

Kauno technologijos universitetas

Informatikos fakultetas

Mokomoji virtualiosios realybės programa

Profesinės praktikos ataskaita

Erikas Mackevičius	(parašas)
Studentas	(data)
Andrius Paulauskas	(parašas)
Vadovas	(data)
Doc. Giedrius Ziberkas	(parašas)
Dėstytojas	(data)

Turinys

Įvadas.....	4
1. Analizės dalis	5
1.1. Informacija apie įmonę.....	5
1.1.1. Įmonės veiklos sritis, specializacija	5
1.1.2. Įmonės vizija	5
1.1.3. Įmonės misija	5
1.1.4. Kokiais būdais įmonė siekia įgyvendinti savo misiją ir viziją	5
1.2. Informacija apie įmonėje naudojamą IT	5
1.3. Įmonės veiklos procesai.	6
1.3.1. Projekto inicijavimas.....	6
1.3.2. Projekto koncepcijos analizė.....	6
1.3.3. Kainos ir investicijų analizė.	6
1.3.4. Projektų vykdymas, pristatymas ir dokumentacija.	6
1.3.5. Komandinis darbas.....	6
1.3.6. Inovacijos.	6
1.3.7. Produktų vystymo procesai.	7
2. Praktikos dalis	8
2.1. Techninė užduotis	8
2.1.1. Sistemos apibrėžimas	8
2.1.2. Bendras veiklos tikslas	8
2.1.3. Sistemos pagrįstumas	8
2.1.4. Konkurencija rinkoje.....	8
2.1.5. Prototipai ir pagalbinių informacija.....	9
2.1.6. Siūlomas sprendimas.....	9
2.1.7. Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti	9
2.2. Reikalavimų specifikacija	9
2.2.1. Komercinė specifikacija.....	9
2.2.2. Sistemos funkcijos	10
2.2.3. Apribojimai	10
2.2.4. Duomenų ir rezultatų specifikacija	10
2.2.5. Vartotojo sąsajos specifikacija	11
2.2.6. Realizacijai keliami reikalavimai.	13
2.2.7. Techninė specifikacija.....	13
2.3. Sistemos projektavimo eiga ir jos projektas.....	13
2.3.1. Projektavimo valdymas ir eiga.....	13
2.3.2. Projektavimo technologija	14
2.3.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos	14
2.3.4. Informacijos srautai.....	15
2.3.5. Sistemos sudėtis	17
2.3.6. Komponentai	21
2.3.7. Duomenų kontrolė.....	22
2.4. Testavimo eiga ir rezultatai	25

2.4.1. Teleportacijos funkcionalumas	25
2.4.2. Objektų manipuliavimo funkcionalumas	25
2.4.3. Prietaisų sujungimo funkcionalumas	26
2.4.4. Scenarijaus valdymo funkcionalumas	27
2.5. Dokumentacija	27
2.6. Sistemos reikalavimai	27
2.7. Sistemos naudojimas	28
3. Refleksija.....	36
Išvados	37
Literatūra	38

Ivadas

Šios praktikos užduotis buvo sukurti virtualios realybės simuliaciją, skirtą apmokyti vartotojus aparatinės įrangos apjungimo ypatybių. Sistema buvo orientuota specifiskai į transliavimo aparatinės įrangos scenarijus. Galutinis tikslas buvo turėti veikiančią ir patogią virtualios realybės simuliaciją, kuri leistų vartotojui santykiauti su aplinka ir įvairiais joje esančiais įrenginiais, juos apjungti pagal tam tikrus scenarijus ir patikrinti ar apjungimas buvo teisingas. Praktika atlikta Kauno Technologijų Universitete, specifiskai Virtualios ir papildytos realybės laboratorijų skyriui. Kauno Technologijų universitetas yra plačios aprėpties universitetas, glaudžiai bendradarbiaujantis su įvairiomis verslo įmonėmis. Virtualios realybės laboratorijų skyriaus veikla daugiausia susijusi su virtualios ir papildytos realybės įrenginiais, bei virtualių aplikacijų ir projektų kūrimu.

1. Analizės dalis

1.1. Informacija apie įmonę

1.1.1. Įmonės veiklos sritis, specializacija

KTU – lyderiaujantis plačios aprėpties universitetas, glaudžiai bendradarbiaujantis su verslu. Universitete vykdomos įvairių kryptių studijos ir moksliniai tyrimai. Universitetas stengiasi formuoti profesionalius, savo sritį išmanančius ekspertus, kurie turėtų galimybę plėsti savo darbo sritį ir turėti įtaką visuomenėje. Universitetas turi plačius žinių ir resursų šaltinius, prieinamus studentams, suteikia praktikos galimybes, bei skatina verslumą.

1.1.2. Įmonės vizija

Tarptautinis informacinių technologijų studijas ir mokslinius tyrimus vykdomas centras, pripažintas Baltijos jūros regione. ^[1]

1.1.3. Įmonės misija

Teikti moksliniais tyrimais grįstas tarptautinio lygio studijas, prisidedant prie tvaraus visuomenės ugdymo, kuriant bei perduodant žinias ir novatoriškas technologijas darniai valstybės raidai ir inovacijų plėtojimui. ^[1]

1.1.4. Kokiais būdais įmonė siekia įgyvendinti savo misiją ir viziją

KTU - rūpestinga bendruomenė, kuri padeda augti vienas kitam akademinėje aplinkoje, plečiant žinias ir žmogaus supratimą. Universitetas palaiko ryšius ir bendradarbiauja su įvairiomis įmonėmis visoje Lietuvoje, į paskaitas dažnai kviečiami įvairių sričių ekspertai. KTU „WANTed“ programa suteikia studentams galimybes rasti praktikos bei darbo vietas, gauti konsultacijas iš įvairių įmonių, planuoti savo studijas atsižvelgiant į ateities karjeros galimybes.

1.2. Informacija apie įmonėje naudojamą IT

KTU svarbu inovacijos, naujausių technologijų kūrimas, įvairių sričių moksliniai tyrimai. Universitetas turi ryšius su įvairiais studijų, mokslo ir verslo centrais, verslo įmonėmis. Dėl šios priežasties universitetas turi prieigą prie naujausių ir moderniausių informacinių technologijų, bei išteklių. Virtualios ir papildytos realybės laboratorija dirba su įvairiais VR prietaisais, todėl svarbu turėti pajėgią kompiuterinę įrangą. Visi laboratorijoje naudojami kompiuteriai turi Intel i7-8700 procesorius. Kompiuteriuose naudojamos vaizdo plokštės: Nvidia RTX 2060, Nvidia GTX 1070, Nvidia GTX 1080Ti. Laboratorijoje naudojami įvairūs skirtingi virtualios realybės įrenginiai: HTC Vive Pro, Oculus Rift S, Oculus Quest 1 ir Oculus Quest 2. VR įrenginių įvairovė leidžia testuoti aplikacijas įvairiose platformose ir skirtingais pajėgumais. HTC Vive Pro ir Oculus Rift S reikalauja kompiuterio, bet jų galia priklauso nuo kompiuterio galimybių. Oculus Quest ir Quest 2 yra belaidžiai savarankiški VR įrenginiai,

^[1] Kauno technologijos universitetas. (2022). [žiūrėta 2022-03-10]. Prieiga per internetą <https://ktu.edu/universitetas/>

nerieikalaujantys kompiuterio. Laboratorijoje plačiai naudojamos 3D aplikacijų kūrimo priemonės. Unity3D ir Unreal Engine 4, vienos iš geriausių ir populiariausių aplikacijų kūrimo priemonių. Taip pat naudojamas Blender įvairių modelių kūrimui ir įvairūs AdobeSubstance3D paketai. Virtualios laboratorijos turi visas priemones reikalingas virtualių aplikacijų kūrimui, testavimui ir naudojimui.

1.3. Įmonės veiklos procesai.

1.3.1. Projekto inicijavimas.

KTU projektų inicijavimas priklauso nuo jų pobūdžio, bet visais atvejais organizuojami susitikimai su partneriais ir klientais, aptariama su projektu susijusi informacija, tokia kaip projekto laikotarpis, finansavimas, reikalavimai.

1.3.2. Projekto koncepcijos analizė.

Po pirmųjų susitikimų su klientais atliekama projekto apimties analizė, kurios metu yra įvertinami turimi resursai bei technologijų galimybės sukurti norimą produktą. Vyksta išsamios diskusijos su klientais, sudaroma projekto vizija.

1.3.3. Kainos ir investicijų analizė.

Universitetas retai kada investuoja į išorinio produkto sukūrimą, nebent projektas to reikalauja.

1.3.4. Projektų vykdymas, pristatymas ir dokumentacija.

Projektų vykdymas susideda iš etapų:

- Reikalavimų surinkimas ir analizė;
- Programinės įrangos kūrimas;
- Programinės įrangos dokumentacija (kuriam projektui pabaigoje arba iteracijomis);
- Projekto pristatymas užsakovui ir atlikimo įvertinimas.

1.3.5. Komandinis darbas.

Komandiniam darbui organizuoti naudojami įvairūs planavimo įrankiai. Slack ir Trello dažniausiai naudojami užduočių paskirstymui ir atlikimo sekimui. Projektai dažniausiai saugomi repozitorijose naudojant Github arba Bitbucket.

1.3.6. Inovacijos.

Inovacijos, jų kūrimas bei skatinimas yra svarbi universiteto dalis. KTU inovacijomis laiko ne tik naujų sprendimų ir technologijų kūrimą, bet ir inovatyvių verslo idėjų skatinimą ir plėtojimą. Pačiame KTU naudojami tik naujausi technologiniai sprendimai ir įranga. Universitetas glaudžiai seka naujų inovatyvių idėjų ir technologijų plėtrą ir stengiasi jas integruoti į savo veiklą. KTU svarbu ne tik naudoti inovatyvius sprendimus, bet ir juos kurti bei plėtoti. KTU Nacionalinis inovacijų ir verslo centras rūpinasi tarptautiniu lygiu pripažintų technologijų kūrimu bei inovatyvius produktus kuriančių įmonių

steigimu ir vystymu. Šio centro dėka egzistuoja tokios priemonės kaip KTU Startup Space ir KTU Atviros prieigos centras. Startup Space yra plačiai visuomenei atvira bendruomenė, skatinanti inovatyvių idėjų ir verslo plėtojimą. Atviros prieigos centras yra įrankis, leidžiantis verslo įmonėms ir įstaigoms prieiti KTU mokslininkų teikiamų paslaugų, bei tyrimų informacijos, kuri palengvina naujų technologinių sprendimų kūrimą.

1.3.7. Produktų vystymo procesai.

Universitete yra taikomi įvairūs produktų vystymo procesai. Tai dažniausiai priklauso nuo projekto pobūdžio ir reikalavimų. Dirbant su europiniais projektais ir valstybiniais užsakymais, dažniausiai taikomi procesai panašūs į krioklio metodą. Jeigu produktai kuriami įmonėms, dažniausiai taikomas iteracinis produkto vystymo modelis. Nepriklausomai nuo metodo, procesai visada įtraukia reikalavimų surinkimą ir analizę, projektavimą, programinės įrangos kūrimą ir jos testavimą. Programinės įrangos palaikymas dažniausiai nėra vykdomas, jeigu sukurtas projektas yra išorinis.

2. Praktikos dalis

2.1. Techninė užduotis

2.1.1. Sistemos apibrėžimas

Su „Unity3D“ varikliu sukurti virtualiosios realybės programą skirtą „Oculus Quest“ platformai. Sistemos turėtų realizuoti mokomuosius scenarijus, kurių metu vartotojas galėtų išmokti teisingai sujungti garso ir vaizdo transliavimo įrangą. Programoje turi būti realizuotas mokomasis scenarijus, kuriame naudotojas turi teisingai paruošti transliavimo įrangą. Šią įrangą turi sudaryti vaizdo kamera, transliavimo pultas, mikrofonas, garso monitorius bei kompiuterinė įranga (kompiuteris, monitorius, įvesties įrenginiai).

2.1.2. Bendras veiklos tikslas

Sukurti programą virtualiai realybei, kurioje būtų realizuojamas transliavimo įrangos komutavimo mokomasis scenarijus. Programa leis paruošti komutavimo scenarijus nereikalaujant fiziškai turėti aparatinės įrangos. Tai padės taupyti kaštus ir tuo pačiu išmokyti vartotoją komutavimo.

2.1.3. Sistemos pagrįstumas

Poreikis mokyti studentus apie transliavimo įrangą ir jos pajungimo ypatybes yra gana didelis, bet gauti visą įrangą ir ją dedikuoti mokymui yra sudėtinga ir neproduktyvu. VR programos paruošimas leidžia suteikti studentams informaciją apie komutavimą interaktyvioje aplinkoje, bet nereikalauja tikros įrangos.

2.1.4. Konkurencija rinkoje

- **Unimersiv** – virtualios realybės programa, leidžianti vartotojui pasirinkti vieną iš daugelio interaktyvių mokomųjų simuliacijų. Programos naudotojai gali pasirinkti iš daugelio kategorijų, tokių kaip istorija, kosmosas, anatomija.
- **Veative** – virtualios realybės platforma, turinti vieną iš didžiausių virtualių scenarijų bibliotekų. **Veative** biblioteka šiuo metu turi 700+ scenarijų iš įvairių mokslo šakų, virtualių turų, prezentacijų.
- **Engage** – virtuali platforma, leidžianti organizacijoms kurti virtualias aplinkas ir scenarijus skirtas mokymo tikslams, darbuotojų parengimui ir kitiems scenarijams.

