

TARTU ÜLIKOOL

Arvutiteaduse instituut

Informaatika õppekava

Mait Lättekivi

Kaugjuhitavate mudelautode videoülekanne VR lahendus

Bakalaureusetöö (9 EAP)

Juhendajad: Raimond-Hendrik Tunnel, MSc

Rainer Paat, MSc

Tartu 2021

Kaugjuhitavate mudelautode videoülekanne VR lahendus

Lühikokkuvõte:

Käesoleva bakalaureusetöö käigus valmis koostöös ettevõtte RCSnail OÜ-ga kaugjuhitavate mudelautode virtuaalreaalsuse lahendus, mis võiks tulevikus saada ettevõtte üheks teenuseks. Loodud lahenduse abil saab sõita interneti kaudu RCSnaili mudelautodega, kasutades virtuaalreaalsuse peakomplekt. Töös võrreldakse ka valitud tehnoloogiaid olemasolevate alternatiividega. Lahenduse loomisel kasutati raamistiku A-Frame.

Võtmesõnad:

Virtuaalreaalsus, arvutigraafika, raadioteel juhitud mudelautod, A-Frame

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

VR Live Stream of Remote-Controlled Racing Cars

Abstract:

In cooperation with RCSnail OÜ, a virtual reality solution was created of remote-controlled racing cars, which will hopefully become one of the company's services in the future. Using the created solution one can drive remote control cars using a virtual reality headset. The thesis also compares technologies and explains why alternatives were not selected. Framework A-Frame was used in creating this thesis.

Keywords:

Virtual reality, computer graphics, remote controlled cars, A-Frame

CERCS: P170 Computer science, numerical analysis, systems, control

Sisukord

1. Sissejuhatus.....	5
2. Vajalikud tehnoloogiad.....	9
2.1 RCSnail'i olemasolev lahendus	9
2.2 Virtuaalreaalsuse peakomplekt Oculus Quest	10
2.3 A-Frame	11
2.4 Three.js.....	11
2.5 WebXR	11
2.6 Virtuaalreaalsuse silumine	13
2.7 Alternatiivsed tehnoloogiad	14
2.7.1 Unity ja Unreal Engine	14
2.7.2 Babylon.js	15
3. Lahendus	15
3.1 Lahenduse kasu ja eelised.....	15
3.2 Olemasolevad lahendused.....	16
3.3 Kaamerate kalibreerimine.....	17
3.4 Lahendus A-Frame'is.....	17
3.4.1 Üksuste ja komponentide süsteem	18
3.4.1.1 Üksuste loogika.....	18
3.4.1.2 Komponentide loogia.....	19
3.4.1.3 Komponentide loomine.....	20
3.4.2 Virtuaalreaalsusesse videovoo lisamine	21
3.4.3 Virtuaalreaalsusesse nuppude lisamine.....	22
3.4.4 Virtuaalreaalsusesse rooli lisamine.....	24
3.4.5 Virtuaalreaalsusesse füüsika ja käte lisamine	25
3.4.6 Mudelauto juhtimine	26
4. Testimine.....	27

5. Kokkuvõte.....	29
5.1 Tänuõnad.....	29
Allikad.....	30
Lisad.....	31
I. Litsents.....	31
II. Kasutatud mõisted.....	32

1. Sissejuhatus

Mudelite ehitamine on eriala, mis hõlmab endas mudelite loomist. Mudel on mingi füüsilise eseme sarnane, vähem keerukam ning enamasti väiksemate mõõtmetega, esitus. Mudeleid saab teha igast füüsilisest objektist, kuid käesolevas uurimistöös keskendutakse mudelautodele, ehk sõiduautosid matkivatele mudelitele.



Joonis 1. RCSnaili mudelautod.

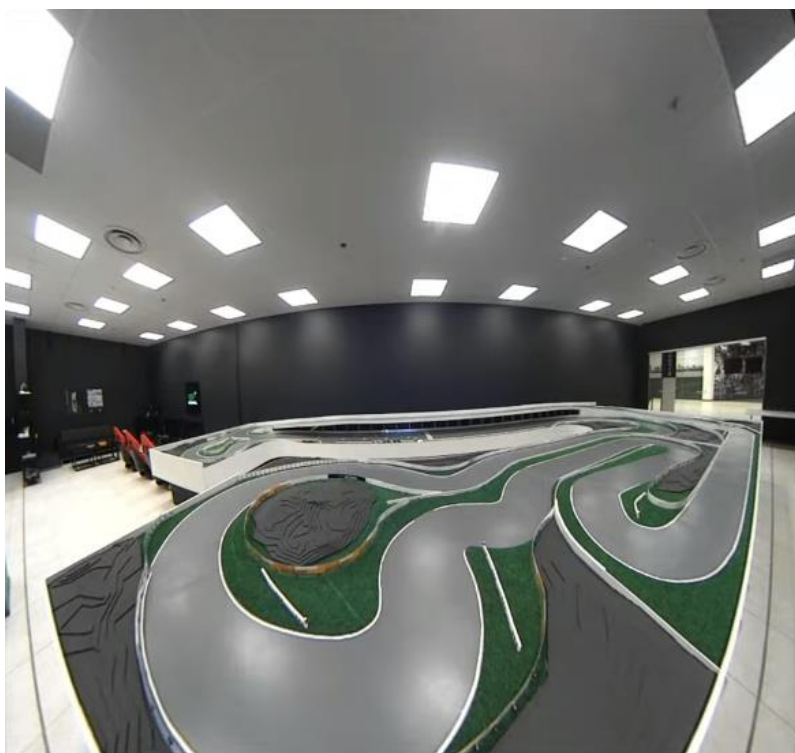
Mudelautosid on peamiselt kolme eritüüpi:

1. statsionaarsed mudelautod, mis ei liigu ja on mõeldud pigem dekoratsiooniks ning kogumiseks
2. *pull-back* (inglise k. tagasi-tõmmatav) mudelautod, mis töötab vedruga ja on mõeldud mängimiseks

3. raadioteel juhitud mudelautod, mis töötavad kas aku või kütuse jõul ning on mõeldud puldi abil juhtimiseks.

Viimastel on üle maailma väga lojaalne kogukond. Nii noored kui ka vanad võistlevad omavahel suurtel ringradadel, et teada saada, kes on kiireim. Ka Eestis on mudelisporti harrastajatel üpriski aktiivne kogukond.^{1,2,3}

RCSnail, edaspidi ettevõtte, on 2017. aastal asutatud meelelahutusettevõtte, mis pakub klientidele võimalust juhtida mudelautosid auto peale paigaldatud kaamera abil. Idee sellist võimalust pakkuda jääb ettevõtte sõnul juba aastasse 2004. Ettevõtte propageerib enda tegevustega keskkonnasäästlikke tehnoloogiaid, milleks on siis elektriautod ning taastuvenergiaallikad. Samuti pakkuda lastele ja nende vanematele põnevat koos ajaveetmise võimalust.



Joonis 2. RCSnaili ringrada.

Ettevõtte ringrada asub Tartu Eedeni kaubanduskeskuses. Mudelautode juhtimiseks on ettevõtte juba välja töötanud erinevaid võimalusi. Näiteks on sõidukogemuse nautimiseks võimalik

