

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Toomas Tamm
DeltaVR
Bakalaureusetöö (9 EAP)

Juhendaja: Raimond-Hendrik Tunnel, MSc

Tartu 2021

DeltaVR

Lühikokkuvõte:

Lõputöö kirjeldab virtuaalreaalsus elamuse DeltaVR loomist ja tehtud disaini otsuseid. Lõputöös on kirjas, kuidas Tartu Ülikooli Delta hoone 3D mudel saadi Unity mängumootorisse ja kuidas 3D mudelit optimeeriti hea jõudluse jaoks. Kirjeldatakse ka erinevate rakendusse loodud virtuaalsete elamuste implementatsiooni. Viimasena sai elamuse jõudlust testitud ning viidi läbi ka kasutajatestimine. Kasutajatestimise tulemusena leiti elamuses parandust vajavaid vigu ja probleeme, mida tuleks parandada. Lisaks leiti veel ideid, mida võiks tulevikus elamusse lisada.

Võtmesõnad: tarkvaraarendus, virtuaalreaalsus, digitaalne kaksik, mängudisain, arvutigraafika

CERCS: P170, arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

DeltaVR

Abstract:

This thesis describes the creation of the virtual reality experience named DeltaVR and the design decisions behind it. It describes how the 3D model of University of Tartu Delta building was put into the Unity game engine and what optimizations were done to make it run well. The thesis contains the implementation of different virtual experiences. The final application was performance and usability tested on users to find issues and bugs with the experience and to find ideas to be improved or added in the future.

Keywords: software development, virtual reality, digital twin, game design, computer graphics

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

Sisukord

1. Sissejuhatus.....	4
2. Virtuaalsed elamused hoonetes.....	6
2.1 PaabelVR	6
2.2 Delta Õppehoone Visualiseerimise projekt	7
2.3 CGVR VR Demo	8
3. Disain	9
3.1 Delta keskus	9
3.2 Elamuses liikumine.....	12
3.3 Virtuaalsed käed.....	13
3.4 Uksed	15
3.5 Elamused.....	15
3.5.1 Vibu lasketiir.....	16
3.5.2 Breakout VR	17
4. Implementatsioon.....	19
4.1 Tehnoloogia	19
4.2 Virtuaalreaalsuse toe lisamine	19
4.3 Virtuaalreaalsuses liikumine.....	20
4.3 Virtuaalsed käed.....	21
4.4 Uksed	22
4.5 Vibu.....	23
4.6 Breakout.....	25
5. Testimine.....	28
5.1 Elamuse jõudlus	28
5.2 Kasutajatestimine	29
5.3 Kasutajatestimise tulemused.....	31
Kokkuvõte.....	36
Viidatud allikad.....	38
Lisad.....	39

1. Sissejuhatus

Virtuaalreaalsus (VR) on tehnoloogia, millega tekitatakse arvuti abil kasutajale taju, et ta on kuskil mujal¹. Esimesed modernsed VR prillide komplektid tulid välja aastatel 2016-2017². Aastast 2017 aastani 2021 on ülemaailmne VR turu tulu tõusnud 0.4 miljardi dollari pealt 1.4 miljardi dollari peale³. Prognooside kohaselt kasvab aastaks 2024 see 2.4 miljardi dollari peale³. Veebruar 2021 oli veidi üle kahe protsendi arvutimängude digipoe Steam⁴ kasutajatel VR prillide komplekt⁵. Seega VR on veel uus tehnoloogia ning selle kasutajabaas on kasvamas.

Lõputöö eesmärk on luua Tartu Ülikooli Delta õppe- ja teadushoonest virtuaalreaalsus elamus nimega DeltaVR. Delta hoone valmis aastal 2020⁶ ning seal õpib ja töötab üle 3200 üliõpilast, õppejõudu ja ettevõtte arendustöötajat⁷. Delta hoones tegutsevad paljud erinevad instituudid, näiteks arvutiteaduse instituut ja majandusteaduskond instituut⁷.

Delta hoones asub arvutigraafika ja virtuaalreaalsuse labor. Laboris saavad Tartu Ülikooli tudengid kasutada, luua arvutigraafika ja arvutimängude projekte ning lõputöid. AVGR laboris on kasutamiseks erinevaid virtuaalreaalsuse peaseadmeid, näiteks OSVR HDK 1.2, Oculus Rift CV1, Oculus Quest, HTC Vive and HTC Vive Pro.⁸

Laboris kasutatakse elamuste demonstreerimiseks HTC Vive Pro peaseadet, seega peamine elamuse sihtseade on HTC Vive Pro. Kuna autoril oli kodus ainult Oculus Rift S peaseade, siis on arendamise jaoks ka sellele tugi. HTC Vive Pro ja Oculus Rift S peaseadmetel on erinevaid puldide mõlemale käele. Elamus on mõeldud nii VR kogemusega kui ka kogemuseta inimestele. Elamuse eesmärk on tõsta huvi inimestes virtuaalreaalsuse vastu.