Visos priemonės leidžia vartotojas patirti įvairius mokomuosius scenarijus, bet turi įvairių trūkumų. **Unimersiv** leidžia vartotojui išbandyti įvairius mokomuosius scenarijus, bet suteikia scenarijus tik tam tikrose kategorijose. Šios programos pagrindinis dėmesys skiriamas simuliacijoms istorijos, kosmoso, bei anatomijos kategorijose. **Veative** turi didelį scenarijų pasirinkimą, suteikia vartotojui galimybę patirti scenarijus ir eksperimentus biologijos, chemijos, fizikos bei matematikos srityse. Ši priemonė taip pat suteikia vartotojui galimybę mokytis kalbų naudojant virtualius pokalbių scenarijus, bei aplankyti įvairias pasaulio vietas naudojant turų simuliacijas. **Engage** neturi tam tikrų nustatytų scenarijų, bet suteikia galimybę organizacijoms kurti naujus scenarijus naudojantis jų įrankiu. Tai leidžia kompanijoms kurti įvairius scenarijus pagal paskirtį. Visos šios kompanijos suteikia galimybę vartotojams patirti įvairius mokomuosius scenarijus, bet **Unimersiv** yra apribota

kategorijų stokos, **Veative** taip pat ribota kategorijų stokos ir ne visos simuliacijos suteikia vienodą kiekį interaktyvumo. **Engage** scenarijai gali būti įvairių kategorijų, bet jie yra riboti to, jog kiekvienas scenarijus kuriamas naudojant **Engage** įrankiais, todėl jeigu **Engage** sistema neturi tam tikrų funkcijų, nėra galimybės scenarijuose įgyvendinti tam tikrų funkcijų. Kuriamą virtualios realybės įrenginių sujungimų sistema susitelkia į vieną kategoriją, prietaisų sujungimą ir validaciją virtualioje erdvėje. Nė viena iš aptartų sistemų nesiūlo būtent tokio tipo scenarijų, **Engage** leistų vartotojui sukurti panašų scenarijų, bet jis būtų ribotas dėl reikiamybės naudoti **Engage** įrankius.

2.1.5. Prototipai ir pagalbinių informacija

Kuriant programą nenaudojami joki prototipai.

2.1.6. Siūlomas sprendimas

Sprendimas bus pritaikytas konkretiems scenarijams, kurie skirti mokyti studentus įrangos komutavimo. Programos eigoje, studentai gaus informacijos apie įvairius transliavimo įrenginius, jų tarpusavio sąveiką ir apjungimą.

2.1.7. Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti

Sistema kuriama vieno programuotojo. Jai skiriama apie 260 valandų. Kūrimui naudojamas paprastas darbinis kompiuteris, pajėgus efektyviai naudoti Unity3D aplikaciją bei Oculus Link. Programos testavimui reikalingi virtualios realybės akiniai Oculus Quest. Unity 3D aplikacija neturi didelių reikalavimų, jie daugiausia priklauso nuo kuriamos aplikacijos, bet rekomenduojama naudoti kompiuterį su Windows 7 arba Windows 10 64-bit operacinėmis sistemomis, CPU su x64 architektūra ir turėti DX10, DX11 arba DX12 palaikančią vaizdo plokštę. Oculus Link turi minimalius reikalavimus, norint paleisti aplikaciją naudojant kompiuterio resursus. Minimalūs reikalavimai yra Intel i5-4590 arba AMD Ryzen 5 1500X procesorius, NVIDIA GeForce GTX 970 arba AMD 400 Series vaizdo plokštė, bent 8GB RAM, Windows 10 operacinė sistema ir viena USB jungtis. Rekomenduojama USB jungties 3.0 versija maksimaliam našumui.

1 lentelė Minimalūs Oculus Link reikalavimai

Komponentas	Rekomenduojamos minimalios specifikacijos
Procesorius	Intel i5-4590 / AMD Ryzen 5 1500X
Vaizdo plokštė	NVIDIA GeForce GTX 970 / AMD 400 Series
Atmintis	8 GB+ RAM
Operacinė sistema	Windows 10
USB jungtis	1x USB jungtis (rekomenduojama 3.0)

2.2. Reikalavimų specifikacija

2.2.1. Komerčinė specifikacija

- Projekto užsakovas: Andrius Paulauskas
- Projekto vykdytojas: Erikas Mackevičius
- Produkto vartotojai: dėstytojai ir studentai

- Produkto kūrimo laikotarpis: 2 mėnesiai
- Skiriamas biudžetas: Produkto kūrimui biudžetas neskiriamas
- Pagrindinės priemonės:
 - Kompiuteris su Unity3D ir Oculus programine įranga
 - Oculus Quest virtualios realybės akiniai
 - Oculus Link laidas

2.2.2. Sistemos funkcijos

Sistemos funkciniai reikalavimai:

- Realizuota virtuali aplinka leidžianti vartotojui judėti, naudoti teleportacijos funkciją ir sąveikauti su objektais;
- Realizuota virtuali įranga su skirtingo tipo jungtimis, bei sistema leidžianti sujungti įrangą tarpusavyje naudojant laidas;
- Realizuota sistema skirta validuoti įrangos sujungimui. Sistema leidžia sudaryti vertinimo kriterijų konfigūraciją, ją validuoja ir sudaro rezultatų suvestinę;
- Sukurti trimačiai aplinkos ir įrangos moduliai;
- Realizuota dalelių sistema;
- Paruošti ir integruoti garso efektai.

2.2.3. Apribojimai

Sistema kuriama naudoti Oculus Quest ir Oculus Quest 2 operacinėje sistemoje. Darbo aplinkai reikalaujamas darbinis kompiuteris, pajėgus paleisti Unity3D ir Oculus Link. Sistemos testavimui reikalingi Oculus Quest virtualios realybės akiniai. Sistemos kūrimo terminas – 2 mėnesiai. Sistemos kūrimui nėra skiriamas joks biudžetas, todėl kuriant naudojami nemokami įrankiai ir kiti resursai.

2.2.4. Duomenų ir rezultatų specifikacija

Aplikacijoje duomenys skaidomi į jungties informaciją ir rezultatus. Jungties informacija bus naudojama tikrinant jungčių tipus, jų sujungimo galimybes ir komutacijos įrangos specifikacijas. Rezultatai bus naudojami atvaizduoti vartotojui komutacijos įrangos sujungimo scenarijaus būseną. Jungtys turės duomenų tipus, kuriuose bus apibrėžtos jų specifikacijos. Jungtis gali turėti pavadinimą, tipą, versiją, papildomus apibūdinimus, tokius kaip galimybė praleisti elektrą arba interneto ryšį. Ši informacija bus naudojama vykdant prietaisų sujungimo validavimą. Rezultatai bus apibrėžti paprasčiau. Šiuo atveju svarbu žinoti kurie prietaisai turi būti sujungti kuriomis jungtimis ir šią informaciją naudoti validuojant sujungimo scenarijų. Rezultatų duomenys turės prietaisą, sąrašą jungčių, kuriomis jis turi būti prijungtas prie kitų prietaisų, bei sujungimo būseną, kurią bus galima naudoti rezultatų atvaizdavimui.

2 lentelė Jungties laukų duomenų informacija

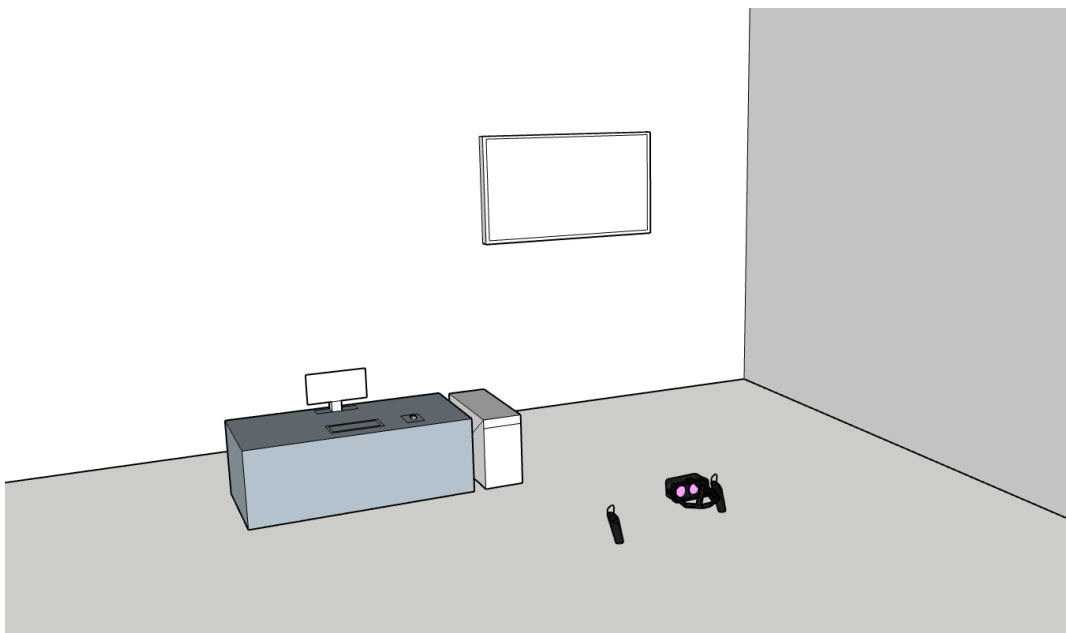
Jungties laukų informacija									
Pavadinimas	Tipas	Pavyzdys							
Name	String	HDMI	USB		Audio		DC		SDI
Type	String	Type A	Type B	2.5mm	3.5mm	SD	HD	3G	
Version	Float	2.0		2.2		3.1		3.2	
Is powered	Bool	True				False			
Has ethernet	Bool	True				False			
Cable diameter	Float	0.644		0.511		0.405		0.321	

3 lentelė Rezultatų laukų duomenų informacija

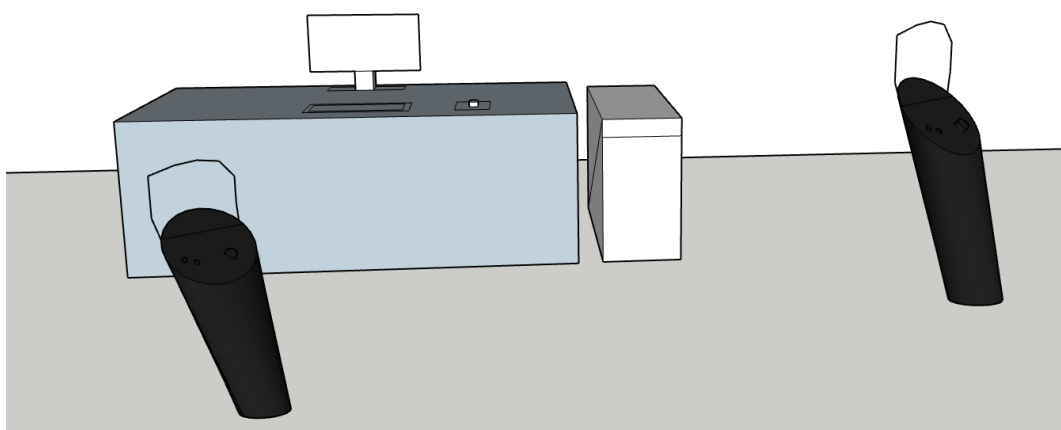
Rezultatų informacija				
Pavadinimas	Tipas	Pavyzdys		
Device	String	Computer	Keyboard	HDMI Splitter
Connections	List<String>	<ul style="list-style-type: none"> TV HDMI OUT Monitor HDMI IN 	<ul style="list-style-type: none"> Computer USB 	<ul style="list-style-type: none"> Computer HDMI OUT Monitor HDMI IN
Status	Float	True		False

2.2.5. Vartotojo sąsajos specifikacija

Projektas kuriamas virtualiai realybei. Esantis virtualioje aplinkoje vartotojas turės rankas, kurios judės naudojant Oculus Touch valdiklius. Vartotojo aplinka bus kambarys su įvairiais prietaisais, kuriuos jis turės sujungti. Siekiant palengvinti kūrimo procesą, sudarytas vartotojo aplinkos prototipas.

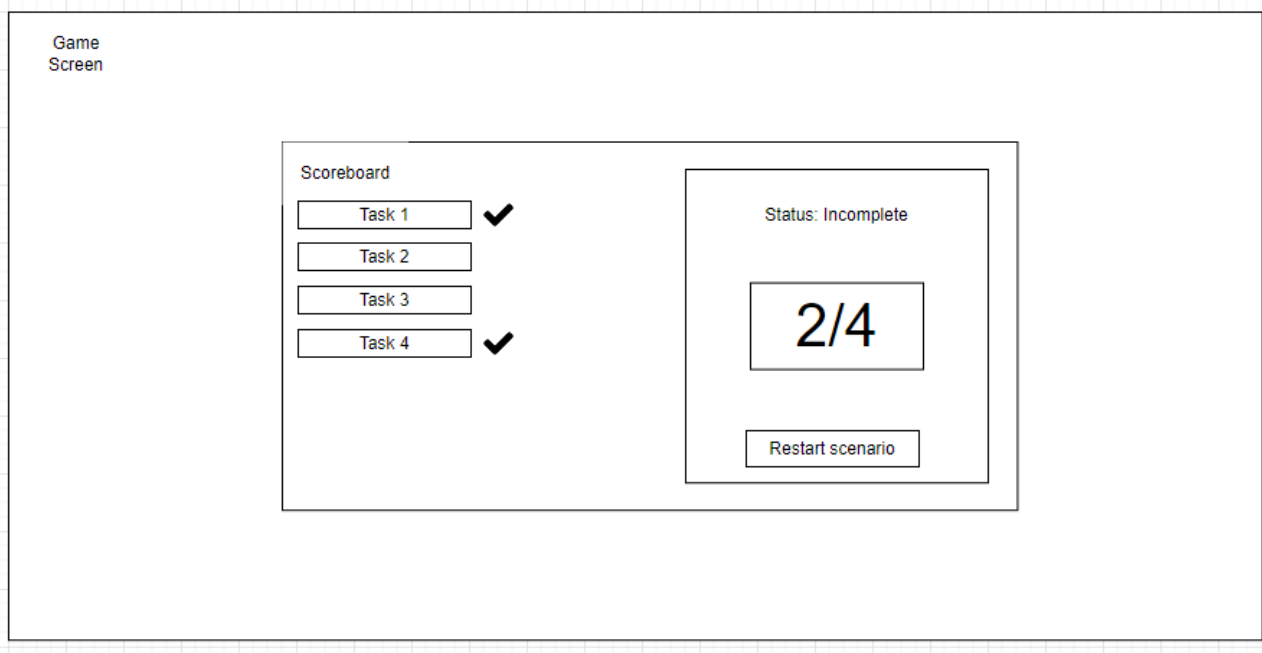


1 pav. Scenarijaus aplinkos prototipas (pavaizduoti ne visi įrenginiai)



2 pav. Scenarijaus vaizdas iš vartotojo perspektyvos

Vienintelis elementas reikalaujantis UI integracijos išvestyje bus rezultatų atvaizdavimo sistema.