¹ <https://www.facebook.com/eestihobi>

² <http://www.mudelismifoorum.eu/>

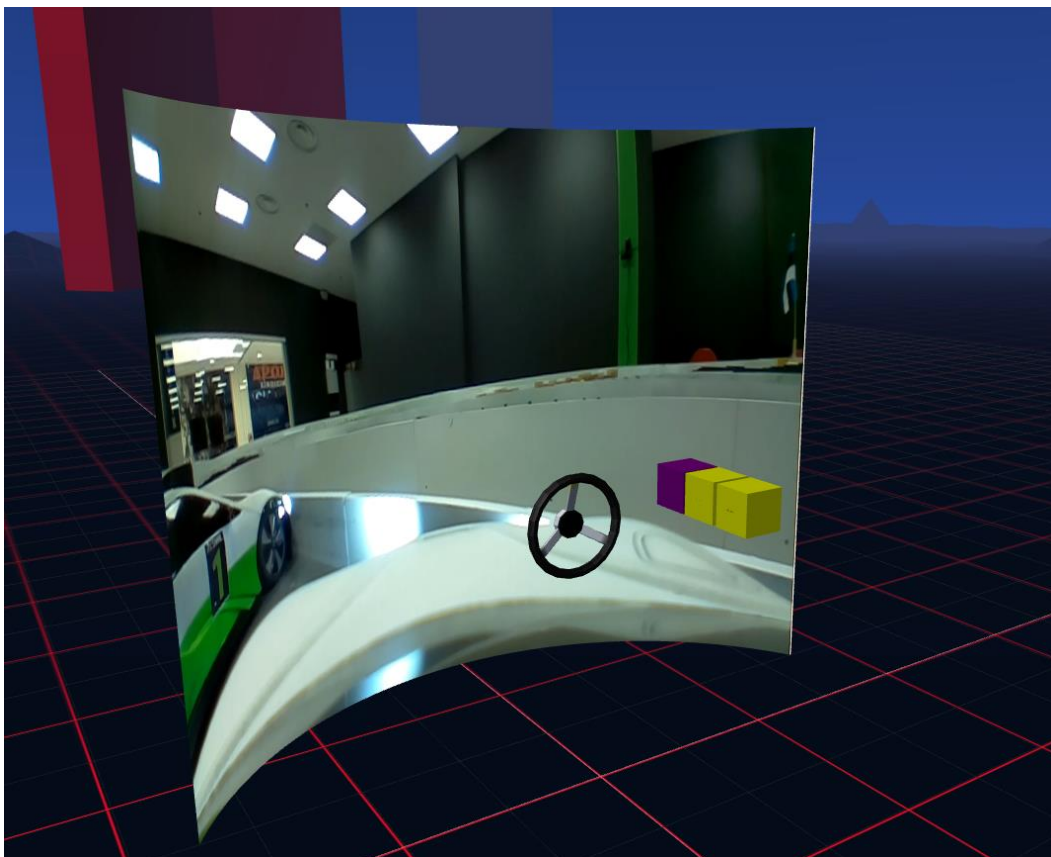
³ <http://modelcar.ee/#>

kasutada Eedenis kohapeal asuvat monitori koos rooli ja pedaalidega. Lisaks on võimalus läbida rada ka oma kodust üle interneti, nutiseadmes või arvutis puldiga.

Käesolev bakalaureusetöö lisab juurde võimaluse mudelautode videovoo edastamise virtuaalreaalsuse peakomplekti ning ka võimekuse virtuaalreaalsuses autosid juhtida. Esialgu saab virtuaalreaalsuses sõita virtuaalse rooli abil, tulevikus soovib ettevõtte aga olemasolevaid roole ja pedaale sisenditeks kasutada.

Ettevõtte tahab pakkuda oma klientidele võimalikult kaasahaaravat kogemust. See tõstaks ettevõtte käivet ning lahendust saaks kasutada ka turunduses, sest oleks Eestis esimene omataoline lahendus.

Bakalaureusetöö esimeses peatükis tutvustatakse erinevaid tehnoloogiaid, mida sarnaseks lahenduseks kasutada saab. Samuti antakse ülevaade iga tehnoloogia nõrkustest ja tugevustest ning selgitatakse ära, miks sai tehnoloogia valitud. Teises peatükis selgitatakse autoripoolset lahendust. Kõigepealt tutvustatakse juba olemasolevaid lahendusi ning seejärel autori lahendust.



Joonis 3. Autori lõpplahendus kolmanda isiku vaatest.

Kolmandas peatükis testitakse lahendust ning analüüsitakse tulemusi. Virtuaalreaalsuses on jõudlusel eriti oluline osakaal, kuna selle vajaka jäämisel on paljudel inimestel küberiivelduse tekkimise oht [3]. Lõputööle on lisatud ka video töötavast lahendusest ning lähtekood ja lisadest leiab ka mõistete seletusi. Lahenduse leiab GitHubi repositooriumist lingil: <https://github.com/rcsnail/rcsnail-vr>

2. Vajalikud tehnoloogiad

Siin peatükis põhjendatakse, miks otsustati mingi kindla tehnoloogia kasuks. Samuti on oluline, et kuna ettevõttel on juba olemasolev lahendus mudelautojuhtimiseks, peaks ka käesolev lahendus sellega ühilduma.

2.1 RCSnail'i olemasolev lahendus

Enne käesoleva lahenduse valmimist oli ettevõttes võimalik sõita ringrajal mudelautoodega rooli, pedaalide ja monitori abil.

Ettevõttel on ka veebipõhine lahendus, mis võimaldab mudelautoodega ühenduda kõikjal üle interneti. Mudelautosse pääseb lehteküljelt <https://play.rcsnail.com/>, kus saab valida, ringraja lahtiolekuaegadel, meelepärase mudelauto ning sellega sõita. Töö kirjutamise ajal olid toetatud nii iOS, kui ka Android nutiseadmed ning Thrustmaster T300 rool ning mängukonsooli Xbox'ga ühilduv pult. Töötavad ka Logitech'i ja teiste tootjate XInput standardile vastavad puldid. Nutiseadmes toimub pööramine seadet kallutades ning on olemas käigukang gaasi, pidurdamise ning tagurdamise jaoks.



Joonis 5. RCSnaili olemasolev lahendus Android seadmes.



Joonis 4. RCSnaili olemasolev lahendus.



Joonis 6. RCSnail mudelautoade valik.

Tegemist on tasulise teenusega, kuid kuna lahendus on töö kirjutamise hetkel veel arendusfaasis, saavad kõik soovijad mudelautodega tasuta sõita.⁴

2.2 Virtuaalreaalsuse peakomplekt Oculus Quest

Lahenduse loomise ajal kasutas autor virtuaalreaalsuse peakomplekti Oculus Quest⁵. Lahenduse mõte on, et soovijad saaks seda kasutada ka kodust ja et see võiks toimida juba seni endale Oculus Go⁶ seadme ostnud kasutajatel. Oculus Go'd täna enam osta ei saa ja ettevõttes endas võetakse tulevikus kasutusele sellel hetkel saadaolevad seadmed, kuid kuna lõpliku lahenduse valmimiseni läheb veel aega, siis ei saa teada, milliseid peakomplekte ettevõtte tulevikus kasutama hakkab. Seda seetõttu, et virtuaalreaalsuse tehnoloogiad on kogumas populaarsust, ning platvorm areneb väga kiiresti. Selle tõestuseks on Oculus välja tulnud juba uue peakomplektiga Oculus Quest 2⁷, mis on võimekam, kui esimene Quest, aga tootja soovituslik jaemüügihind on 100\$ odavam. Kui lahendus töötab hästi vanade peakomplektide peal, siis töötab see hästi ka uuemate peal.



Joonis 7. Oculus Quest peakomplekt pultidega.

Nende kahe peakomplekti suurim vahe on see, et Oculus Go on mõeldud istuvateks virtuaalreaalsuseelamusteks, aga Quest on mõeldud kasutamiseks suuremal alal. Autor võttis seda arvesse ning disainis lahenduse nii, et kasutaja ei pea kuhugi liikuma ja mudelautot saab juhtida ka istudes. Samuti tuleb Go vaid ühe virtuaalreaalsuse puldiga, seevastu Questil on neid

⁴ <https://www.facebook.com/RCSnail/photos/a.1370140053074816/3808686082553522/>

⁵ <https://developer.oculus.com/get-started/>

⁶ https://en.wikipedia.org/wiki/Oculus_Go

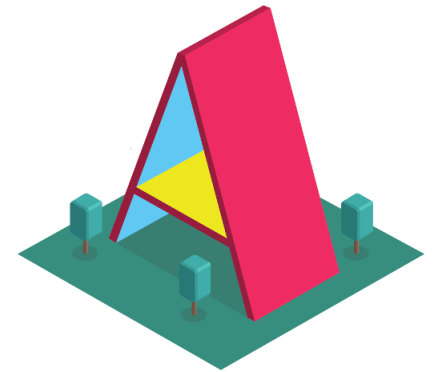
⁷ <https://developer.oculus.com/get-started/>

kaks. Ka seda peeti lahendust luues silmas ning tööga kaasasolevas videos⁸ on ka näha, kuidas mudelautoga sõidetakse ühe käega.

2.3 A-Frame

A-Frame⁹ on veebitarkvararaamistik virtuaalreaalsuse elamuste ehitamiseks. Seda hooldavad arendajad ettevõtetest Google ja Supermedium. A-Frame kasutab populaarset JavaScripti raamistiku Three.js, mis on suunatud veebipõhise 3D-graafika loomiseks.