Peatükis 2 vaadatakse sarnaseid töid ja elamusi hoonetes, vaadates, mida nendes on tehtud hästi ja halvasti. Peatükis 3 ja 4 kirjutatakse elamuse disaini lahendusi ja virtuaalse hoone elamuste

¹ <https://www.wired.com/story/wired-guide-to-virtual-reality/>

² <https://www.vrs.org.uk/virtual-reality/history.html>

³ <https://www.statista.com/statistics/499714/global-virtual-reality-gaming-sales-revenue/>

⁴ <https://store.steampowered.com/about>

⁵ <https://www.statista.com/statistics/265018/proportion-of-directx-versions-on-the-platform-steam/>

⁶ <https://eik.ut.ee/portfolio/deltamaja/>

⁷ <https://delta.ut.ee/koostoo/avasta-delta/>

⁸ <https://cgvr.cs.ut.ee/wp/index.php/people/>

lahendusi. Peatükis 5 viidi läbi jõudlus- ja kasutajatestimine. Kasutajatestimisel koguti tagasisidet elamuse meeldivuse kohta ning mida võiks veel täiendada.

Lisa I on sõnastik. Lisa II on töö loomise ajal kasutatud masinad ja nõuded. Lisa III on pilte valmis rakendusest. Lisa IV on kasutajajuhend, kuidas rakendust käima panna ning mõned tõrkeotsingu sammud. Lisa V on kirjeldus lõputööga kaasa tulevate failidest.

Rakenduse repositoorium asub aadressil <https://gitlab.toomastamm.ee/toomastamm/delta-base>.

2. Virtuaalsed elamused hoonetes

Sarnaste tööde analüüsimine on oluline, et vältida sama töö uuesti tegemist. Lisaks sarnased töid analüüsid on võimalik leida vigu, mida uues elamuses vältida, ning uusi ideid, mida elamusse lisada. Järgmises kolmes alapeatükis vaatleme kolme digitaalsete kaksikute projekti, mida AGVR tudengid seni teinud on.

2.1 PaabelVR

PaabelVR⁹ versioon kaks oli aastal 2018 Tartu Ülikooli aine Arvutigraafikas loodud projekt, mille raames tehti Paabeli õppehoones olnud AGVR labori ruumidest virtuaalreaalsuse elamus (vt Joonis 1). Projektis tehti modelleerimistarkvaraga Blender¹⁰ labori ruumist Paabeli õppehoones digitaalne kaksik, kasutades nii käsitsi modelleeritud varianti kui ka fotogrammeetriaga tehtud mudelit. Elamusse lisati veel mööblit ja tegevusi, näiteks õhupallide puhumine, seina ja põranda kaotamine kosmose näitamiseks, projekti juhendaja Madis Vasseri 3D skaneeritud pea klaaspurki lisamine kahe erineva näoilmega ja muud sellist.



Joonis 1: Pilt PaabelVR elamusest.

Kuna aastal 2019 kolis AGVR labor Delta hoonesse¹¹, siis PaabelVR ei paku enam kasutajale sujuvat üleminekut virtuaalsesse maailma. Lisaks esines autoril PaabelVR rakendust testides

⁹ <https://courses.cs.ut.ee/2018/cg/fall/Main/Project-PaabelVR-v2>

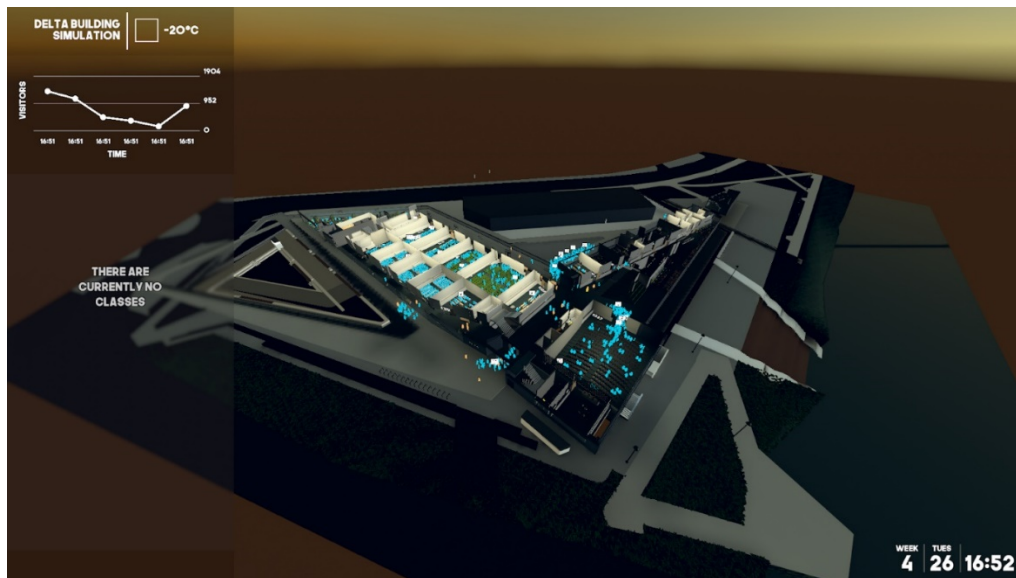
¹⁰ [https://et.wikipedia.org/wiki/Blender_\(tarkvara\)](https://et.wikipedia.org/wiki/Blender_(tarkvara)).