3 pav. Rezultatų atvaizdavimo UI prototipas

Įvykdžius arba vykdant komutavimo scenarijų, vartotojas galės atidaryti rezultatų lentelę, kurioje bus išvardintos užduotys ir jų atlikimo būseną. Vartotojas turės galimybę bet kurio metu paleisti scenarijų iš naujo.

2.2.6. Realizacijai keliami reikalavimai.

- Programa turi veikti 72 Hz darbinio dažniu, kad nesukeltų vartotojui diskomforto;
- Programa turi būti optimizuota ir turėti kuo mažesnę uždelimą tarp vartotojo veiksmų realybėje ir jų atvaizdavimo virtualioje aplinkoje;
- Programoje neturėtų būti ryškių arba mirgančių vaizdų, tam kad vartotojui nesukeltų diskomforto.

2.2.7. Techninė specifikacija.

Programai naudoti reikalingi VR akiniai Oculus Quest arba Oculus Quest 2. Norint programą leisti per kompiuterį, reikalinga Oculus programinė įranga, bei įranga leidžianti prijungti VR akinius prie kompiuterio.

2.3. Sistemos projektavimo eiga ir jos projektas

2.3.1. Projektavimo valdymas ir eiga

Prieš pradėdant dirbti prie projekto buvo aptarti sistemos reikalavimai, programinė ir fizinė įranga, kuri bus naudojama jos kūrimui ir testavimui. Sudarytas planas, kuriame atvaizduoti sistemos kūrimo žingsniai ir jiems įgyvendinti skirtas laikas. Plano etapai suskirstyti pagal sistemos funkcijas, pirmiausia planuojama realizuoti pagrindines VR funkcijas, tokias kaip judėjimas erdvėje ir objektų manipuliacija. Tuomet planuojama realizuoti prietaisus ir jungtis, jų sujungimo galimybes. Galiausiai planuojama sukurti scenarijaus validavimo sistemą. Po šių žingsnių seka sistemos apipavidalinimas

trimačiais aplinkos ir prietaisų modeliais, pridedami garso ir vaizdiniai efektai. Atliekamas sistemos testavimas. Projektavimo metu detaliau aptarta sistemos struktūra, planuojamos naudoti duomenų struktūros ir tipai. Baigus projektavimą, pradėta kurti sistema glaudžiai remiantis kūrimo planu. Sistemos kūrimo pabaigoje atlikti testai, remiantis sudarytais testo scenarijais.

4 lentelė Sistemos kūrimo planas

Laiko tarpas	Planuojami atlikti veiksmai
2022-03-22 – 2022-03-29	Saveikos ir orientavimosi virtualioje realybėje realizavimas: <ul style="list-style-type: none"> • Realizuojama teleportavimo funkcija orientavimuisi virtualioje erdvėje; • Paruošiama bazinė žaidimo objekto realizacija, kuri skirta vartotojui sąveikauti virtualioje realybėje.
2022-03-30 – 2022-04-06	Virtualiosios įrangos ir komutavimo funkcijų realizacija: <ul style="list-style-type: none"> • Realizuojamos transliavimo įrangos jungtys ir jų tipai; • Realizuojama jungčių sujungimo galimybė.
2022-04-07 – 2022-04-14	Komutavimo rezultato vertinimo realizacija: <ul style="list-style-type: none"> • Vertinimo kriterijų konfigūracijos realizacija; • Komutavimo atitikimo konfigūracijai validavimas; • Vertinimo rezultatų suvestinės sudarymas ir atvaizdavimas.
2022-04-15 – 2022-04-21	Programos turinio paruošimas: <ul style="list-style-type: none"> • Trimačių modelių aplinkai sukūrimas; • Trimačių įrangos modelių sukūrimas; • Dalelių sistemos sukūrimas; • Garso efektų paruošimas.
2022-04-22 – 2022-04-29	Sistemos testavimas ir defektų taisymas.

2.3.2. Projektavimo technologija

Sistemos kūrimo progresui sekti ir sistemai saugoti buvo naudojama Github platforma. Kadangi sistema kuriama vieno žmogaus, nebuvo reikiamybės naudoti dedikuotų užduočių paskirstymo ir sekimo sistemų, tokių kaip Trello ir Jira. Sistemos kūrimo planas kartu su Github repozitorija suteikė pakankamai informacijos sistemos kūrimo organizavimui. Papildomai informacijai ir užrašams dokumentuoti buvo naudojami paprasti .txt failai. Projekto prototipo vaizdam formuoti naudojamas nemokamas internetinis įrankis Sketchup. Šis įrankis leido patogiai ir lengvai sukurti virtualios aplinkos vaizdų eskizus. UI eskizams kurti naudotas Diagrams.net įrankis.

2.3.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos

C sharp, Android, Windows, Oculus quest Sistemos kūrimui naudota Unity programinė įranga, kodas rašomas C# programavimo kalba. Unity paleisti naudotas kompiuteris su Windows 10 operacine sistema. Sistemai paleisti ir testuoti naudotas Oculus Quest su Android operacine sistema. Oculus Link programinė įranga naudota susieti virtualios realybės akinius su kompiuteriu, derinimo tikslais.

5 lentelė Techninė ir programinė įranga naudota sistemos kūrimui ir testavimui

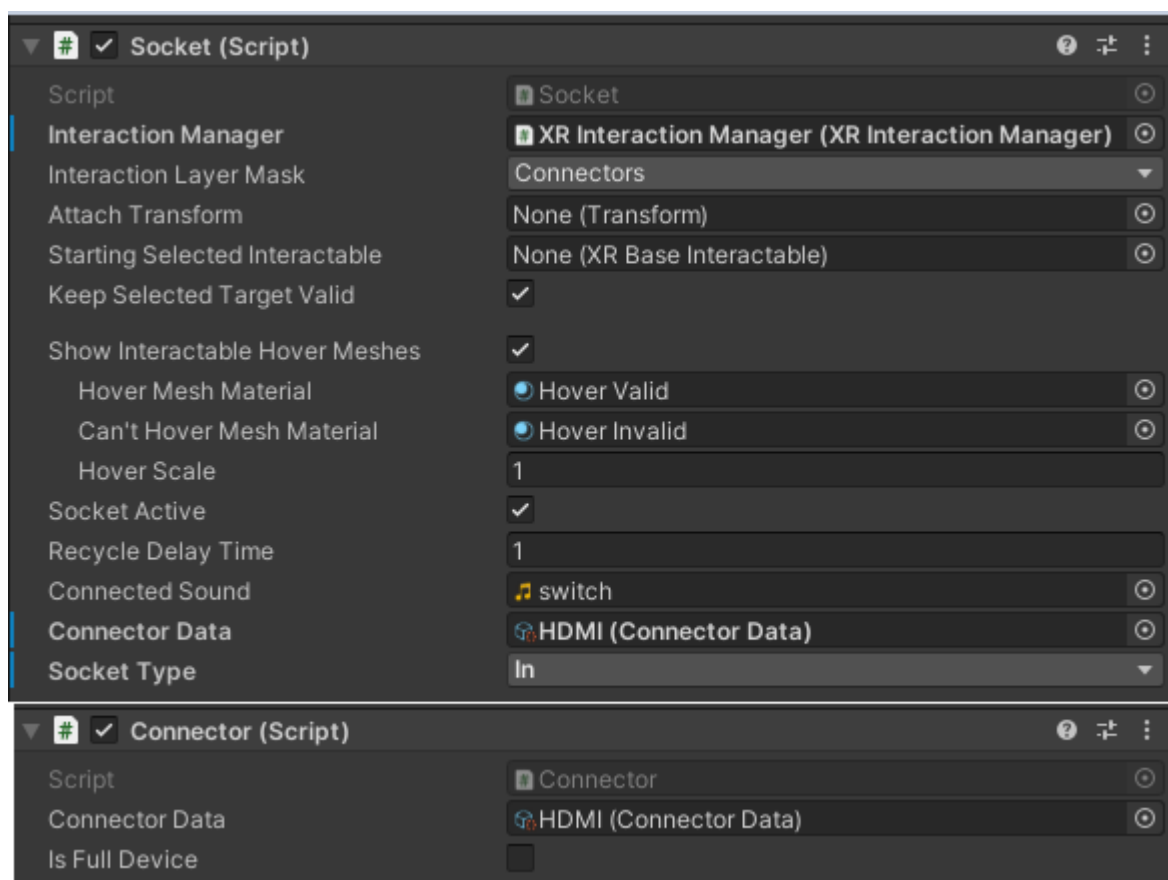
Sistemos kūrimo informacija		
Kategorijos	Kūrimui ir derinimui naudotos priemonės	Sistemos paleidimui ir testavimui naudotos priemonės
Techninė įranga	Kompiuteris	Oculus Quest virtualios realybės akiniai
Programinė įranga	Unity, Oculus Link	-
Operacinė sistema	Windows 10	Android
Programavimo kalba	C#	-

2.3.4. Informacijos srautai

Sistemos veikimo metu, pagrindiniai informacijos mainai vyksta vykdant jungčių sujungimą su prietaisais. Jungtys sujungimo metu komunikuoja su prietaisų lizdais, tokiais kaip HDMI, USB, Audio ir t.t. Įvykdžius dviejų prietaisų sujungimą, jungties duomenys yra nusiunčiami prietaiso valdymo klasei ir saugomi tol, kol prietaisai neatjungiami vienas nuo kito. Tuo pačiu metu, jungtis nusiunčia sujungimo garso failą, garso valdymo klasei.

Informacijos srautai:

- **Connector** su **Socket** – sujungiant jungtį su lizdu, vyksta informacijos mainai siekiant validuoti sujungimą
 - **Connector** siunčia **Socket** jungties tipą (HDMI, USB, Audio ir t.t.) su papildoma informacija priklausomai nuo jungties. Jeigu kitas jungties galas jau yra sujungtas su prietaisu, **Socket** taip pat gauna prietaiso identifikacijos informaciją;
 - **Socket** siunčia **Connector** prietaiso, kuriam priklauso jungtis informaciją;



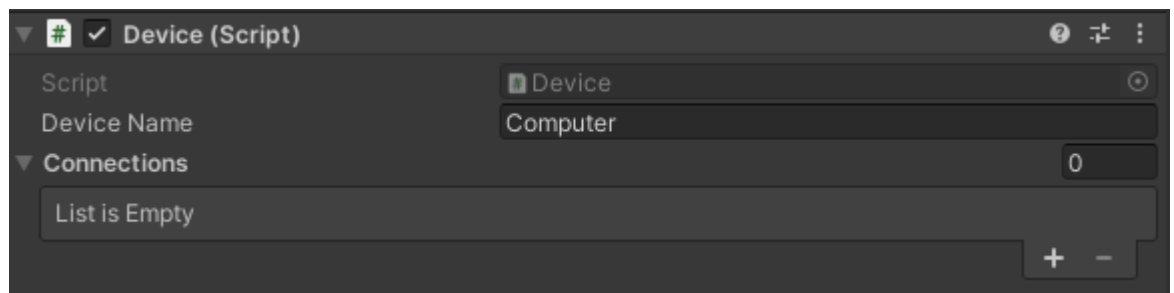
4 pav. Jungtys ir lizdai saugo savo tipo informaciją naudodami ScriptableObject duomenų ruošinius

- **Connector** su **SoundManager** – įvykstant sujungimą, jungtis nusiunčia garso valdymo klasei savo sujungimo garso failo informaciją. SoundManager klasė atsakinga už garso leidimą, bet pati nesaugo garso klipų. Kiti objektai nusiųsti šiai klasei užklausas su reikiama garso klipais.

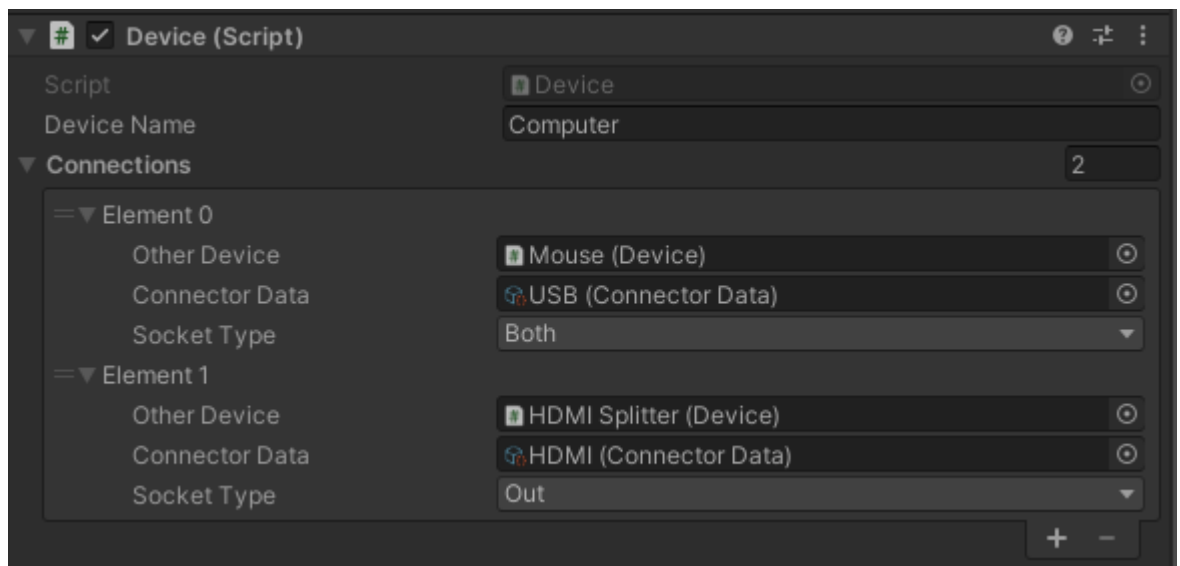
```
public void PlaySoundEffectAtPosition(AudioClip audioClip, Vector3 pos, float volumeScale = 1f)
{
    soundEffectsAudioSource.transform.position = pos;
    soundEffectsAudioSource.PlayOneShot(audioClip, volumeScale);
}
```

5 pav. SoundManager klasė gauna garso klipą ir poziciją iš kitų objektų

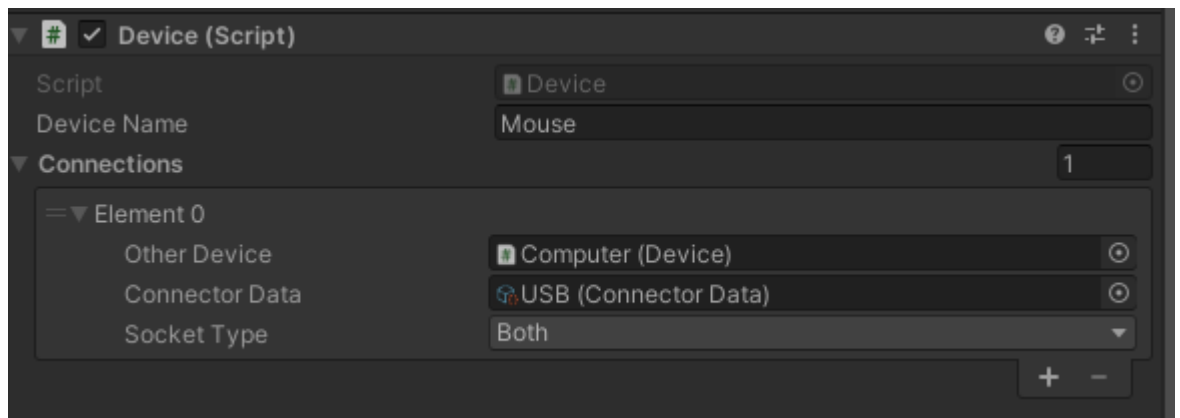
- **Socket** su **Device** – sujungus du prietaisus, jų jungtis nusiunčia sujungimo informaciją prietaisų valdymo klasėms. Vėliau ši informacija naudojama scenarijaus įgyvendinimo validacijai.