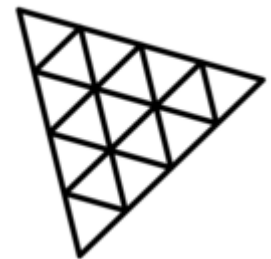
Tehnoloogia sai valitud, kuna ettevõtte RCSnail vajab veebirakendust ning kasutab raamistiku Three.js, millega oli töö autor juba varasemalt tuttav. Samuti kasutab A-Frame WebXR liidest virtuaalreaalsuse seadmetega ühendamiseks.



Joonis 8. A-Frame logo.

2.4 Three.js

Three.js¹⁰ (edaspidi Three) on JavaScripti raamistik, mida kasutatakse veebibrauseris kuvatava 3D-graafika loomiseks ja animeerimiseks. Three teegiga saab luua efekte, stseene, kaameraid, valgusi, materjale, võrestike, varjutajaid, animatsioone ja 3D-objekte. Three kasutab WebGL-i, mis on JavaScripti rakendusliides, mida kasutatakse veebibrauserites interaktiivse 3D- ja 2D-graafika renderdamiseks ilma pistikprogramme kasutamata.



Joonis 9. Three.js logo.

2.5 WebXR

WebXR rakendusliides¹¹ (API) pakub juurdepääsu nii virtuaalreaalsuse peakomplekti ja pultide sensoritele, peakomplekti kuvarile kui ka liitreaalsuse seadmetele. See



Joonis 10. WebXR logo.

⁸ Video on kättesaadav ka siit: <https://youtu.be/5809R3btQVo>

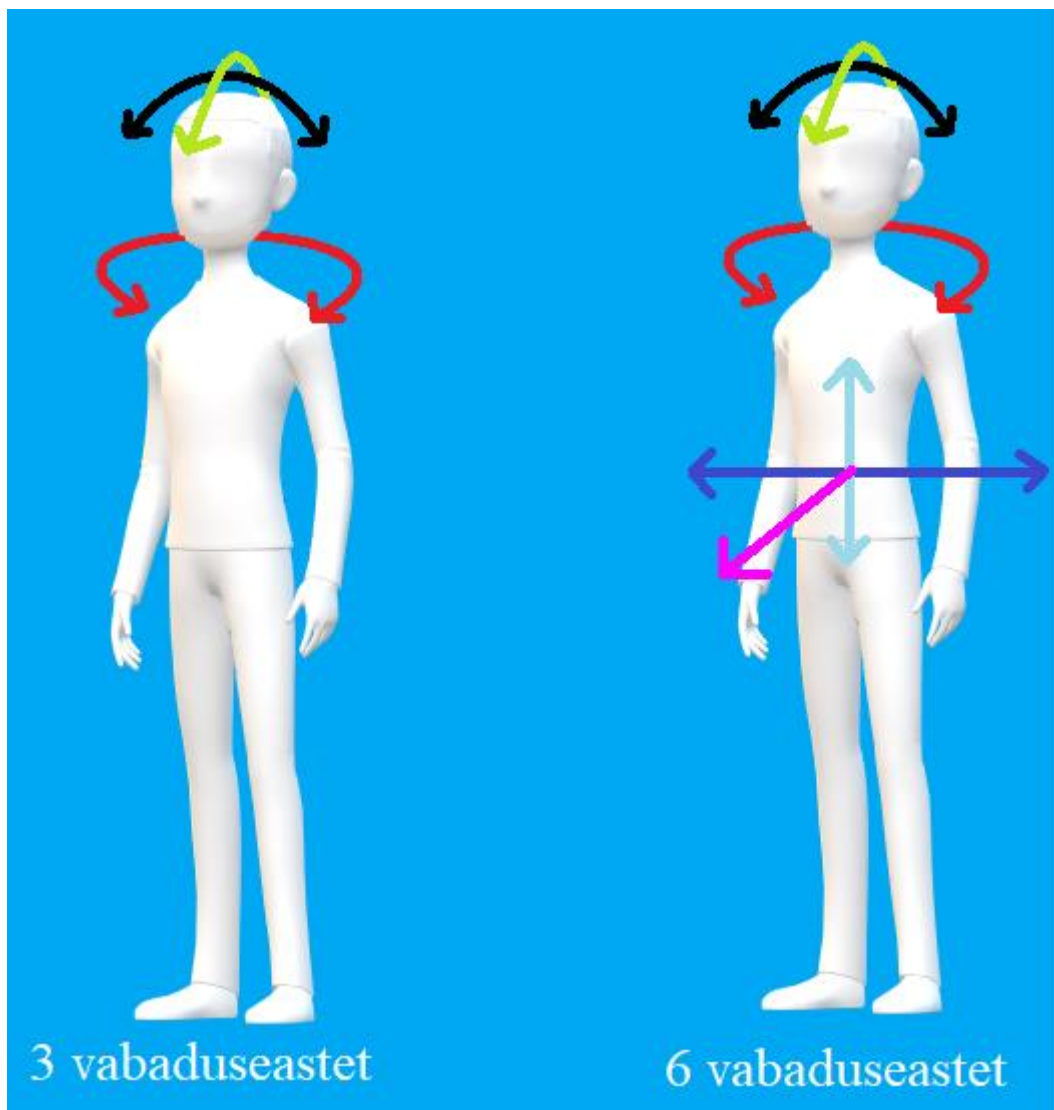
⁹ <https://aframe.io/>

¹⁰ <https://threejs.org/>

¹¹ <https://immersiveweb.dev/>

võimaldab arendada veebis virtuaalreaalsuse ja liitreaalsuse kogemusi.

WebXR tunneb ära, kas mõni virtuaal- või liitreaalsuse seade on ühendatud ning, mis on selle seadme võimekus. Virtuaalreaalsuses eristatakse kahte vabadusastme taset: kolm vabadusastet ja kuus vabadusastet. Erinevad vabadusastmed näitavad seadme jälgimise võimekust. Kolme vabadusastmega seadmed suudavad jälgida seadme pöördliikumist. Kuue vabadusastmega seadmed suudavad lisaks jälgida veel seadme liikumist üles-alla, vasakule-paremale, edasi-tagasi.



Joonis 11. Kaks erinevat vabadusastet.

Kolme vabadusastmega seadete eelis on nende hind ning kättesaadavus. Pea kõik nutiseadmed on kolme vabadusastme võimekusega seadmed¹² ning, kasutades näiteks Google Cardboard'i¹³, saab kolme vabadusastmega virtuaalreaalsuse palju soodsamalt.

2.6 Virtuaalreaalsuse silumine

Silumine (inglise k. *debugging*) on iga tarkvaraarenduse juurde kuuluv protsess, mille kaudu leitakse ja parandatakse vigu programmi lähtekoodis. Veebirakenduste puhul on lihtsaim viis kasutada veebilehitsejatesse sisseehitatud arendustööriistu. Virtuaalreaalsuse rakenduses ei saa arendaja samal ajal veebilehitsejas olevaid tööriistu avada. Selle probleemi lahendamiseks on mitmeid võimalusi, millest tutvustame kolme selle töö raames.

Esimene ja ilmselt kõige mugavam lahendus on virtuaalreaalsuse peakomplekti ühendamine juhtmega arvuti külge. Oculus Quest'i puhul kutsutakse seda ühendust nimega Oculus Link ning tänu sellele ühendusele kasutab peakomplekt hoopiski arvuti veebilehitsejat, ehk samaaegselt saab arvutist ligi veebilehitseja arendaja tööriistadele. Negatiivseteks omadusteks selle lahenduse puhul on halb ühilduvus. Link'i edukaks ühenduseks töötavad vaid teatud graafikakaardid¹⁴ ning emaplaat peab ka toetama USB-C/3.0 tüüpi pesasid, mis ei ole veel nii laialdaselt levinud.

Teiseks lahenduseks on A-Frame'is liikumise jäädvustamine¹⁵ ning selle taasesitamine. A-Frame võimaldab virtuaalreaalsuses tehtud liigutusi ja interaktsioone salvestada JSON-tüüpi failidesse ning hiljem neid taasesitada. See annab arendajale mugavuse arendada ilma peakomplektita. Samuti on see hea viis rakenduse automaat-testimiseks, sest see kõrvaldab sisendist tulevad variatsioonid.

