä:

0

Vapaa kommenttikenttä:

Sisustusta voisi olla enemmän. Pelikokemus todella mielenkiintoinen ja pulmat sopivan haastavia. Pulmien ratkaisuun vaikuttavat komponentit voisivat olla paremmin piilossa.

Pelaajan tausta:

Pelannut useita VR-pakohuonepelejä ja yli 20 oikean elämän pakohuonepelejä.

Liite 5. Pelaajan 4 pelitestausta

Pelaajan 4 vastaukset:

Kysymys: Koitko pahoinvointia pelin aikana tai välittömästi pelin jälkeen?

-Ei kokemus oli mieluista.

Kysymys: Mikä oli vaikein pulma?

-Pulma numero 9

Kysymys: Mikä oli helpoin pulma?

-Pulma numero 1

Kysymys: Oliko pulmia sopiva määrä?

-Vaikeustasoon nähden sopiva määrä.

Kysymys: Häiritsikö yleisen teleportaatio-ominaisuuden puute?

-Teleportaatio on pelimekaniikkana huono.

Pelaaja 4	Aika: H/Min/Sek
Pulma 1	0.02.26
Pulma 2	0.10.52
Pulma 3	0.17.01
Pulma 4	0.23.44
Pulma 5	0.31.06
Pulma 6	0.34.24
Pulma 7	0.36.03
Pulma 8	0.38.01
Pulma 9	1.01.31
Pulma 10	1.05.38

Otettujen vinkkien määrä:

0

Vapaa kommenttikenttä:

Paljon kivoja ylimääräisiä esineitä ja tauluja. Pulmien looginen vaikeustason kasvu on hyvässä suhteessa ja vaikeammat pulmat selvästi ovat pelin lopussa. Salausavain puzzle tuntui hankalalta, mutta jälkikäteen pidän sitä puzzlea hyvänä ja loogisena.

Pelaajan tausta:

Ei aikaisempaa kokemusta VR-pakohuonepeleistä, mutta pelannut useamman oikean elämän pakohuonepelin.