6 pav. Prietaisas neturi saugomų jungčių, kadangi neturi prijungtų kitų prietaisų.

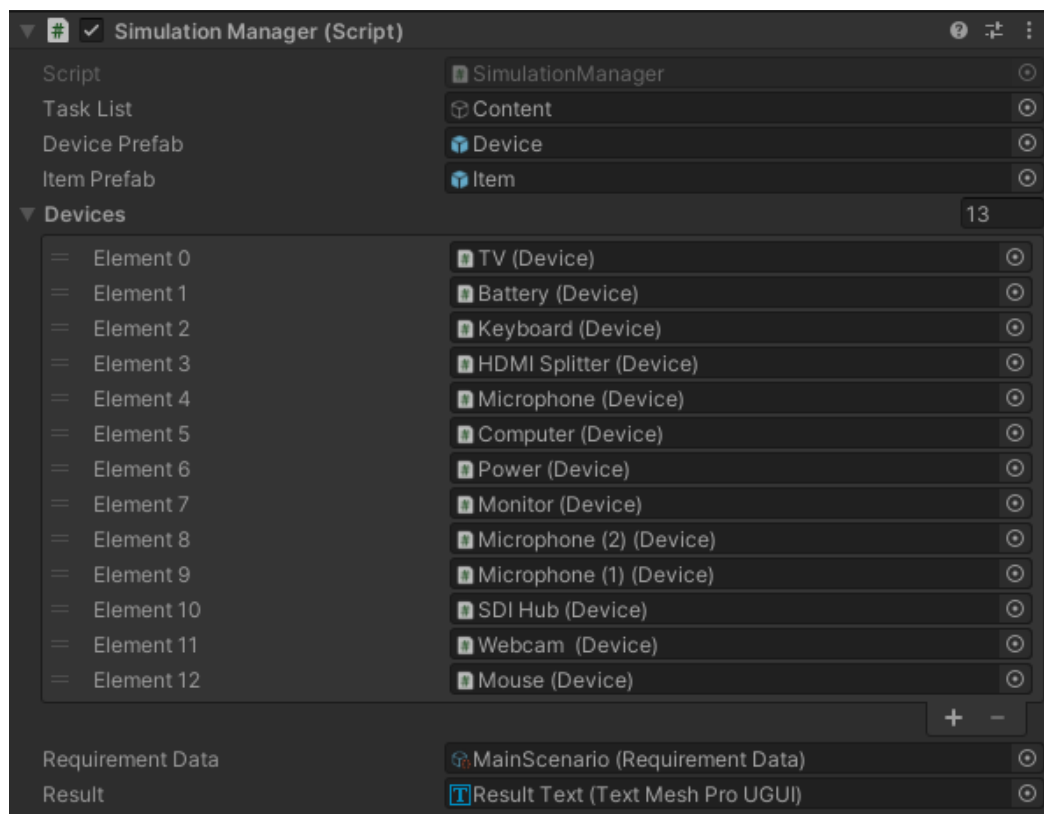


7 pav. Įvykdžius sujungimus sąrašas užsipildo sujungimų informacija



8 pav. Sujungimų informacija atspindi visuose sujungtuose prietaisuose

- **Device** su **SimulationManager** – vykdamas scenarijaus įgyvendinimo validavimą, prietaisai siunčia simuliacijos valdymo klasei savo jungčių informaciją. **SimulationManager** turi **ScriptableObject** apibūdinantį scenarijaus reikalavimus, kurio duomenys vėliau yra naudojami patikrinti ar sujungimai teisingi.



9 pav. SimulationManager turi prieigą prie prietaisų, bei scenarijaus įgyvendinimo reikalavimų.

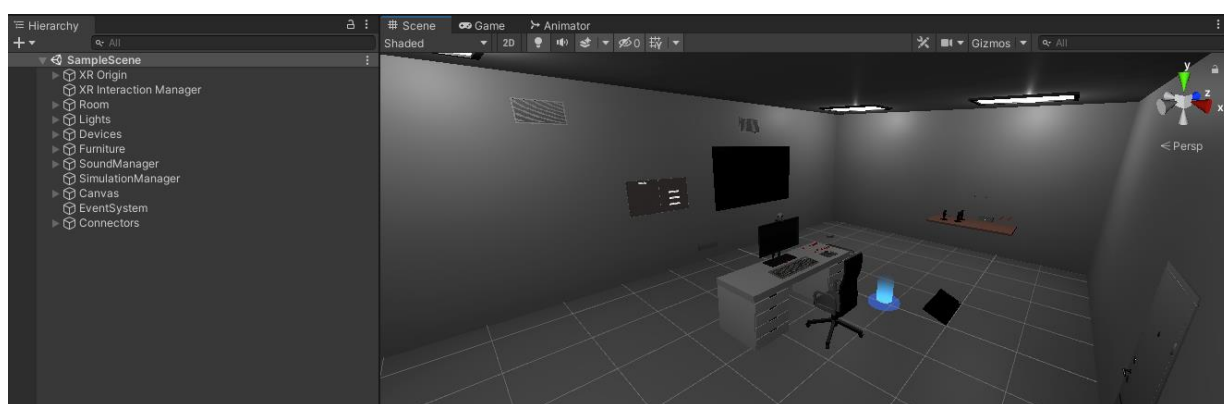
2.3.5. Sistemos sudėtis

Sistema susideda iš techninės ir programinės įrangos. Sistemai paleisti naudojami **Oculus Quest** virtualios realybės akiniai su dviem **Oculus Touch** valdikliais. Sistemos programinė įranga yra saugoma .apk faile, kuris naudojamas sistemos instaliacijai.

Name	Size	Packed Size	Modified
assets	153 568 527	35 285 044	
lib	65 199 448	23 476 776	
META-INF	54 785	21 899	
res	6 420	5 964	
AndroidManifest.xml	5 972	1 767	1981-01-01 01:01
classes.dex	127 888	59 710	1981-01-01 01:01
classes2.dex	3 300	1 509	1981-01-01 01:01
classes3.dex	7 900	4 151	1981-01-01 01:01
resources.arsc	1 888	1 888	1981-01-01 01:01
VRProject	0	0	1981-01-01 01:01

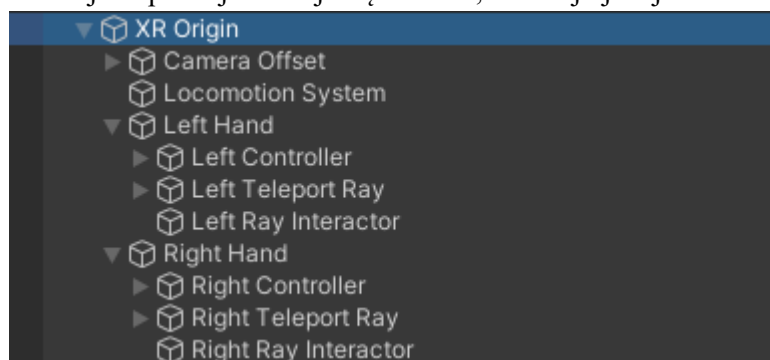
10 pav. .apk failo struktūra

Sistemos viduje, ji susideda iš aplinkos objektų, prietaisų, jungčių ir įvairių valdiklių. Valdikliai atsakingi už skaičiavimus, objektų valdymą, rezultatų skaičiavimą, garsus scenarijuje.



11 pav. Sistemos sudėtis

- **XR Origin** – objektas susidedantis iš pagrindinių VR dalių, tokių kaip vartotojo kamera, rankos, įvairūs spinduliai naudojami pozicijos ir objektų sekimui, vartotojo judėjimo valdiklis.



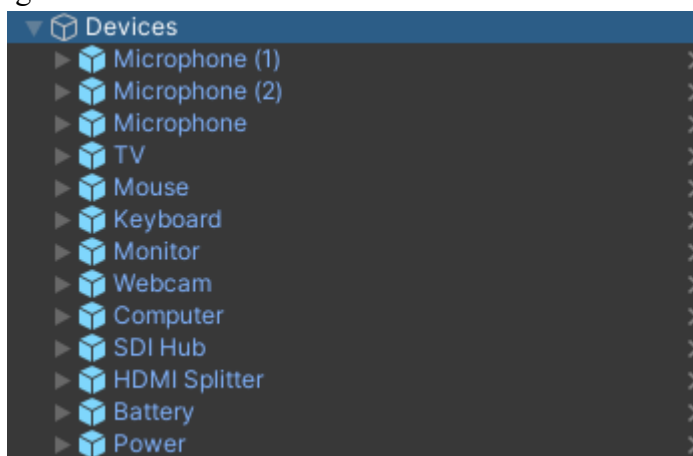
12 pav. XROrigin sudėtis

- **XR Interaction Manager** – UnityXR integracijos valdiklis, kontroliuojantis VR interaktyvumo aspektus.
- **Room** – virtualios erdvės sudėtinės dalys. Šis objektas apibrėžia tik statines ir neinteraktyvias erdvės dalis, tokias kaip sienos, durys, ventiliacijos angos ir t.t.

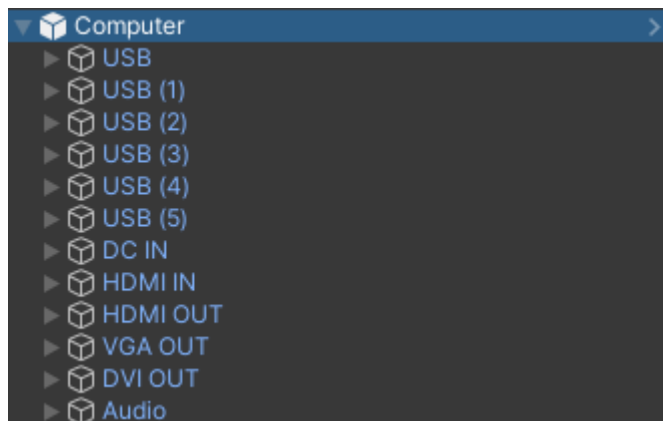


13 pav. Virtualios erdvės sudedamieji elementai

- **Furniture** – papildomi aplinkos elementai, toki kaip stalai, pelės kilimėlis, kėdė ir t.t.
- **Lights** – šviesos šaltiniai naudojami scenoje
- **Devices** – visi prietaisai naudojami scenarijuje. Kiekvienas prietaisas turi lizdus, kurie naudojami jų apjungimui.

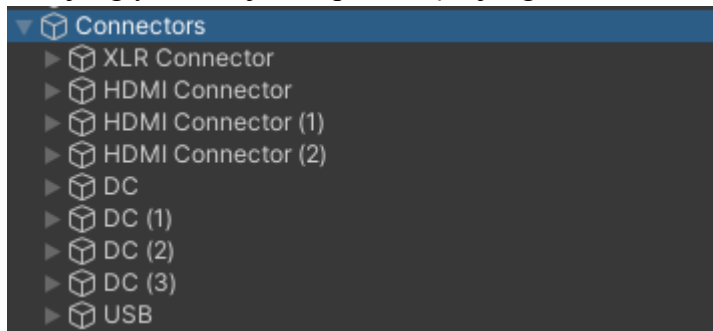


14 pav. Prietaisai naudojami scenarijuje



15 pav. Kiekvienas prietaisas turi atskiras jungtis

- **Connectors** – įvairios jungtys naudojamos prietaisų sujungimui.