Kolmandaks on võimalus ühendada peakomplekt ühe juhtmevaba võrguga. Seda lahendust kasutas ka autor ning selleks kasutatakse tööriista adb¹⁶ (inglise k. Android Debug Bridge), ehk Androidi silumissild. Selleks, et ühendus tööle saada, tuleb kõigepealt alla laadida Androidi tarkvaraarenduse komplekt¹⁷. Seejärel ühendada peakomplekt arvutiga ning kasutades käsurida

¹² <https://www.mechatech.co.uk/journal/what-is-a-3dof-vs-6dof-in-vr>

¹³ <https://arvr.google.com/cardboard/>

¹⁴ <https://support.oculus.com/444256562873335/>

¹⁵ <https://github.com/dmarcos/afame-motion-capture-components/>

¹⁶ <https://developer.android.com/studio/command-line/adb>

¹⁷ <https://developer.android.com/studio/releases/platform-tools>

kontrollida käsuga `adb devices`, kas arvuti ühendas peakomplektiga. Kindlasti tuleb lubada peakomplektist USB-silumine. Seejärel peavad olema nii arvuti kui ka peakomplekt samas võrgus ning, kasutades jälle käsurida, tuleb anda kaks käsku, `adb tcpip <port>`, et port oleks õige ning seejärel `adb connect <ipaddress>:<port>`, kus `<port>` on 5555 ning `<ipaddress>` on peakomplekti IP-address. Seejärel veebilehitseja Google Chrome otsinguribasse sisestada `chrome://inspect/#devices` ning sealt saab ligi peakomplekti veebilehitseja arendaja tööriistadele.

2.7 Alternatiivsed tehnoloogiad

Esialgu oli lahenduse loomiseks kavas kasutada mängumootorit Unity või Unreal Engine, aga lõpuks otsustas ettevõtte siiski A-Frame'i kasuks. Veebipõhine lahendus võimaldab väiksema töömahuga ideed valideerida, kuna aluseks sai võtta juba olemasoleva veebipõhise autode juhtimise rakenduse koodi. Sellisel viisil rakenduse levitamine on oluliselt lihtsam ning antud rakenduse jaoks on veebilahenduse pakutav võimekus virtuaalreaalsuse prillides piisav (ei ole vaja keerukaid 3D mudeleid). See ei tähenda, et tulevikus ei võiks siiski Unity, Unreal või Godot peale mõelda näteks. Järgnevad peatükid toovadki välja nende tehnoloogiate positiivsed ja negatiivsed omadused.

2.7.1 Unity ja Unreal Engine

Unity ja Unreal Engine on videomänguplatvormil Steam virtuaalreaalsusmängude jaoks kõige populaarsemad mängumootorid. Neid kasutatakse veidi üle kolmandiku mängude jaoks¹⁸.

Otsus neid mängumootoreid käesolevas lõputöös mitte kasutada põhines kolmel põhjusel. Esiteks, oli ettevõtte just loonud uue veebipõhise rakenduse mudelautodega sõitmise jaoks ning nende uus suund ongi liikuda veebipõhiseks platvormiks. Kuigi nii Unity kui ka Unreal Engine toetavad veebiprogramme, on A-Frame'l eelis lihtsuses ja kiiruses¹⁹. Teiseks on A-Frame JavaScripti põhine ja just mõeldudki virtuaalreaalsuse veebi toomiseks. Juba paari koodireaga saab töötava virtuaalreaalsuse maailma luua²⁰. Kolmandaks, lahendus ei vaja ka mingit erilist

¹⁸ <https://circuitstream.com/blog/unity-vs-unreal/>

¹⁹ <https://aframe.io/docs/1.2.0/introduction/faq.html#how-is-a-frame%E2%80%99s-performance>

²⁰ https://gamechef.io/unity_vs_a-frame

keerukat funktsionaalsust, mida Unity ja Unreal Engine seevastu pakuvad, saigi valik tehtud A-Frame'i poolt.

2.7.2 Babylon.js

Autor kaalus ka Babylon.js'i²¹ (edaspidi Babylon) kasutamist. Babylon on nagu Threegi JavaScripti raamistik 3D-graafika loomiseks. Babyloni arendab Microsoft ning pakub väga häid rakenduste arenduskeskondi ning dokumentatsiooni. Kuna aga A-Frame on üles ehitatud Three peale ning Babyloni virtuaalreaalsusel on suurem riisvaraline nõudlus kui Threel, siis jäi see valik ära.

3. Lahendus

Käesolevas peatükis antakse ülevaade virtuaalreaalsusesse toodud kaugjuhitavate mudelautode videoülekanne lahendusele. Uuritakse, miks lahendus üldse vajalik on, tutvustatakse erinevaid, juba olemasolevaid, lahendusi ning seejärel antakse ülevaade autori poolt väljatöötatud lahendusest.

3.1 Lahenduse kasu ja eelised

VR lahenduste kasutamisel videomängudel on mitmeid eeliseid. Kahjuks on teaduslikku kirjandust, mis võrdleks inimeste mängukogemusi virtuaalreaalsuses ja tavalisel monitoril on tänaseks siiski veel vähe. Sellegipoolest saame väita, et virtuaalreaalsus pakub kaasahaaravat, põnevamat ja realistlikumat mängukogemust, seda nii audiovisuaalselt kui ka füüsiliselt mängijat kaasa haarates. Peamine vahe tavamängude ja VR lahendusi kasutatavate mängude vahel ongi nende immersiiivsus mängukogemusesse, mis võimendavad mängijate emotsioone ja kaasatust mängu [6]. Tänu ümbritseva heli ja pildi kasutusele kaovad ka välised segavad faktorid ja tähelepanu püsib teravamana. Paremini keskendunud mängija saavutab ka edukamaid tulemusi.

Saame ka väita, et kasu seisneb nii eeltoodud eelistes kui ka selles, et mängijad on mängu ajal liikuvad. See aitab teha mängimist tervislikumaks kogemuseks, kui lihtsalt arvutiekraani taga hiire ja mõne klaviatuurinupu kasutamine. Kuigi virtuaalreaalsuse pikaajaline mõju pole

²¹ <https://www.babylonjs.com/>

veel palju uuritud teema ja e-sport ei ole kindlasti kõige tervislikum spordiliik, siis ikkagi kulutab mängija selle käigus rohkem kaloreid ja hoiab rohkem lihaseid ja kehaosi liikuvamana kui niisama paigal istudes ekraani vaadates.

Innovatiivsete ja uudsete mängukogemust arendavate lahenduste, nagu seda on virtuaalreaalsus, kasutuselevõtt avab palju uusi mänguvalikuid nii mängijatele endile kui ka mängutootjatele. Ka ettevõtted, nagu RCSnail saavad pakkuda oma klientidele midagi uut ja põnevat, suurendades seeläbi oma kliendibaasi ja tuntust.

Kui lisada aga juurde virtuaalreaalsuse prillid, muutub mäng juba suurema publiku jaoks huvitavaks, sest seeläbi muutub tegelikult kaugel asuv auto palju käegakatsutavamaks ja tekib tunne justkui istukski päriselt selle roolis. Seda nii kogenuid mängijatele, kui ka neile, kes poleks muidu kunagi sellist mängu proovinud, aga on huvitatud virtuaalreaalsuse proovimisest.

3.2 Olemasolevad lahendused

Sarnast lahendust on proovitud teha ka juba varem. Linus Sebastian proovis teha sarnast tarkvaralahendust tema korraldavale tehnikaüritusele LTX [9]. Tal valmis kontseptsiooni tõestus, aga LTX jäi kahjuks maailmas leviva viiruse COVID-19 tõttu ära, ning projekt jäigi kontseptsioonifaasi. Sebastian kasutas enda lahenduses ettevõtte DJI kaamerasüsteemi, peakomplekti ning Thrustmasteri rooli ning pedaale. DJI on ettevõtte, mis on toodab droone nii loodusfotograafidele, kui ka droonivõistlustele. Kuna droonid mudelautodest palju kiiremad, siis ei olnud neil latensusega mingeid probleeme. Sebastiani ja käesoleva töö lahenduste erinevus on selles, et Sebastian lähenes probleemile riistvaraliselt ning tema lahenduses näidatakse mudelautost tulevat kaamerapilti otse peakomplektis. Autori lähenemine on aga tarkvaralisem. Riistvaraline pool on juba ettevõtte poolt ette antud ning see tuli sobitada juba olemasoleva lahendusega.

Täislahenduse on aga valmis saanud Clemens Mayer, kes kasutas Arduino mikrokontrollerit, vana mängukonsooli rooli ning VR peakomplekti oma mudelautoga sõitmiseks [6]. Tema soov on teha oma projektist avatud standard kõigile kasutamiseks ja samuti hakata müüma valmiskomplekte, et ka kõik huvilised elamusest osa saaksid. Ka see lahendus on pigem riistvaraline, aga ettevõtte soovib toetada mitmeid eri mudelauto juhtimise võimalusi. Ahmadi lahendus nii paindlik ei ole, sest ka seal on kaamera mudelautoga otseühenduses.