¹¹ <https://cgvr.cs.ut.ee/wp/index.php/news/>

jõudlusega probleeme ning projektil puudub tugi muudele virtuaalreaalsuse peaseadmetele kui HTC Vive. Kuna laboris on palju erinevaid seadmeid, võiks elamus toetada rohkem kui ühte seadet.

2.2 Delta Õppehoone Visualiseerimise projekt

Delta Õppehoone Visualisatsioon¹² (DBV) on arenduses olnud aastast 2017 ning selle eesmärk on visualiseerida inimeste liikumist Delta keskus kasutades hoones olevaid sensoreid (vt Joonis 2). Projektis on loodud hoone esimesest kahest korrusest reaajas renderdamiseks optimeeritud mudel. Lisatud on Delta hoones kahel esimesel korrusel reaalsusele vastav mööbel (Leesment, 2020, lk 13). Lisaks on veel efekte Delta õppehoone ümbruse ja ilmastiku näitamiseks ja muud sellist (Linde, 2019, lk 21-23) (Voitenko, 2018, lk 25-33).



Joonis 2: Pilt DBV projektist.

DBV projektiga on DeltaVR sarnane selle poolest, et mõlemal on eesmärk jälgendada Delta õppe- ja teadushoonet mängumootoris Unity¹³. Siiski ei ole DBV projekti 3D mudel niisama kõlbulik virtuaalreaalsuses kasutamiseks. Selles on puudu laed ja lambid, seinad on lõigatud pooleks, et parandada ülevalt alla vaatamist (Nikolajev, 2018, lk 17). Lisaks on paljudes seintes augud, mis on nähtavad virtuaalreaalsuses esimeses isikus.

¹² <https://dbv-sc.weebly.com/>

¹³ <https://unity.com/unity/features/vr>

2.3 CGVR VR Demo

CGVR VR Demo¹⁴ on 2020 aasta aine arvutigraafika projekt (MTAT.03.328), mille raames tehti Delta keskusest VR demo, keskendudes peamiselt AVGR labori ruumi 2007 jäljendamisele (vt Joonis 3).



Joonis 3: Pilt CGVR VR Demo elamusest.

Mängumootoris Unreal Engine¹⁵ implementeerida VR tugi, käed ja õhupallide loomine. Demos on võimalik ringi liikuda kogu Delta keskuses. Suureks puuduseks on, et väljaspool AVGR labori 2007 ruumi puudub mööbel. Proovides jooksutada projekti enda arvuti peal (vt Lisa II: Nõuded riistvarale) oli kaadriaeg üle 11,1 ms. Seega rakendusel puudub virtuaalreaalsuses sujuva elamuse jaoks piisav jõudlus.

Seega Delta hoone ja AVGR labori kohta on tehtud erinevaid digitaalseid kaksikuid ja virtuaalreaalsuse kogemusi. Siiski on puudu ajakohane virtuaalreaalsuse elamus, kus on võimalik avastada suuremat osa sisustatud Delta hoonest. Lisaks peab uuem elamus käima parema jõudlusega, kui varasemad rakendused.