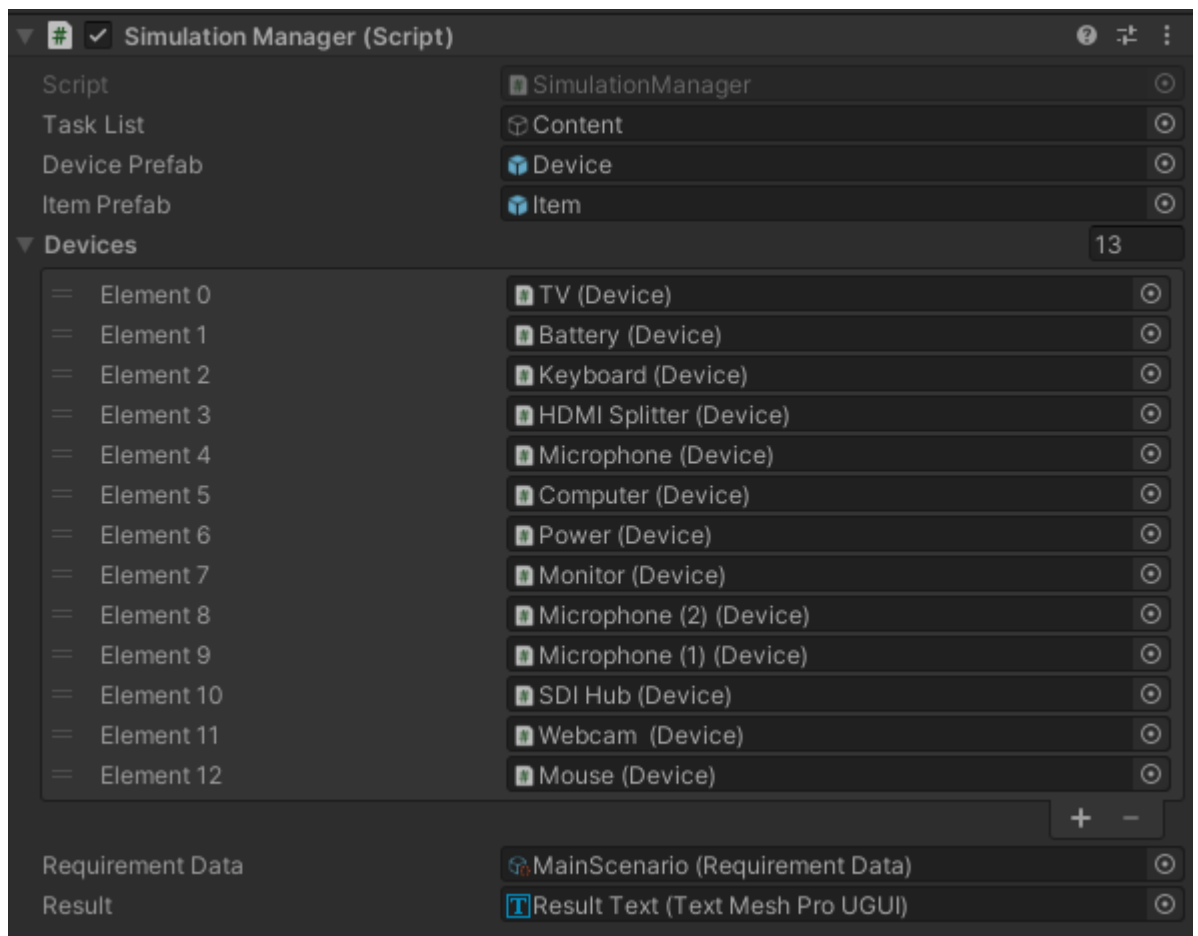
16 pav. Jungtys

- **SoundManager** – garso valdiklis turintis du garso šaltinius. Vienas naudojamas muzikai, kitas įvairiems garso efektams.



17 pav. Garso valdiklis ir garso šaltiniai

- **SimulationManager** – scenarijaus valdiklis. Turi prieigą prie prietaisų informacijos ir saugo scenarijaus reikalavimus. Įvykdžius scenarijų, tikrina jo įvykdymą.



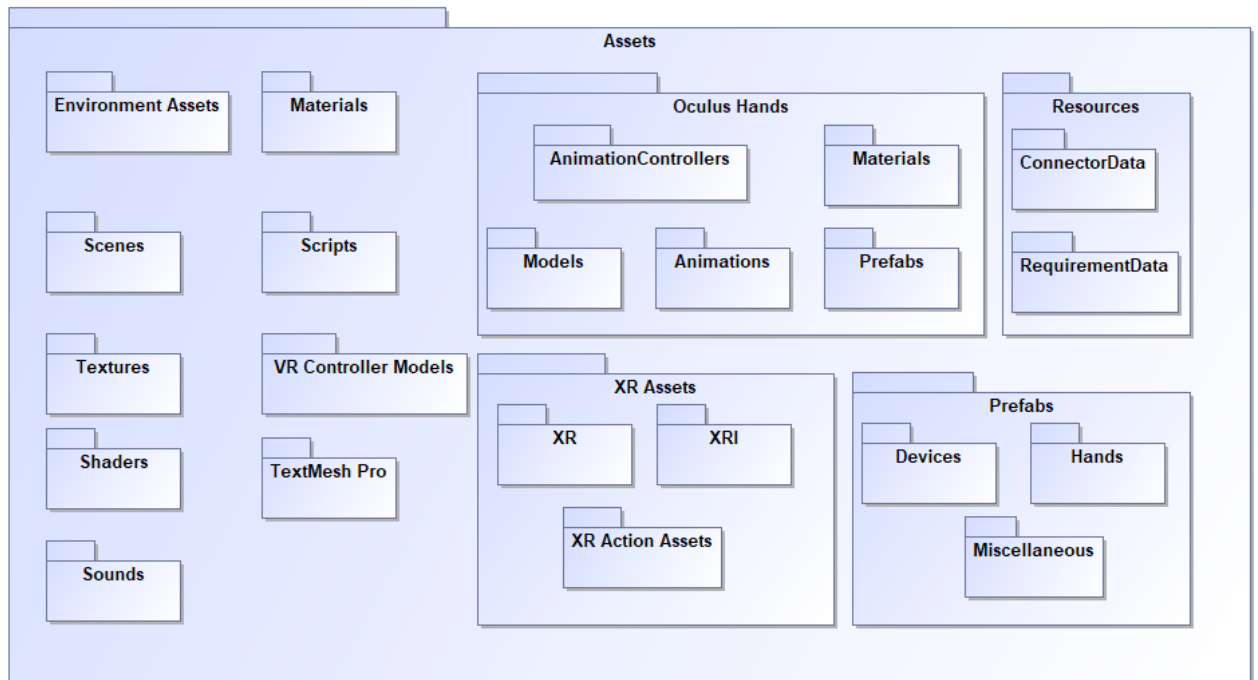
18 pav. Scenarijaus valdiklis

- **Canvas, EventSystem** – Canvas yra Unity elementas talpinantis vartotojo sąsajos objektus. Šiuo atveju naudojamas užduočių lentelė. EventSystem yra neatsiejama Canvas dalis, leidžianti vartotojui sąveikauti su Canvas objektais.

2.3.6. Komponentai

Sistemos komponentai:

- **Assets** – pagrindinis sistemos aplankas
 - **EnvironmentAssets** – VR aplinkos detalėms naudojami resursai;
 - **Materials** – Unity medžiagos naudojamos modifikuoti objektų vaizdines savybes;
 - **Oculus Hands** – sistemoje naudojamų rankų modelių komponentai
 - **AnimationControllers** – rankoms animuoti naudojami Unity animacijų valdikliai;
 - **Animations** – VR rankų animacijos;
 - **Materials** – VR rankų medžiagos;
 - **Models** – VR rankų modeliai;
 - **Prefabs** – VR rankų Unity objektai, apjungiantys modelius ir animacijas;
 - **Prefabs** – Unity objektai susidedantys iš modelių ir juos valdančio kodo;
 - **Devices** – prietaisų Unity objektai;
 - **Hands** – rankų Unity objektai;
 - **Miscellaneous** – įvairūs neklasifikuoti Unity objektai;
 - **Resources** – jungčių ir prietaisų apjungimo informacijos ruošiniai;
 - **ConnectorData** – prietaisų jungčių tipų ruošiniai;
 - **RequirementData** – apjungimo scenarijų reikalavimų ruošiniai;
 - **Scenes** – Unity scenos;
 - **Scripts** – sistemos kodas;
 - **Connector** – kodas atsakingas už sujungimų valdymą;
 - **ConnectorData** – kodas leidžiantis kurti naujus jungčių tipus;
 - **Device** – kodas atsakingas už prietaisų identifikaciją ir sujungimų saugojimą;
 - **DrawerSetup** – kodas leidžiantis vartotojui sąveikauti su stalčiais;
 - **PhysicsMover** – papildomas kodas naudojamas judinti objektus su Rigidbody komponentu;
 - **HandPresence** – kodas valdantis vartotojo rankas, jų animacijas;
 - **RequirementData** – kodas leidžiantis kurti sujungimo scenarijų ruošinius;
 - **SimulationManager** – kodas atsakingas už scenarijaus užduočių sudarymą, sujungimo validaciją ir scenarijaus valdymą;
 - **Socket** – kodas valdantis jungčių sujungimą su prietaisais;
 - **SoundManager** – kodas atsakingas už garsų paleidimą scenarijaus metu;
 - **TeleportationManager** – kodas atsakingas už teleportacijos užklausų validavimą ir įvykdymą;
 - **Wire** – kodas vykdomas jungčių laido vizualizaciją;
 - **Shaders** – nestandartinės medžiagos;
 - **Sounds** – sistemos garsai;
 - **TextMesh Pro** – Unity paketas
 - **Textures** – tekstūros naudojamos sistemoje;
 - **VR Controller Models** – VR pultelių modeliai;
 - **XR Assets** – Unity XR resursai;
 - **XR** – Unity XR virtualios realybės pagrindinis paketas;
 - **XR Action Assets** – Unity XR virtualios realybės valdiklių įvedimo ruošiniai;
 - **XRI** – Unity XR virtualios realybės paketas saugantis objektų sąveikos ruošinius;



19 pav. Paketų diagrama atvaizduojanti projekto struktūrą

2.3.7. Duomenų kontrolė

Duomenų kontrolė sistemoje vyksta vykdant prietaisų sujungimą ir scenarijaus įvykdymo patvirtinimą. Vykstant jungčių sujungimui, jungties duomenys yra nusiunčiami lizdui ir vykdoma validacija. Kiekviena jungtis gali būti skirtingo tipo (HDMI, USB, Audio ir t.t.) bei turėti skirtingas charakteristikas (jungties versija, antgalio tipas, t.t.). Sujungimo metu ši informacija yra palyginama su lizdo informacija, pvz.: Bandant sujungti HDMI jungtį, su USB lizdu, validacija neleis sujungti prietaisų ir jungties duomenys nebus išsaugoti. Šiai validacijai įgyvendinti yra naudojami pamodifikuoti UnityXR integracijos metodai. Sujungus du prietaisus, vykdoma papildoma duomenų validacija, kurios metu nustatoma ar jungtis tarp dviejų prietaisų yra aktyvi. Pvz.: Sujungus du prietaisus naudojant HDMI jungtį, bet su abejais prietaisais naudojant HDMI OUT tipo lizdus, jungčių sujungimas pavyks, bet tarp prietaisų sujungimas nėra veiksmingas. Tokiu atveju tarp prietaisų nevyks duomenų mainai ir jų sujungimo informacija nebus siunčiama simuliacijos valdymo klasei, vykstant scenarijaus įgyvendinimo tikrinimą.

```

0 references
public override bool CanHover(IXRHoverInteractable interactable)
{
    return base.CanHover(interactable) && MatchHover(interactable);
}

0 references
public override bool CanSelect(IXRSelectInteractable interactable)
{
    return base.CanSelect(interactable) && MatchSelect(interactable);
}

1 reference
private bool MatchHover(IXRHoverInteractable interactable)
{
    if (interactable.transform.GetComponent<Connector>() != null)
    {
        return interactable.transform.GetComponent<Connector>().CheckType(connectorData);
    }
    else
    {
        return false;
    }
}

1 reference
private bool MatchSelect(IXRSelectInteractable interactable)
{
    if (interactable.transform.GetComponent<Connector>() != null)
    {
        return interactable.transform.GetComponent<Connector>().CheckType(connectorData);
    }
    else
    {
        return false;
    }
}

```

20 pav. Vykdant jungčių sujungimą yra naudojami modifikuoti UnityXR metodai

```

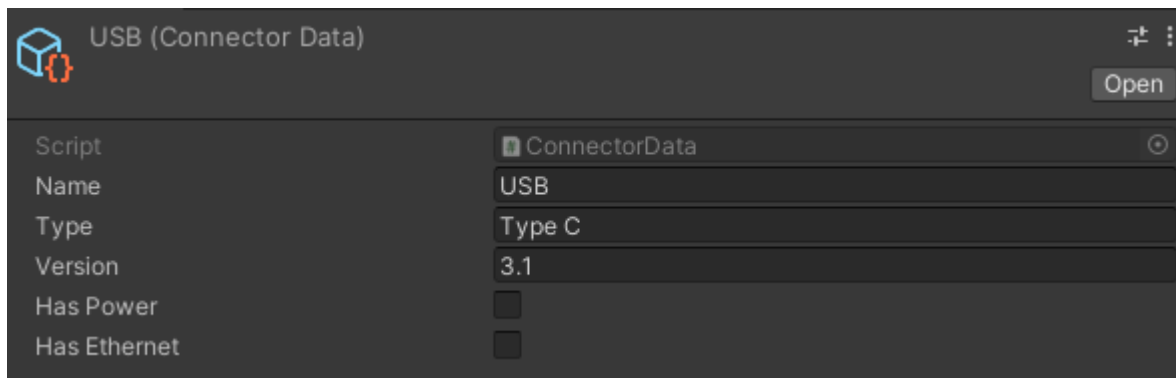
if (thisEnd.socketType == SocketType.Both && (otherEnd.socketType == SocketType.Both || otherEnd.socketType == SocketType.In || otherEnd.socketType == SocketType.Out)
|| (thisEnd.socketType == SocketType.Both || thisEnd.socketType == SocketType.In || thisEnd.socketType == SocketType.Out) && otherEnd.socketType == SocketType.Both
|| thisEnd.socketType == SocketType.In && otherEnd.socketType == SocketType.Out
|| thisEnd.socketType == SocketType.Out && otherEnd.socketType == SocketType.In)
{
    currentConnection = new Connection(otherEnd.connectedToDevice, connectorData, otherEnd.socketType);
    currentOtherEndConnection = new Connection(device, connectorData, socketType);
    device.connections.Add(currentConnection);
    otherEnd.connectedToDevice.connections.Add(currentOtherEndConnection);
}

```

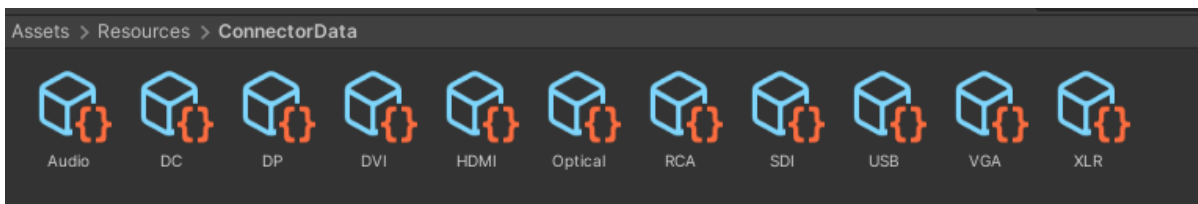
21 pav. Sujungus prietaisus vykdoma papildoma validacija

Vykstant scenarijaus įgyvendinimo tikrinimui, visų scenoje esančių prietaisų sujungimo informacija yra siunčiama simuliacijos valdymo klasei. Ši klasė saugo scenarijaus įgyvendinimo kriterijus, su kuriais nustatoma ar prietaisų sujungimai yra pagrįsti.