3.3 Kaamerate kalibreerimine

Kõik kaamerad ning läätсед moonutavad veidi nendega tehtud pilte ja videoid²². See moonutus on iga kaamera ja läätse paari puhul erinev. Parima võimaliku pildikvaliteedi saamiseks tuleb kaamera kalibreerida. See tähendab, et leida kaamera (moonutus)parameetrid, sealhulgas fookuskaugus ning moonuse keskpunkt ning neid kasutades tarkvara abil kaamerast tulevat pilti töödelda.

Kaamera parameetrite leidmiseks kasutatakse enamasti kindlaid mustreid, eriti populaarsed on malelauamustrid²³ kuna malelaua ruudud on kõik ühe suuruse ning kujuga. Kõigepealt tuleb teha kaameraga malelauast palju pilte, mida rohkem, seda täpsem on tulemus. Oluline ei ole malelaua orientatsioon ja kalle, küll aga, et malelaud oleks tervenisti kaadris ning tasane. Seejärel saab tehtud pildid parameetri leidmine programmile sisendiks anda. Autor kasutas selleks programmeerimiskeeles Python kirjutatud programmi, mis oli valminud aine Arvutigraafika raames²⁴.

Järgmiseks sammuks on moonutatud pildi varjutajaga (inglise k. *shader*) korrigeerimine. Selleni aga autor ajapuuduse tõttu ei jõudnud ning varjutaja kasutamine jääb projekti edasiarenduseks.

3.4 Lahendus A-Frame'is

Selleks, et mudelauto videovoog jõuaks VR peakomplektini, kasutasin tarkvararaamistikku A-Frame. A-Frame'i saab arendada tavalises HTML-failis. Faili struktuur näeb välja järgmine:

```
<html>
  <head>
<script
src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>
  </head>
  <body>
    <a-scene>
      <a-assets>
```

²² <https://www.sciencedirect.com/topics/engineering/lens-distortion>

²³ <https://dsp.stackexchange.com/questions/24734/camera-calibration-why-chessboards>

²⁴ <https://courses.cs.ut.ee/2020/cg/fall/Main/Project-VRLiveRacingForRCCars>

```
        <a-asset-item></a-asset-item>
    </a-assets>
    <a-entity></a-entity>
</a-scene>
</body>
</html>
```

A-Frame kasutamiseks tulebki lihtsalt faili päisesse lisada A-Frame'i uusim versiooni, mis käesoleva töö kirjutamise ajal on 1.2.0. A-Frame'i stseen tuleb lisada faili kehasse. Kõik loodud üksused tuleb lisada stseeni.

3.4.1 Üksuste ja komponentide süsteem

A-Frame kasutab üksuste ja komponentide süsteemi (inglise k. *Entity component system*). See tähendab, et iga stseenis olev objekt on üldine ning enamasti on neil ainult üks ainulaadne id, mille järgi neid omavahel eristada.

A-Frame stseeni loogika lähtekoodis on järgmine:

```
<a-scene>
  <a-assets>
    <a-asset-item></a-asset-item>
    ...
  </a-assets>
  <a-entity></a-entity>
  ...
</a-scene>
```

Andmete (inglise k. *assets*) plokk koodis on vajalik andmete eellaadimiseks. See kindlustab, stseeni hakatakse renderdama on kõik andmed, näiteks 3d-mudelid, pildid ja helid, alla laetud. Samuti aitab see kaasa ka jõudlusele, kuna andmed laetakse enne stseeni renderdamist ära, jääb rohkem arvutusressursse vabaks.

3.4.1.1 Üksuste loogika

Kõikide üksuste struktuur koos komponentidega on järgmine:

```
<a-entity component_1 component_2></a-entity>
```

Igale üksusele saab lisada suvalise arvu komponente, mis annavad objektile tema välimuse ja kuidas ta teiste objektidega suhtleb. Samuti on A-Frame'is olemas ka juba mõned primitiivid, nagu näiteks kast, koonus, kera, aga ka tekst, taevas ja heli. Need on tehtud programmeerija elu mugavamaks tegemiseks, ehk on juba mingite eelväärtustatud komponentidega üksused. Käesolevas lahenduses kasutati ka mõnda primitiivi ning neid kirjeldatakse täpsemalt järgnevates peatükkides.

3.4.1.2 Komponentide loogia

Komponendid (inglise k. *components*) on modulaarsed, taaskasutatavad andmed, mida saab üksustele juurde lisada. A-Frame'is on juba olemas ka mitmeid eeldefineeritud komponente, nagu positsioon ja rotatsioon, aga ka keerukamaid komponente näiteks juhtimispultide jaoks. Mitmete komponentide kombineerimisel saab luua keerukaid üksuseid. Komponente on kahte tüüpi:

1. Ühe omadusega komponent
2. Mitme omadusega komponent.

Ühe omadusega komponentide lisamine ja muutmine on väga lihtne. Näiteks positsiooni komponendi üksusele lisamine näeb välja selline:

```
<a-entity position="1 2 3"></a-entity>
```

Selles näites on üksuse koordinaadid $x=1$, $y=2$ ja $z=3$. Kui jätta komponent väärtustamata, ehk lihtsalt:

```
<a-entity position></a-entity>
```

Saab üksus enda positsiooni väärtusteks komponendi vaikeväärtused, mis antud näite puhul on $x=0$, $y=0$ ja $z=0$.

Mitme omadusega komponentide väärtustamine toimub sarnaselt ühe omadusega komponentide väärtustamisele. Näiteks üksusele punase teksti "Tere Maailm" lisamine näeb välja selline:

```
<-entity text="value: Tere Maailm; color: red;"></a-entity>
```

Teksti komponendil on tegelikult rohkem omadusi kui ainult väärtus (inglise k. *value*) ja värv (inglise k. *color*), aga arendaja saab ise valida, milliseid ta muuta tahab, kui jätta väärtus muutmata, jäävad kasutusse vaikeväärtused.

3.4.1.3 Komponentide loomine

Tihti peale jääb A-Frame'is eeldefineeritud komponentidest väheseks. Sellisel juhul on kaks võimalust: esiteks, kas otsida kellegi teise poolt kirjutatud komponent või siis vajalik komponent ise kirjutada. Autor kasutas töös teiste arendajate poolt kirjutatud komponente üldisema funktsionaalsuse loomiseks, näiteks virtuaalreaalsusepultide jälgimise jaoks, aga kirjutas ka mõne komponendi ise, näiteks nuppude ning auto juhtimise funktsionaalsuse.

Komponente kirjutatakse programmeerimiskeeles JavaScript.

Komponendi loomiseks tuleb luua uus js.laiendiga fail ning see lisada ka html-faili päisesse. Komponendi nimega “minuKomponent” loomise loogika on järgmine:

```
AFRAME.registerComponent("minuKomponent", {
  schema: {},
  init: function () {},
  update: function () {},
  tick: function () {},
  remove: function () {},
  pause: function () {},
  play: function () {}
});
```

Komponent koosneb skeemist (inglise k. *schema*) ja programmi tsükli jälgivatest meetoditest (inglise k. *Lifecycle Handler Methods*). Skeemis hoitakse komponendi väärtuseid. Skeemi struktuur on järgmine:

```
schema: {
  color: {default: '#FFF'},
  size: {type: 'int', default: 5}
}
```

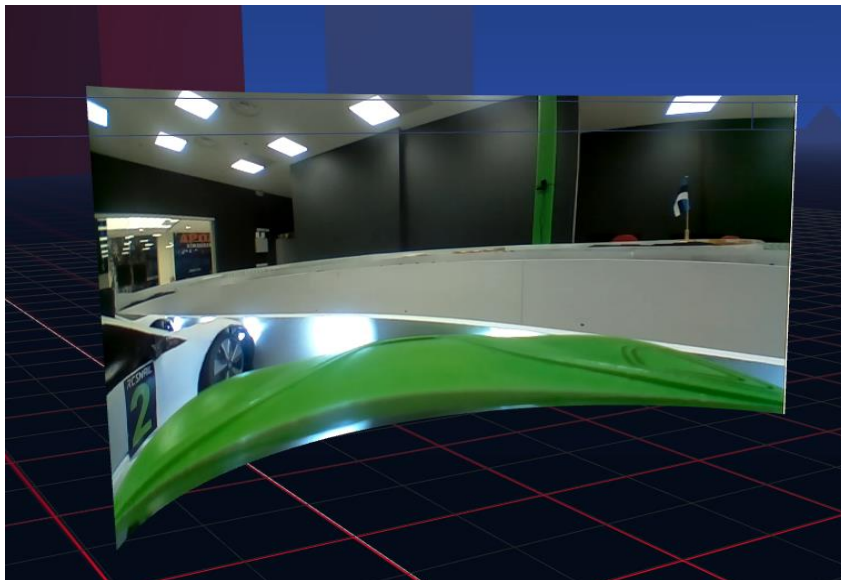
Nagu näitest näha saab skeemi lisada erinevat tüüpi väärtuseid. Samuti saab komponendi meetodites neid väärtusi muuta.