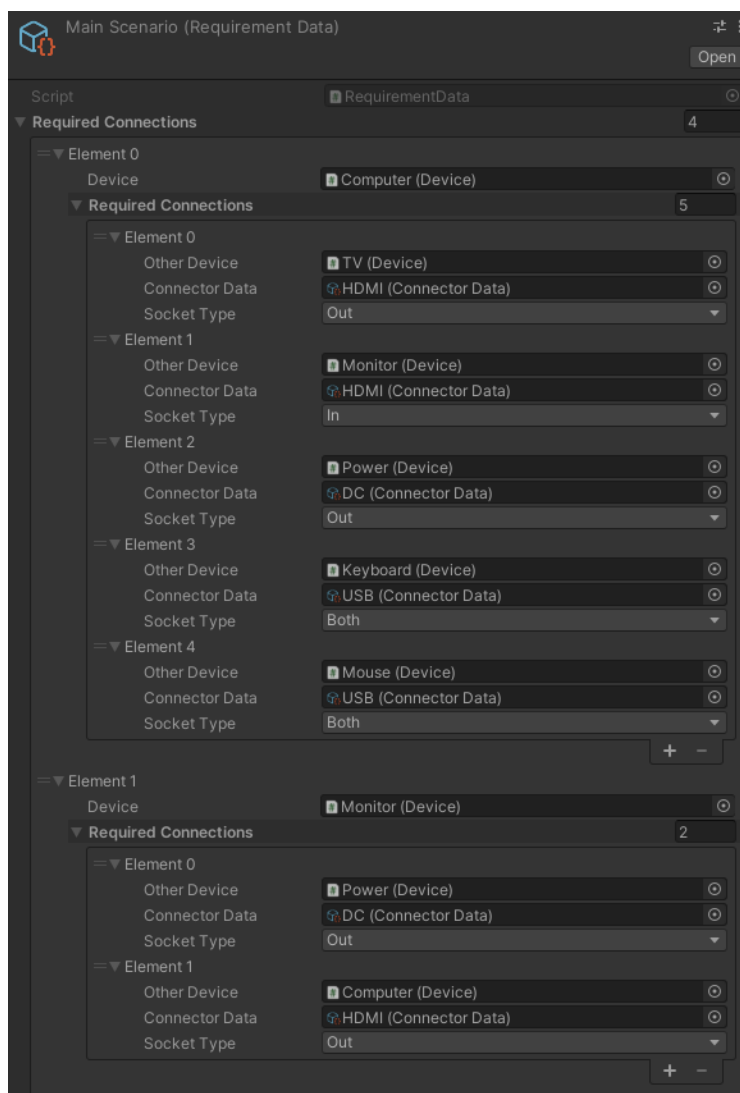
Pagrindinė jungčių ir lizdų informacija, bei scenarijaus įgyvendinimo reikalavimai apibrėžiami naudojant Unity ScriptableObject objektus. Šie objektai veikia kaip duomenų ruošiniai, kurie gali būti sukurti pagal nustatytas specifikacijas. ScriptableObjects palengvina jungčių identifikavimą ir sujungimų validaciją, leidžia greitai pakeisti jungčių tipų informaciją arba sukurti naujus tipus. Šie objektai taip pat leidžia greitai modifikuoti ir kurti naujus sujungimo scenarijus.



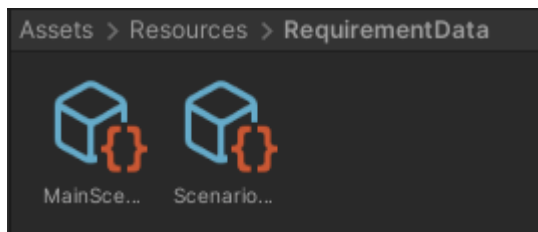
22 pav. ScriptableObject naudojamas apibrėžti jungties parametrus



23 pav. Kiekvienam jungties tipui apibrėžti naudojamas skirtingas ScriptableObject



24 pav. ScriptableObject naudojami scenarijaus reikalavimų apibrėžimui



25 pav. Skirtingi ScriptableObjects gali apibrėžti skirtingus scenarijus

2.4. Testavimo eiga ir rezultatai

Sistemai testuoti naudotas testavimas scenarijais. Sudaryti scenarijai tikrinti skirtingus sistemos aspektus. Testavimo scenarijaus struktūra:

- **Aprašymas** – aprašomas testuojamas scenarijus;
- **Testo atvejai** – galimi testo rezultatai;

2.4.1. Teleportacijos funkcionalumas

Aprašymas: Testuojamas teleportacijos funkcionalumas virtualioje realybėje. Testuojama teleportacija į galimas zonas, bei sistemos veikimas, vartotojui bandant teleportuotis į draudžiamas zonas, tokias kaip sienos, objektai, durys ir t.t.

6 lentelė Teleportacijos funkcionalumo atvejų lentelė

Atvejis	Aprašymas	Žingsniai	Tikėtinas rezultatas	Gautas rezultatas
Teleportacija įvyksta	Vartotojas bando nusiteleportuoti į galimą zoną	1. Valdiklio lazdelė nukreipama į priekį	Iš pultelio išsauna teleportacijos vietos indikatorius	Iš pultelio išsauna teleportacijos vietos indikatorius
		2. Valdiklis nukreipiamas į tinkamą teleportacijai paviršių	Indikatorius nusidažo balta spalva	Indikatorius nusidažo balta spalva
		3. Valdiklio lazdelė atleidžiama	Vartotojas atsiranda norimoje vietoje	Vartotojas atsiranda norimoje vietoje
Teleportacija neįvyksta	Vartotojas bando nusiteleportuoti į draudžiamą zoną	1. Valdiklio lazdelė nukreipama į priekį	Iš pultelio išsauna teleportacijos vietos indikatorius	Iš pultelio išsauna teleportacijos vietos indikatorius
		2. Valdiklis nukreipiamas į netinkamą teleportacijai paviršių	Indikatorius nusidažo raudona spalva	Indikatorius nusidažo raudona spalva
		3. Valdiklio lazdelė atleidžiama	Niekas neįvyksta	Vartotojas nusiteleportuoja

Vykdam teleportacijos funkcionalumo testavimą scenarijais, rasta klaida, dėl kurios bandantis nusiteleportuoti į draudžiamą zoną, teleportacija įvyktų. Klaida buvo ištaisyta.

2.4.2. Objektų manipuliavimo funkcionalumas

Aprašymas: Testuojama galimybė manipuluoti objektus virtualioje realybėje. Vartotojas bando paimti ir judinti šalia esančius objektus. Taip pat testuojamas tolumo paėmimo funkcionalumas, leidžiantis vartotojui paimti tolimiau esančius objektus nereikalaujant daug pastangų.

7 lentelė Objektų manipuliavimo funkcionalumo atvejų lentelė

Atvejis	Aprašymas	Žingsniai	Tikėtinas rezultatas	Gautas rezultatas
Vartotojas paima ir manipuliuoja šalia esantį objektą	Vartotojas bando paimti objektą, naudodamas savo virtualią ranką ir jį judinti virtualioje realybėje.	1. Vartotojas užveda ranką ant objekto	Niekas neįvyksta	Niekas neįvyksta
		2. Vartotojas sugniaužia ranką	Objektas atsiranda vartotojo rankoje	Objektas atsiranda vartotojo rankoje
		3. Vartotojas judina ranką	Objektas juda su vartotojo ranka	Objektas juda su vartotojo ranka
		4. Vartotojas atleidžia ranką	Objektas iškrenta iš vartotojo rankos	Objektas iškrenta iš vartotojo rankos
Vartotojas paima ir manipuliuoja toli esantį objektą	Vartotojas bando paimti objektą, naudodamas tolimų objektų paėmimo funkciją ir jį judinti virtualioje realybėje.	1. Vartotojas nutaiko ranką į toliau esantį objektą	Iš rankos išsauna baltas paėmimo indikatorius	Iš rankos išsauna baltas paėmimo indikatorius
		2. Vartotojas sugniaužia ranką	Objektas atsiranda vartotojo rankoje	Objektas atsiranda vartotojo rankoje
		3. Vartotojas judina ranką	Objektas juda su vartotojo ranka	Objektas juda su vartotojo ranka
		4. Vartotojas atleidžia ranką	Objektas iškrenta iš vartotojo rankos	Objektas iškrenta iš vartotojo rankos

Įvykdžius objektų manipuliavimo scenarijus nerasta klaidų.

2.4.3. Prietaisų sujungimo funkcionalumas

Aprašymas: Testuojamas prietaisų sujungimo funkcionalumas virtualioje realybėje. Prietaisai sujungiami tarpusavyje, tikrinamas jungčių atjungimas ir kiti aspektai.

8 lentelė Prietaisų sujungimo funkcionalumo atvejų lentelė

Atvejis	Aprašymas	Žingsniai	Tikėtinas rezultatas	Gautas rezultatas
Sujungiami prietaisai	Vartotojas bando sujungti du prietaisus naudojant jungtis	1. Vartotojas paima jungties galą ir sujungia jį su laisvu prietaiso lizdu	Jungtis susijungia su lizdu ir pasigirsta sujungimo garsas	Jungtis susijungia su lizdu ir pasigirsta sujungimo garsas
		2. Vartotojas paima kitą jungties galą ir sujungia jį su laisvu kito prietaiso lizdu	Jungtis susijungia su lizdu, pasigirsta sujungimo garsas, vartotojas mato, jog tarp dviejų prietaisų yra ryšys	Jungtis susijungia su lizdu, pasigirsta sujungimo garsas, vartotojas mato, jog tarp dviejų prietaisų yra ryšys
Atjungiamo jungtis	Vartotojas bando ištraukti jungtį iš prietaiso lizdo	1. Vartotojas užveda ranką ant jungties įkištos į prietaiso lizdą	Niekas neįvyksta	Niekas neįvyksta
		2. Vartotojas sugniaužia ranką ir ištraukia jungtį	Jungtis išsitraukia iš lizdo ir juda su vartotojo ranka	Jungtis išsitraukia iš lizdo ir juda su vartotojo ranka
		3. Vartotojas atleidžia ranką	Jungtis iškrenta ir vartotojo rankos, ji nėra sujungta su prietaisu	Jungtis iškrenta ir vartotojo rankos, ji nėra sujungta su prietaisu
Nepavyksta įkišti jungties į lizdą	Vartotojas bando įkišti jungtį į užimtą arba netinkamo tipo lizdą	1. Vartotojas paima jungties galą	Jungtis atsiranda vartotojo rankoje	Jungtis atsiranda vartotojo rankoje
		2. Vartotojas bando įkišti jungtį į lizdą	Niekas neįvyksta	Niekas neįvyksta

Įvykdžius prietaisų sujungimo scenarijus nerasta klaidų.

2.4.4. Scenarijaus valdymo funkcionalumas

Aprašymas: Testuojamas prietaisų sujungimo funkcionalumas virtualioje realybėje. Prietaisai sujungiami tarpusavyje, tikrinamas jungčių atjungimas ir kiti aspektai.

9 lentelė Scenarijaus valdymo funkcionalumo atvejų lentelė

Atvejis	Aprašymas	Žingsniai	Tikėtinas rezultatas	Gautas rezultatas
Scenarijus įvykdytas sėkmingai	Vartotojas teisingai įvykdęs scenarijų vykdo validaciją	1. Vartotojas nueina prie scenarijaus reikalavimų ekrano	Vartotojas mato scenarijaus reikalavimus	Vartotojas mato scenarijaus reikalavimus
		2. Vartotojas sujungia prietaisus pagal reikalavimus	Nieko neįvyksta	Nieko neįvyksta
		3. Vartotojas nuspaudžia tikrinimo mygtuką esantį reikalavimų lentelėje	Lentelėje atsiranda sėkmingo scenarijaus įvykdymo tekstas	Lentelėje atsiranda sėkmingo scenarijaus įvykdymo tekstas
Scenarijus įvykdytas nesėkmingai	Vartotojas vykdo scenarijaus validaciją, neįvykdęs arba nepilnai įvykęs prietaisų sujungimą	1. Vartotojas neteisingai sujungia prietaisus	Niekas neįvyksta	Niekas neįvyksta
		2. Vartotojas nuspaudžia tikrinimo mygtuką esantį reikalavimų lentelėje	Lentelėje atsiranda nesėkmingo scenarijaus įvykdymo tekstas	Lentelėje atsiranda nesėkmingo scenarijaus įvykdymo tekstas
Scenarijus pradedamas iš naujo	Vartotojas bando pradėti scenarijų iš naujo	1. Vartotojas įvykdo prietaisų sujungimus ir kitus pakeitimus virtualioje erdvėje	Vartotojas mato pakeitimus virtualioje erdvėje	Vartotojas mato pakeitimus virtualioje erdvėje
		2. Vartotojas nueina prie scenarijaus reikalavimų ekrano	Niekas neįvyksta	Niekas neįvyksta
		3. Vartotojas nuspaudžia scenarijaus perkrovimo mygtuką	Vartotojas atsiranda pradinėje pozicijoje, visi objektai yra savo pradinėse pozicijose, prietaisai nėra sujungti	Vartotojas atsiranda pradinėje pozicijoje, visi objektai yra savo pradinėse pozicijose, prietaisai nėra sujungti

Įvykdžius scenarijaus valdymo funkcionalumo scenarijų nerasta klaidų.

2.5. Dokumentacija

2.6. Sistemos reikalavimai

Sistemai paleisti reikalingi Oculus Quest arba Oculus Quest 2 virtualios realybės akiniai, bei jų valdikliai. Norint sistemą paleisti per kompiuterį, papildomai reikalingas Oculus Link kabelis ir Oculus programinė įranga. Minimalūs reikalavimai naudoti Oculus Link yra Intel i5-4590 arba AMD Ryzen 5 1500X procesorius, NVIDIA GeForce GTX 970 arba AMD 400 Series vaizdo plokštė, bent 8GB RAM, Windows 10 operacinė sistema ir viena USB jungtis. Rekomenduojama USB jungties 3.0 versija maksimaliam našumui.

10 lentelė Minimalūs Oculus Link reikalavimai

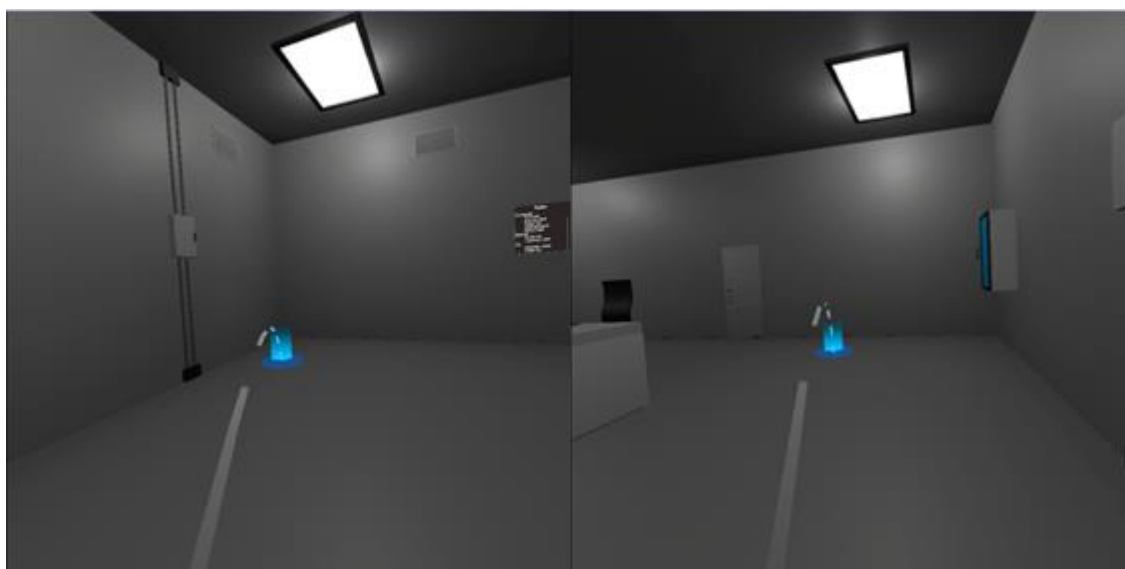
Komponentas	Rekomenduojamos minimalios specifikacijos
Procesorius	Intel i5-4590 / AMD Ryzen 5 1500X
Vaizdo plokštė	NVIDIA GeForce GTX 970 / AMD 400 Series
Atmintis	8 GB+ RAM
Operacinė sistema	Windows 10
USB jungtis	1x USB jungtis (rekomenduojama 3.0)

2.7. Sistemos naudojimas

Sistema paleidžiama ją pasirenkant Oculus programų kataloge ir nuspaudžiant atidarymo mygtuką. Programai atsidarius, vartotojas atsiranda uždarame kambaryje su įvairiais prietaisais:

- Kompiuteris
- Monitorius
- Televizorius
- HDMI signalo daliklis
- WEB kamera
- Klaviatūra
- Kompiuterio pelė
- 3 mikrofoni
- SDI šakotuv

Vartotojas gali judėti kambaryje naudodamas teleportacijos funkciją. Teleportacija vykdoma nuspaudžius kairio arba dešinio valdiklio lazdele į šiaurės poziciją. Įvykdžius šį veiksmą virtualioje erdvėje atsiranda teleportacijos indikatorius, kurį vartotojas gali naudoti vietos pasirinkimui. Pasirinkus galimą teleportacijos vietą, indikatorius nusidažys balta spalva. Norėdamas patvirtinti teleportaciją, vartotojas turi atleisti valdiklio lazdele.

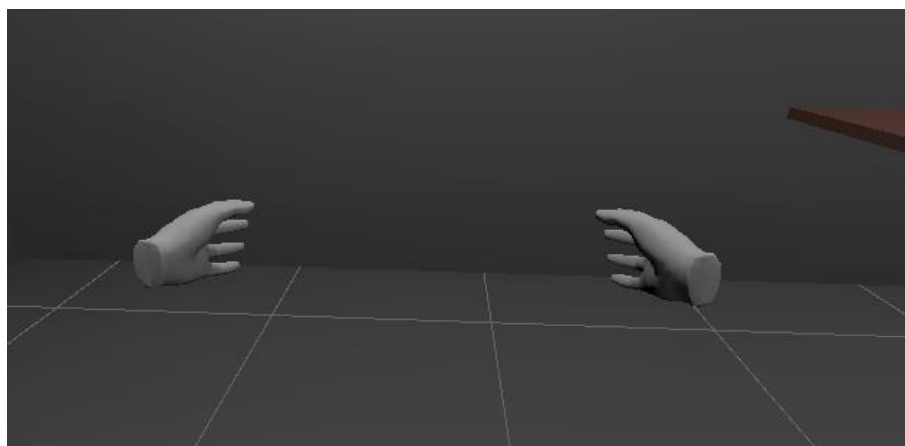


26 pav. Teleportacijos vietos indikatorius leidžia vartotojui matyti kur jis atsiranda



27 pav. Indikatorius nusidažo raudonai, jeigu vartotojui negalima teleportuotis į tam tikrą vietą

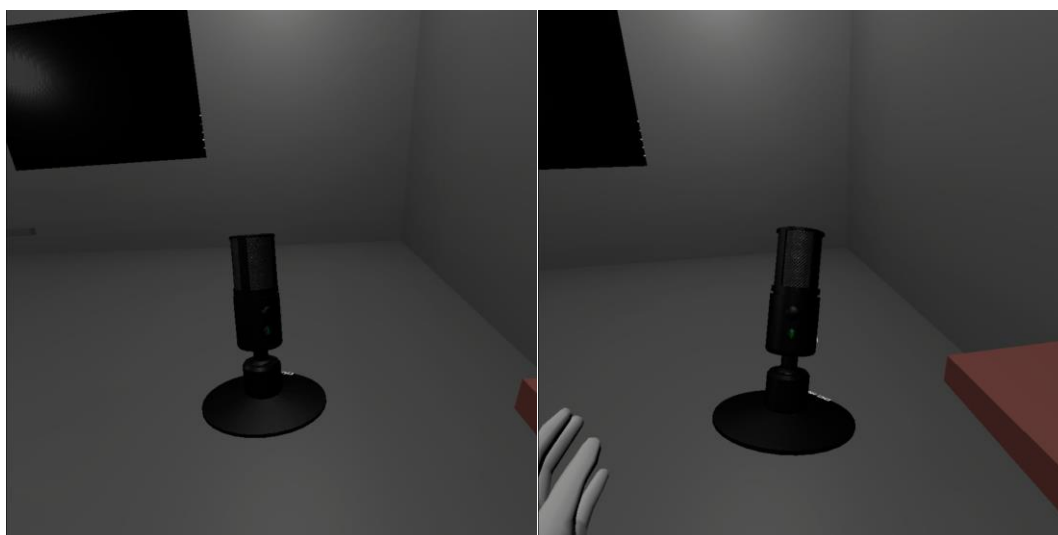
Vartotojas turi dvi animuotas rankas, kuriomis gali manipuluoti objektais. Vartotojas gali paimti ir judinti įvairius objektus, prietaisus bei jungtis užvesdamas ant jų ranką ir nuspausdamas šoninį valdiklio mygtuką. Vartotojas gali įkišti jungtis į prietaisų lizdus, jas paimdamas ir nuvesdamas prie lizdo. Vartotojas gali atjungti prietaisus paimdamas įkištas jungtis ir atitraukdamas jas nuo prietaiso.



28 pav. Vartotojo rankos virtualioje erdvėje



29 pav. Vartotojo rankos yra animuotos. Sugniaužus kumštį, tai atsispindi virtualioje erdvėje



30 pav. Vartotojas rankose gali laikyti įvairius objektus

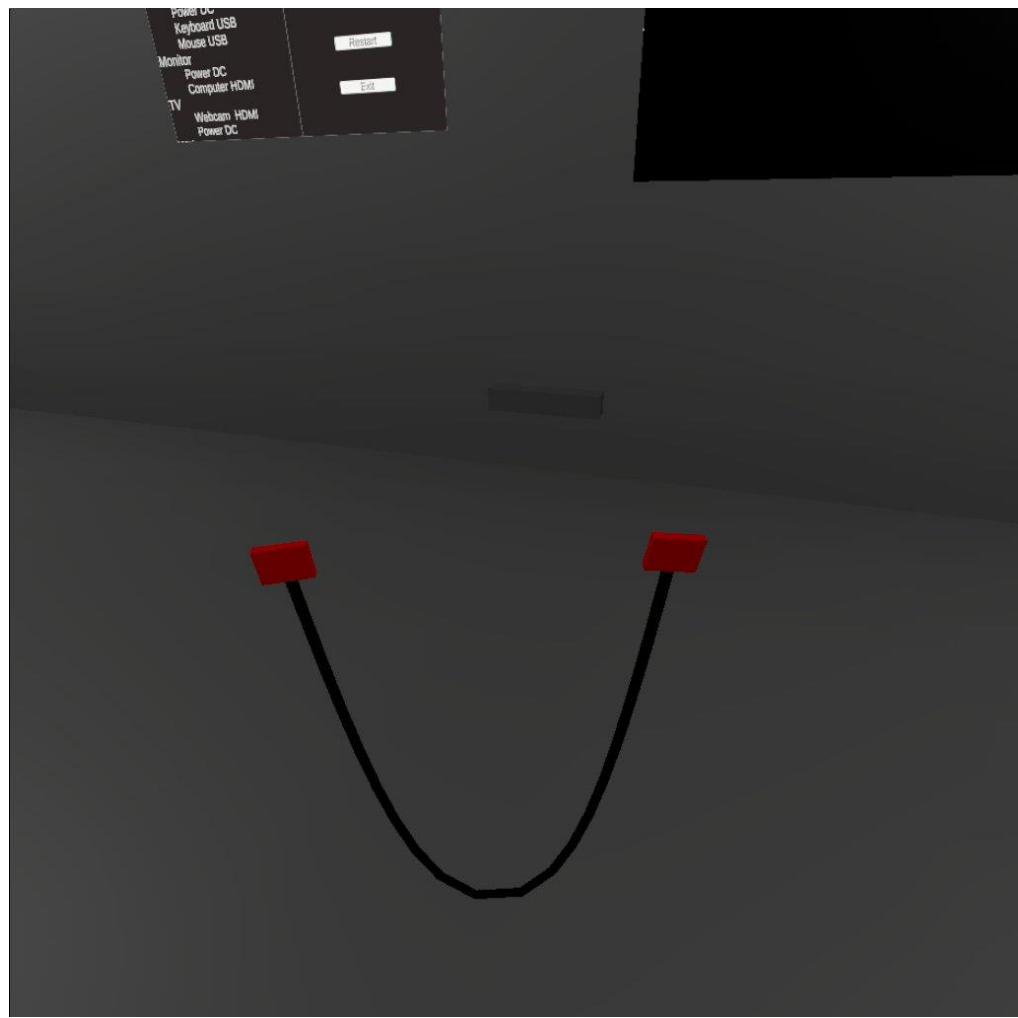
31 pav. Objektą laikanti ranka tampa nematoma



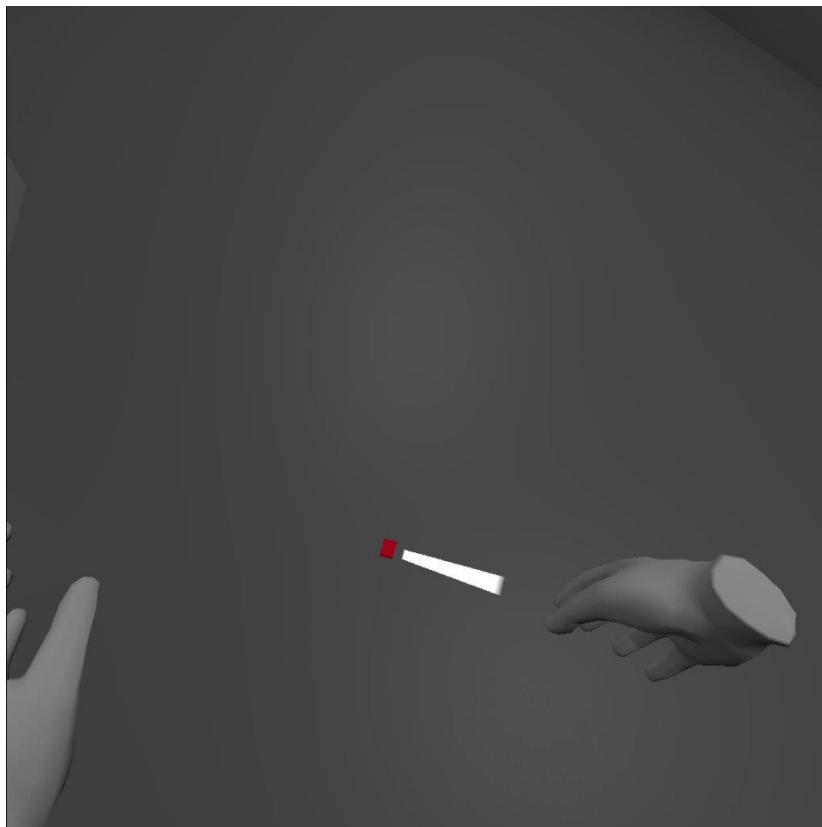
32 pav. Objektai gali vienas su kitu sąveikauti virtualioje erdvėje