Komponendi meetodeid tuleb kasutada vastavalt selle, millal tahetakse, et mingit koodi jooksutatakse. Autor kasutas oma lahenduses neist kolme:

1. meetodit **init** kutsutakse välja üks kord, kui komponent luuakse. Seda kasutatakse näiteks üksuse algse oleku ning muutujate väärtustamiseks.
2. meetodit **tick** kutsutakse välja iga kaadri renderdamise korral. Seda kasutatakse, kui on vaja pidevalt üksust muuta või tema olekut kontrollida.
3. meetodit **update** kutsutakse välja iga kord, kui komponendi skeemis midagi muutub. Seda kasutatakse, kui ongi vaja teha komponendile muudatusi.

3.4.2 Virtuaalreaalsusesse videovoo lisamine

Töö aluseks on ettevõttes välja töötatud veebipõhine autode juhtimise testprogramm, millele autor lisas virtuaalreaalsuse toe. Autost videopildi kättesaamine oli ettevõtte lahenduses juba olemas, aga selle pidi tegema A-Frame'ile arusaadavaks. Autori idee oli, et luua virtuaalreaalsusesse suur kumer ekraan, mille peale autost tulev kaamerapilt kuvada.



Joonis 12. Silinder videovooga.

Kui rakendusse siseneda virtuaalreaalsuse peakomplektiga on see ekraan kasutajale nii lähedal, et kogu ta vaateväli on sellega kaetud. Selleks, et seda ekraani luua, kasutasin A-Frame'i silindri primitiivi. Silindri lähtekood on järgmine:

```
<a-cylinder radius="1"
```

```

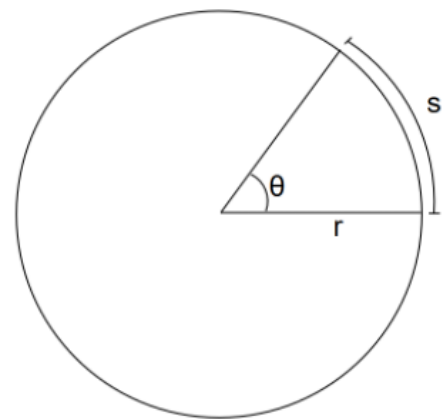
    height="3.1417"
    scale="-1 1 1"
    material="uMap:      #remote-video;      side:      back;
shader:distortion;"
    open-ended="true"
    theta-start="120"
    theta-length="120"
    position="0 1.5 0.5">
</a-cylinder>

```

Silindri õige suurus ja raadius oleneb kaamera vaateväljast ning seda arvutatakse järgneva valemiga:

$$S = r * \theta \quad (1)$$

Kus S on ringjoonelõigu pikkus, r on kaugus silindrist ning θ on autost tuleva videopildi suurus radiaanides. Samuti mängib olulist rolli silindri kõrgus. Silindri kõrguse ja ringjoonelõigu pikkuse kuvasuhe peab jääma samaks, muidu saame moonutatud pildi.



Joonis 13. Silindri arvutusvalem.

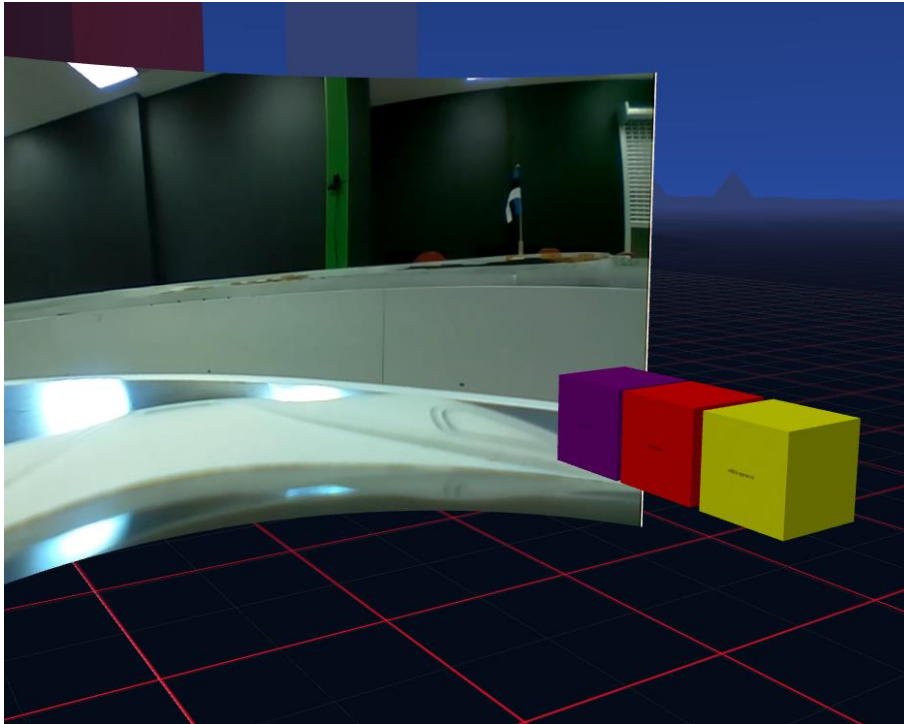
Kuna lisame tekstuuri silindri siseküljele, siis on sinna ilmuv pilt peegelpildis. Selle lahendamiseks saab kasutada silindri mõõteskaala muutujat. Muutes skaala x väärtuse miinus üheks saamegi videopildi peegelpildi, ehk siis kui me vaatame silindrit seestpoolt, siis meie vaatest õiget pidi.

3.4.3 Virtuaalreaalsusesse nuppude lisamine

Lisaks tuli veel lahendusse lisada mõned nupud, et kasutaja saaks oma virtuaalreaalsuseprillid ühendada mudelautoga, mõne muu auto valida või ühenduse sulgeda. Selleks tegi autor kolm eri värvi kasti, mida kasutaja virtuaalreaalsuse puldiga vajutada saab.

Auto vahetuse nuppu vajutades saab valida kõigi kuue auto vahel. Nupu peal olev tekst näitab ka vastavalt, mis autoga ühendama hakatakse. Autot saab vahetada vaid siis, kui ei olda ühegi

autoga ühenduses. Ühenduse sulgemise nupp peatab mudelauto ja rakenduse vahelise andmevoo.



Joonis 14. Nupud virtuaalmaailmas.

Järgnev koodinäide on autosse ühendamise nupu loomine stseenis:

```
<a-box addtoqueue id="add-to-queue-vr" position="0.496 1.370  
0.437"  
  rotation="0 270 0" color="blue" scale="0.2 0.2 0.2"  
  grabbable dynamic-body text="value: add-to-queue-vr;  
  align: center; baseline: center; zOffset:0.51;  
  color: black;" class="clickable"  
></a-box>
```

Igal nupul on ka autori loodud komponent, kuna igal nupul on erinev funktsionaalsus. Eelnevatoodud koodinäites on kollasega on märgitud autori loodud nupu komponent. Rohelisega on esiletoodud komponendid, mis vastutavad selle eest, et virtuaalreaalsuse puldid nuppudega suhelda saaks.

Autosse ühendamise komponent näeb välja selline:

```
AFRAME.registerComponent('addtoqueue', {
```

```

dependencies: ['raycaster'],

init: function () {
  var el = this.el;
  var abi;

  el.addEventListener('gripdown', abi = function () {
    el.setAttribute("material", 'color', "yellow");
    let q = `https://api.rcsnail.com/v1/queue`;

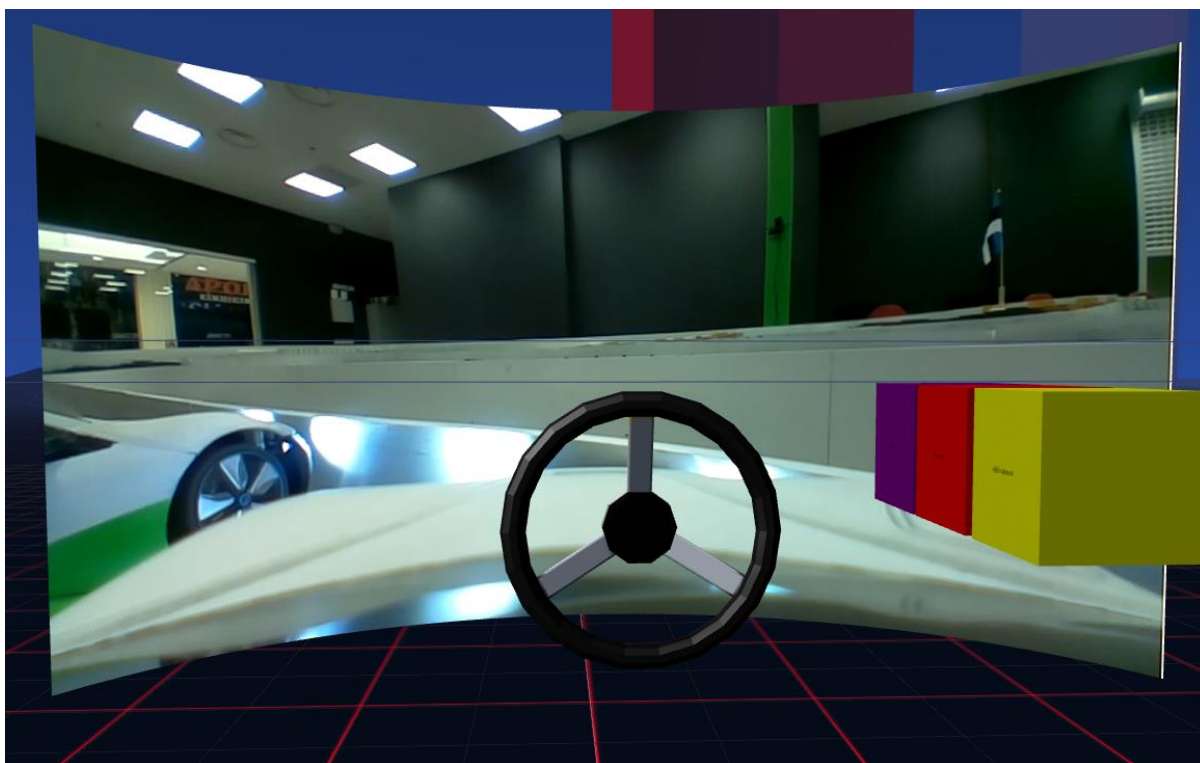
    postJson(q, {track: remoteTrack, sid: "dev", car:
carName})
      .then(data => {...})
      .catch(error => console.error(error));
  });
  el.addEventListener('click', abi);
}
});

```

Kasti peal on ka kirjas, nupu funktsionaalsus, samuti muudab nupp värvi, kui seda vajutada (kollasega esiletoodud), et kasutajale nupu vajutusest tagasisidet anda. Rohelisega esiletoodud kaks erinevat sündmust (inglise k. *event*). Kuna erinevatel virtuaalreaalsuse pultidel on erinevad nupud, siis antud komponent reageerib kahele nupuvajutamise sündmusele: päästiku allavajutamisele ning tagavaravariandina hiire vasaku klahvi vajutusele. See võimaldab, lisaks virtuaalreaalsuse puldile, nuppu vajutada ka hiirega.

3.4.4 Virtuaalreaalsusesse rooli lisamine

Auto juhtimiseks tuli lisada rool. See loodi 3D modelleerimistarkvara Blender. Rool on esialgne variant testimiseks, produktsiooniversiooniks saab teha uue keerulisema rooli ning seda mudelit saab väga lihtsalt muuta, jättes rooli funktsionaalsuse samaks.

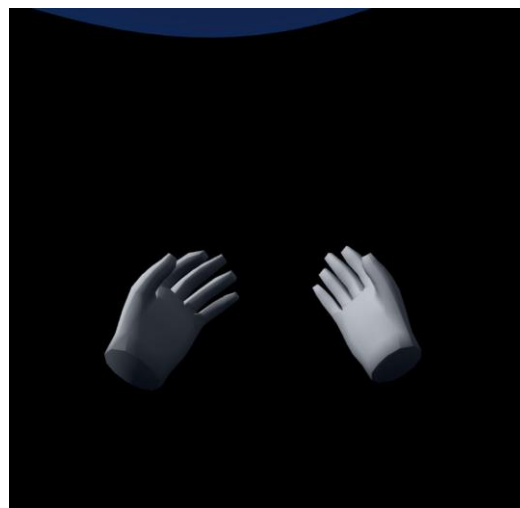


Joonis 15. Rool virtuaalmaailmas.

Rooli mudel eellaetakse, et kui stseeni renderdama hakatakse, on mudel juba stseenis olemas. Samuti hoiab see ära ka kõikvõimalikud veateated, mis on seotud sellega, et rooli ei saa luua, sest mudelit ei ole allalaetud. Eellaadimise negatiivseks küljeks on aga see, et rakendust avades tuleb mõned hetked oodata, kuni mudelit alla laetakse.

3.4.5 Virtuaalreaalsusesse füüsika ja käte lisamine

Kõigepealt lisas autor kaks üksust, millest üks on vasaku, ja teine parema virtuaalreaalsuse puldi jaoks. Kasutasin William Murphy loodud super-hands komponenti.²⁵ Tänu sellele saab kasutaja rooli ja nuppe vajutada ning paigast liigutada. Komponent võimaldab ka üksuste suuruse muutmist, kui neist kahe käega kinni hoida ning käte vahelist kaugust muuta, aga seda funktsionaalsust autoril vaja ei olnud. Samuti tuli roolile lisada juurde füüsikalised omadused, muidu sai rooli liigutada vaid paralleelselt x-, y- ja z-teljel, aga mitte keerata ümber oma telje. Selleks



Joonis 16. Virtuaalsed käed.

²⁵ <https://github.com/wmurphyrd/aframe-super-hands-component>

kasutasin sama autori poolt loodud füüsika komponenti²⁶, mis on loodud töötama koos komponendiga super-hands.

3.4.6 Mudelauto juhtimine

Mudelauto juhtimiseks peab kasutama rooli ning virtuaalreaalsuse pultidel olevaid päästikuid. Selle jaoks kirjutasin ka uue komponendi, mis jälgib, kas virtuaalreaalsuse puldi päästikut on allavajutatud ja vastavalt sellele mudelauto sõidab, kas edasi või tagasi. Samuti jälgib komponent rooli asendit. Kuna ettevõtte olemasolev lahendus kasutab mudelauto rataste keeramiseks vahemikku [-1,1], aga autori lahenduses olevat rooli saab lõpmatuseni ühes suunas keerata, tuli rooli pöörlemise väärtus normaliseerida. Selleks kasutasin valemit:

$$2 * \frac{\text{rooliHetkePööre} - \text{väikseimVäärtus}}{\text{suurmiVäärtus} - \text{väikseimVäärtus}} - 1 \quad (2)$$

Suurimaks väärtuseks valisin 1000 ning väikseimaks -1000, et rool liiga tundlik ei oleks. Juurde lisasin ka erandjuhu, kui rooli hetke pöörde väärtus on suurem kui suurim või väiksem kui väikseim väärtus, siis määrab rakendus väärtuseks vastavalt kas 1 või -1, et ei tekiks olemasoleva lahendusega konflikti.

Lõputööga kaasas olevates failides on ka demovideo mudelautoga sõitmisest antud lahendusega.

²⁶ <https://github.com/wmurphyrd/afame-physics-extras>

4. Testimine

Tarkvaraarenduse lahutamatu osa on testimine. Testimisega leiame loodud tarkvara murekohad, mida parandada või edasi arendada.

Virtuaalreaalsuses on väga oluline, et kõik töötaks võimalikult sujuvalt. Samuti peab olema latentsus sisendseadmest, virtuaalreaalsuse puhul pult väga väike, muidu kannatab kasutajakogemus ning võib ka tekkida küberiiveldus [2,3]. Sama kehtib ka liiga väikse kaadrisageduse korral [1].