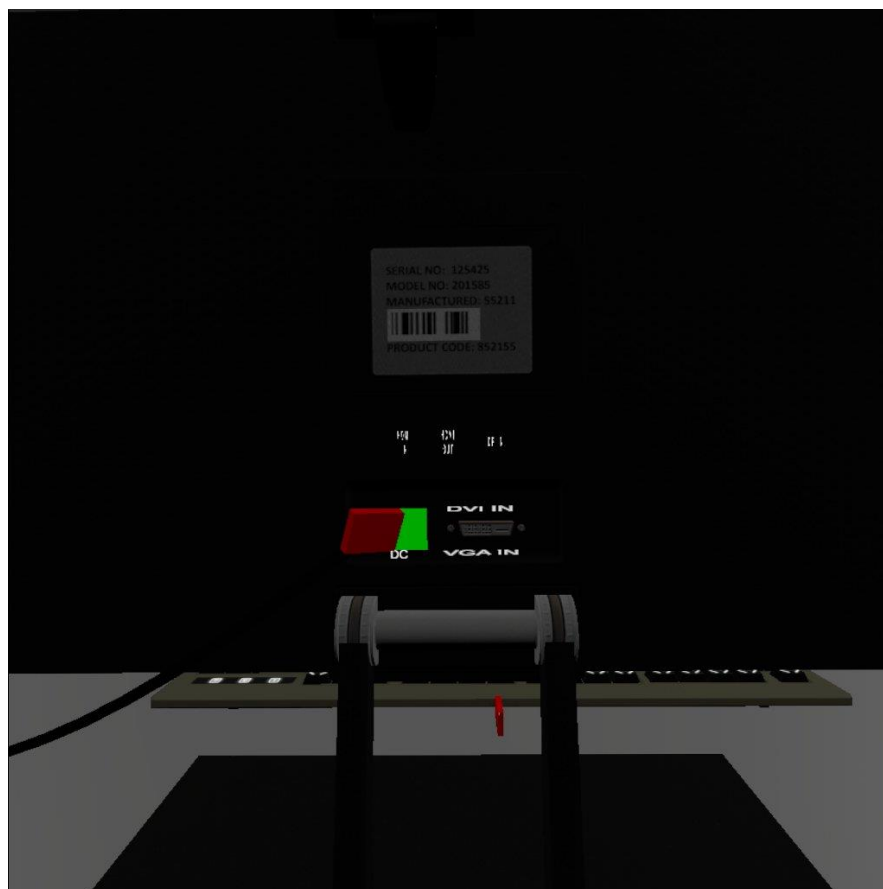
33 pav. Virtualiame kambaryje esantys įvairūs prietaisai



34 pav. Jungtys skirtos apjungti prietaisus. Jos susietos laidu

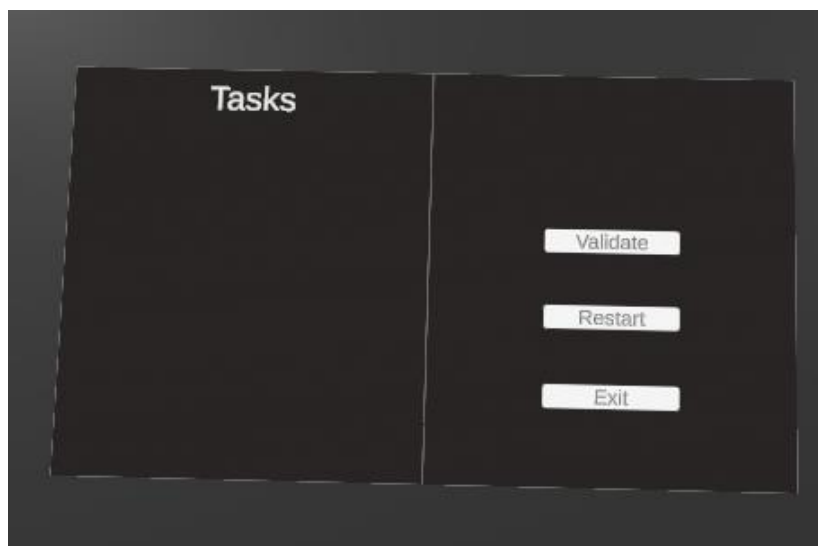


35 pav. Iš vartotojo rankos iššauna spindulys, leidžiantis paimti objektus iš toliau

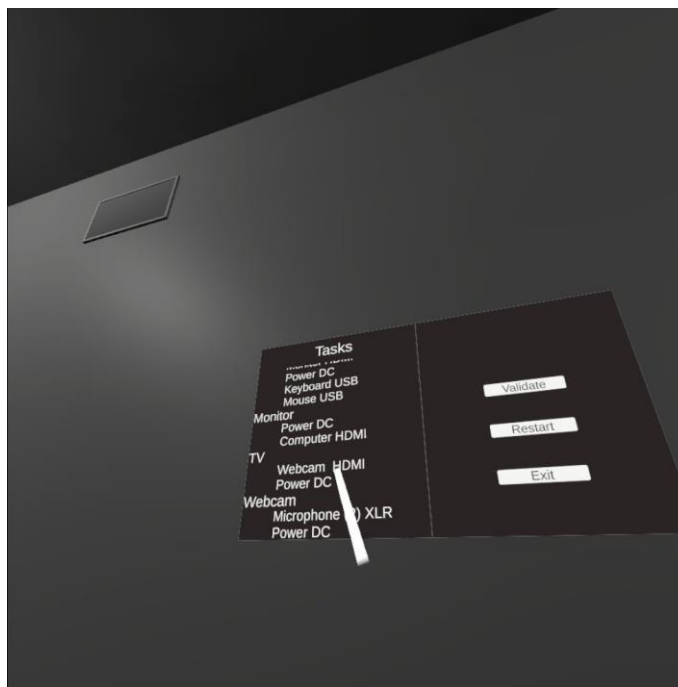


36 pav. Vartotojui bandant sujungti prietaisus atsiranda indikatorius, leidžiantis jam žinoti, jog sujungimas galimas

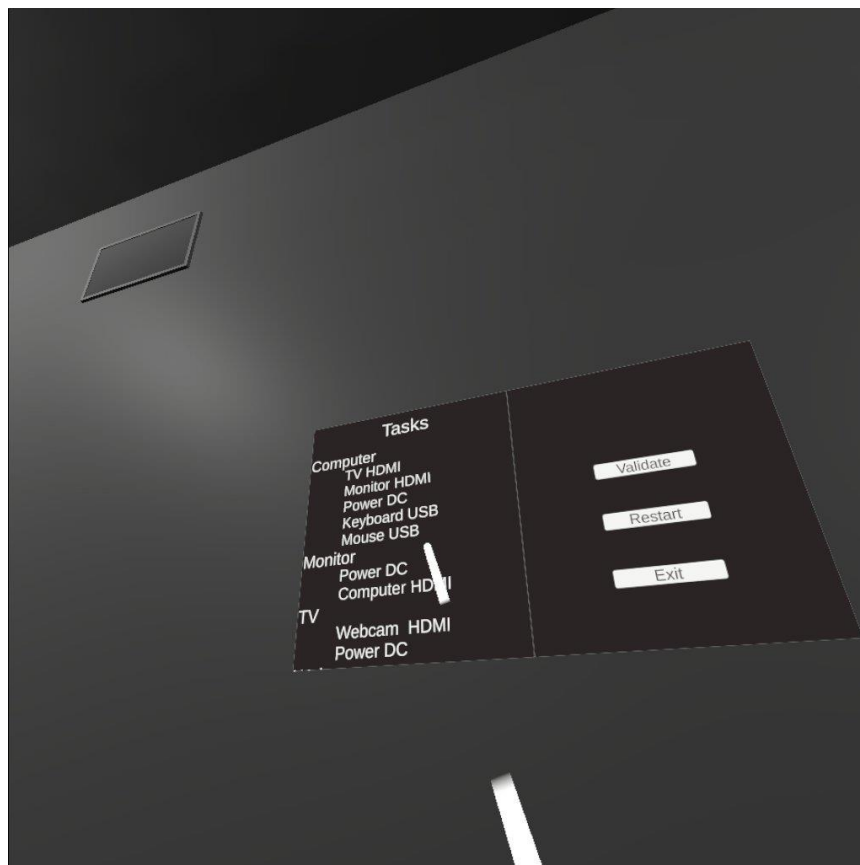
Vartotojo tikslas yra sujungti visus kambaryje esančius objektus pagal tam tikrą scenarijų. Scenarijaus reikalavimai gali būti matomi ant kambario sienos, esančios priešais vartotoją, prasidėjus scenarijui. Šalia reikalavimų yra trys mygtukai, Validate, Restart ir Exit. Validate leidžia vartotojui patikrinti ar įvykdyti visi scenarijaus reikalavimai. Reset mygtukas leidžia vartotojui perkrauti scenarijų iš naujo. Exit mygtukas uždaro programą. Navigacija lange vykdoma naudojant spindulį, kuris atsiranda vartotojui nusitaikius į langą ranka.



37 pav. Scenarijaus valdymo ir reikalavimų ekranas



38 pav. Vartotojui nusitaikius į užduočių langą, iš jo rankos išsauna spindulys.



39 pav. Naudodamas spindulį vartotojas gali naviguoti po užduočių langą

3. Refleksija

Praktikos pradžia buvo gana sudėtinga. Nors prieš tai buvau dirbęs prie programų naudojant Unity3D, niekuomet nedirbau prie virtualios realybės aplikacijų. Daug dėmesio skyriau įvairių virtualios realybės paketų bei integracijų tyrimui, įvairių virtualios realybės įrankių ieškojimui. Sukūrus pradinę sistemos scenarijų reikėjo jį ištestuoti naudojant virtualios realybės akinius, bet šioje vietoje kilo keblumų. Nors prieš tai buvau naudojęs Oculus Quest įrangą pramoginiams tikslais, niekada nebuvo jos naudojęs aplikacijų kūrimui. Pirmoji kliūtis buvo prietaiso susiejimas su kompiuteriu. Teko atsisiųsti keletą programų bei prietaiso tvarkyklių, tam, kad kompiuteris atpažintų Oculus sistemą. Tuomet teko susikurti kūrėjo paskyrą Oculus svetainėje, bei aktyvuoti kūrėjo režimą pačiuose virtualios realybės akinuose. Galiausiai teko atsisiųsti įvairius paketus Unity paketus, kurie leistų aplikaciją transliuoti į akinius.

Įveikus pradines kliūtis, sistemos kūrimas nebuvo ypatingai sudėtingas dirbant prie pagrindinių funkcijų. UnityXR paketas suteikia valdiklių įvesčių profilį Oculus Quest platformai, tai leido lengvai iš Oculus Quest išgauti mygtukų paspaudimus, vartotojo vietos informaciją, valdiklius ir jų būseną. Vartotojo interaktyvavimo su objektais funkcionalumas buvo lengviausia sistemos kūrimo dalis, bet pradėjus dirbti prie teleportacijos ir judėjimo funkcionalumo kilo keblumų su objektų atpažinimu ir jų judinimu teleportacijos metu, vietų į kurias galima teleportuoti nustatymu, įvairiais indikatoriais kurie suteiktų vartotojui informaciją. Šio etapo metu taip pat pastebėjau, jog mano VR įvesčių sekimo modelis yra senesnės kartos ir nėra ypatingai lankstus. Teko atlikti įvesčių sistemos keitimą ir kodo pertvarkymą.

Dirbant prie įrenginių ir jungčių valdymo progresas judėjo žymiai greičiau, kadangi buvau labiau pripratęs prie UnityXR ypatybių. Bet tai nereiškia, jog nebuvo jokių keblumų. Pirmiausia reikėjo sugalvoti, koku būdu turėtų būti apibūdžta jungčių informacija ir kaip ta informacija dalintųsi objektai. Vartotojas turėjo galimybę paaimti ir judinti objektus, bet jie neturėjo jokių tipų ir požymių. Jungčių apibūdinimas turėjo būti lankstus. Per laiką atsiranda nauji jungčių tipai, naujos versijos. Atradus Unity ScriptableObject, jis tapo aiškiu pasirinkimu. Objektai su pasirinktiniais laukais, kurios galima sukurti ir patalpinti Unity aplanke leido tvarkyti jungčių tipus ir informaciją visiškai nereikalaujant sąveikos su scenoje esančiais objektais. Prieš tai niekuomet nebuvo naudojęs ScriptableObject objektų savo programose ir jų atradimas atrakino daug galimybių su ateityje kuriamais Unity projektais.

Paskutiniai sistemos kūrimo žingsniai ėjosi gana sklandžiai. ScriptableObject buvo panaudotas scenarijaus reikalavimų kūrimui ir validacijai. Prieš tai sukurta jungčių validacija palengvino scenarijaus įvykdymo tikrinimą. Sistema nereikalavo daug garsų ir garsų integravimas į Unity projektą nėra sudėtingas. Prietaisų ir aplinkos sukūrimas nereikalavo programavimo, reikėjo tik išdėlioti juos scenoje. Sistema buvo ištestuota scenarijais.

Praktikos metu pavyko praplėsti ne tik savo Unity žinias, bet ir išmokti virtualių aplikacijų kūrimo. Daugelis VR kūrimo ypatybių neturėjo ryšio su mano praktinėmis žiniomis, teko pačiam ieškoti naujos informacijos, bet įvairių validacijų integravimas, pagrindinės Unity funkcijos rėmėsi mano teorinėmis žiniomis. Mano nuomone praktikos procesas vyko gana sklandžiai ir praktika buvo vertinga.

Išvados

Praktikos užduoties atlikimas buvo sėkmingas. Kūrimo metu kilo įvairių problemų, vieni etapai buvo sudėtingesni negu kiti, bet kliūtys buvo įveiktos, su problemom susidorota. Sunkiausi praktikos aspektai kilo iš to, jog prieš tai niekuomet nebuvo dirbta su VR įrenginiais, kuriant VR aplikaciją. Dirbant prie ateities projektų su nepažystamais įrenginiais būtų pravartu naudoti informaciją iš keleto skirtingų šaltinių. Tai leistų rasti daugiau kūrimo galimybių ir identifikuoti geriausią kelią sistemos kūrimui.

Literatūra

1. Kauno technologijos universitetas. (2022). [žiūrėta 2022-03-10]. Prieiga per internetą <https://ktu.edu/universitetas/>