Tarkvara testimiseks kasutasin arendamise ajal kasutatud Oculus Questi ning ka peakomplekti HTC Vive Pro, et teada saada, kuidas tulemus muutub, kuna HTC Vive Pro on pea kolm korda kallim ning mõeldud professionaalseks kasutamiseks. Lahenduse testimiseks kasutati A-Frame'i sisseehitatud statistika (inglise k. stats) komponenti. Mõõdeti keskmist kaadrisagedust kahes situatsioonis: esiteks peakomplekti ja pulte rahulikult käes hoides ning teiseks peakomplekti ning pulte raputades, simuleerides intensiivset kasutamist. Saadud tulemused on nähtaval järgnevas tabelis.

Tabel 1. Virtuaalreaalsuskomplektide jõudluse mõõtmistulemused.

	Oculus Quest	HTC Vive Pro ²⁷
Keskmine kaadrisagedus ühes sekundis	55.8	59.9
Keskmine latentsus millisekundites	16.4	16.7
Tootja soovituslik jaehind	\$399	\$1199

²⁷ https://www.niora.net/en/p/htc_vive_pro

Tabelist näeme, et kolm korda kallim HTC Vive Pro on vaid paar kaadrit sekundis kiirem, kui Oculus Quest. Questi madalam keskmine kaadrisagedus tulenes sellest, et kui liigutati virtuaalreaalsuse peakomplekt ja pulte liiga kiiresti, tekkis peakomplektil raskusi puldide asukoha määramisega, kuna sellel peakomplektil on sensorid ainult peakomplekti küljes. HTC Vive Pro'l seda probleemi aga ei tekkinud, sest sellel on eraldi majakad peakomplekti ja pultide asukoha määramiseks. Tavapärase kasutamise korral suutsid mõlemad peakomplektid stabiilselt kaadrisagedust 60 hoida, mis on ka käesoleva lahenduse piir. Latentsus on mõlema peakomplekti puhul ideaalne.²⁸

Lahenduse hea jõudlus on tingitud sellest, et kasutatud tehnoloogiad on väga hästi optimeeritud. Samuti ei ole lahenduses resursinõudlikke elemente, mis aeglasemale peakomplektile raskusi valmistaks. Antud tulemustest võib eeldada, et lahendus töötab sujuvalt kõikidel peakomplektide.

28

<http://www.javascriptkit.com/javatutors/requestanimationframe.shtml#:~:text=The%20standard%20requestAnimationFrame%20runs%20at.rate%20of%20the%20typical%20monitor.>

5. Kokkuvõte

Bakalaureusetöö eesmärk oli luua ettevõtte RCSnail olemasolevale lahendusele virtuaalreaalsuse tugi.

Töö esimeses osas kirjeldati ning võrreldi lahenduseks vajalikke tehnoloogiaid ning nende alternatiive. Valitud tehnoloogiate puhul toodi välja, mis on nende eelised alternatiivide ees. Töö teises osas selgitatati loodud lahendust, kirjeldati lahenduse loomise keskkonna erinevaid võimalusi ning testiti ka rakenduse jõudlust.

Käesoleva bakalaureusetöö tulemusena valmis veebirakenduse prototüüp. Valminud prototüüp võimaldab ettevõtte klientidel ka kodust oma isikliku virtuaalreaalsuse komplektiga sõita mudelautodega Tartus Eedeni kaubanduskeskuses asuval ringrajal. Lahendus ei ole kindlasti veel lõplik ning sellel on palju potentsiaali edasiarenguks. Autor loodab, et ettevõtte kasutab loodud lahendust oma tootmisversioonis.

5.1 Tänuõnad

Täna ettevõtet RCSnail väga läheda võimaluse eest sellist bakalaureusetööd teha ning samuti juhendajaid Rainer Paati ning Raimond-Hendrik Tunnelit, nende hämmastava pühendumuse ning kannatlikkuse eest.

Allikad

[1] Lucile Sassatelli, Marco Winckler, Thomas Fisichella, Antoine Dezarnaud, Julien Lemaire, Ramon Aparicio-Pardo, Daniela Trevisan. “New interactive strategies for virtual reality streaming in degraded context of use”. *Computers & Graphics*, Volume 86, 2020, 27-41, ISSN: 0097-8493. <https://doi.org/10.1016/j.cag.2019.10.005>

[2] Kjell Brunnström, Elijs Dima, Tahir Qureshic, Mathias Johanson, Mattias Andersson, Mårten Sjöströmb. “Latency impact on Quality of Experience in a virtual reality simulator for remote control of machines”. *Signal Processing: Image Communication*, Volume 89, 2020, 116005, ISSN: 0923-5965- <https://doi.org/10.1016/j.image.2020.116005>

[3] Kjetil Raaen, Ivar Kjellmo.”Measuring Latency in Virtual Reality Systems”. *Entertainment Computing - ICEC 2015, 14th International Conference, ICEC 2015, Trondheim, Norway, 2015*. https://dx.doi.org/10.1007/978-3-319-24589-8_40

[4] Miao Hu, Xianzhuo Luo, Jiawen Chen, Young Choon Lee, Yipeng Zhou, Di Wu. “2020 Virtual reality: A survey of enabling technologies and its applications in IoT”. *Journal of Network and Computer Applications*, Volume 178, 15 märts 2021, 102970, ISSN: 1084-8045. <https://doi.org/10.1016/j.jnca.2020.102970>

[5] Clemens Mayer, “First Person View RC Car with PS1 Steering Wheel”. *Element 14 forum* 2.03.2019. <https://www.element14.com/community/docs/DOC-91965/1/episode-391-first-person-view-rc-car-with-ps1-steering-whee?ICID=e14-presents-videos> (12.01.2021)

[6] Federica Pallavicini, Alessandro Pepe, Maria Eleonora Minissi. “Gaming in Virtual Reality: What Changes in Terms of Usability, Emotional Response and Sense of Presence Compared to Non-Immersive Video Games?” *Simulation Gaming*, Volume 5, 136-159, 2019, doi:10.1177/1046878119831420. <https://journals-sagepub-com.ezproxy.utlib.ut.ee/doi/10.1177/1046878119831420>

[9] Jake Tivy, “First Person View RC Car Racing!!”. *Linus Tech Tips forum*, 15.05.2019. <https://linustechtips.com/topic/1064124-first-person-view-rc-car-racing/> (12.01.2021)

Lisad

I. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Mait Lättekivi,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Kaugjuhitavate mudelautode videoülekanne VR lahendus,

mille juhendajad on Raimond-Hendrik Tunnel ja Rainer Paat

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Mait Lättekivi

07.05.2021

II. Kasutatud mõisted

API - rakendusliides, mis on mõeldud tarkvara arendamise lihtsustamiseks.²⁹

raamistik - tarkvaraarenduses mingi kindla ülesande täitmiseks mõeldud lähtekood³⁰

teek - funktsioonide, klasside ja moodulite kogum, mida programmis kasutada saab³¹

silumine - tarkvaraarenduse protsess, mille käigu leitakse ja parandatakse koodis vigu³²

virtuaalreaalsus - arvutiga tekitatav interaktiivne, tegelikkust realistlikult jäljendav v kujutluslik multimeediumkeskkond³³

virtuaalreaalsuspeakomplekt - seadeldis virtuaalreaalsus elamustetega suhtlemiseks.

liitreaalsus - reaalsusse virtuaalsete objektide lisamine, näiteks filtrid, aga ka muud objektid, mis on kuvatud läbi ekraani pärismaailma peale.

küberiiveldus - sümptomid, mis võivad tekkida virtuaalreaalsuse seadmete või konkreetsete rakenduste tõttu. Tavaliselt väljendub ajutise peapöörituse või iiveldusena.³⁴

varjutaja - varjutaja on koodiprogramm, mis käivitatakse graafikakaardis ja on tavaliselt kindla ülesandega graafika standardses töövoos.³⁵

renderdamine - renderdamine on kaadri joonistamine.³⁶

²⁹ <https://et.wikipedia.org/wiki/Rakendusliides>

³⁰ <https://et.wikipedia.org/wiki/Tarkvararaamistik>

³¹ <https://et.wikipedia.org/wiki/Teek>

³² <https://et.wikipedia.org/wiki/Silumine>

³³ <http://kn.eki.ee/?Q=virtuaalreaalsus>

³⁴ <https://eevr.ee/vr-dictionary/>

³⁵ <https://cgvr.cs.ut.ee/wp/index.php/arvutigraafika-terminid/>

³⁶ <https://cgvr.cs.ut.ee/wp/index.php/arvutigraafika-terminid/>