¹⁴ <https://courses.cs.ut.ee/2020/cg/fall/Main/Project-CGVR-VRDemo>

¹⁵ <https://www.unrealengine.com/en-US/>

3. Disain

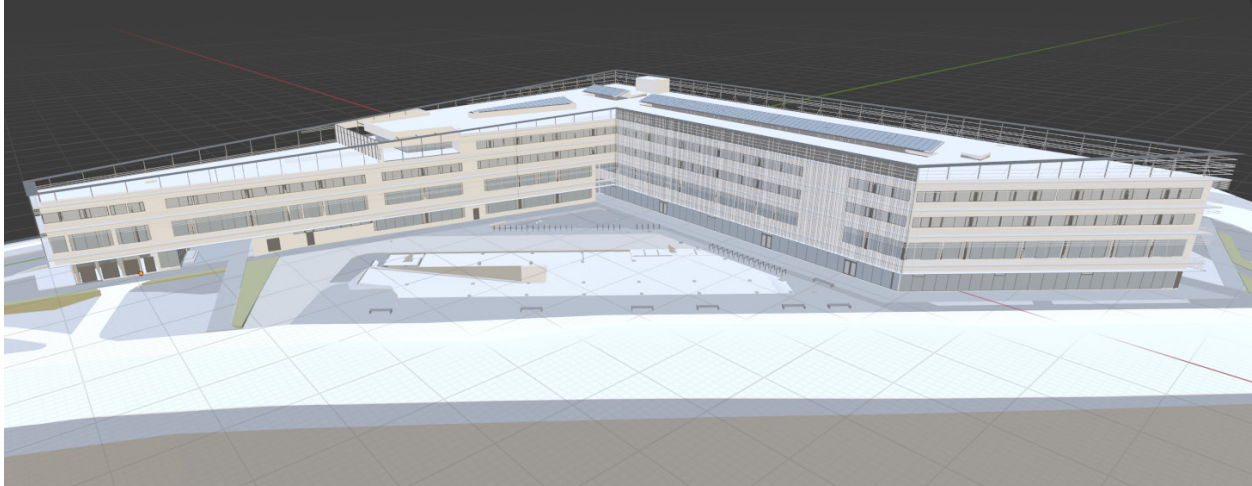
DeltaVR elamuse loomise ajal lähtuti *iterative game design* protsessist, kus mängu korduvalt prototüübitakse, testitakse ja analüüsitakse nii tihti, kui võimalik. Selle tulemusena areneb projekt üle aja, ning vead leitakse varakult üles (Zimmerman, 2003). Selles peatükis kirjutatakse Delta hoone digitaalse kaksiku ja selle elamuste disainist. Alampeatükis 1 kirjutatakse, kuidas hoone virtuaalne 3D mudel saadi ja seda kohendati. Alampeatükkides 3.1-3.4 kirjutatakse, kuidas elamusega saab interakteeruda. Alampeatükis 3.5 kirjutatakse hoonesse lisatud mängude kohta.

3.1 Delta keskus

Delta keskuse õppehoone digitaalseks jälgendamiseks leiti kaks varianti. Esimene nendest oli arhitektuuribüroo Arhitekt11 antud IFC formaadis fail, mis sisaldab hoone mudelit, selle materjale ja jooniseid. Selle faili kasutamise plussiks on see, et ta on hästi detailne ning sisaldab kõiki Delta keskuse korruseid. Halb on fail sellepärast, et hoone ehituse jooksul paljud detailid muutusid, ning faili sisu ei ole täpne tegelikkusega. Näiteks IFC failis on Delta esimese korruse trepid punased, aga tegelikkuses on need rohelised. Lisaks on IFC fail mõeldud projekteerimistarkvaras kasutamiseks, mitte mängudes. Selle tõttu puudub populaarsemates mängumootorites selle jaoks tugi.

Faili mängumootorisse saamiseks tuleks see teisendada mõnda teisse formaati. Teisendamiseks saab kasutada vabavara IfcOpenShell¹⁶. See on võimeline faili teisendama mitmesse erinevasse enim toetavasse formaati. IFC formaadis fail teisendati FBX formaati, aga saadud failis on ligi 20 000 unikaalset võrestikku ning nende renderdamisel on kaadriaeg üle 11,1 ms. Seetõttu tuleks võrestikke väga palju optimeerida. Lisaks puudub tulemil suur osa mööblist ning paljud materjalid on tegelikkusega võrreldes valed (vt Joonis 4). Selle probleemiga on tegelenud teised tudengid mitu aastat DBV projektis (Voitenko, 2018) (Nikolajev, 2018) (Linde, 2019) (Leesment, 2020), seega selle failiga edasi töötades oleks töö maht liiga suur.

¹⁶ <http://ifcopenshell.org/>



Joonis 4: Virtuaalse hoone välisvaade pärast IFC faili teisendamist.

Teine variant oli Delta Õppehoone Visualiseerimise projekt. Kuna DBV projektis on juba suur osa hoone mudelist mängumootoris kasutamise jaoks optimeeritud ja Leesmenti leitud tulemuste põhjal on see sarnane päriseluga, siis otsustati võtta DeltaVR elamuse aluseks DBV projekti 3D mudel (Leesment, 2020, lk 29).

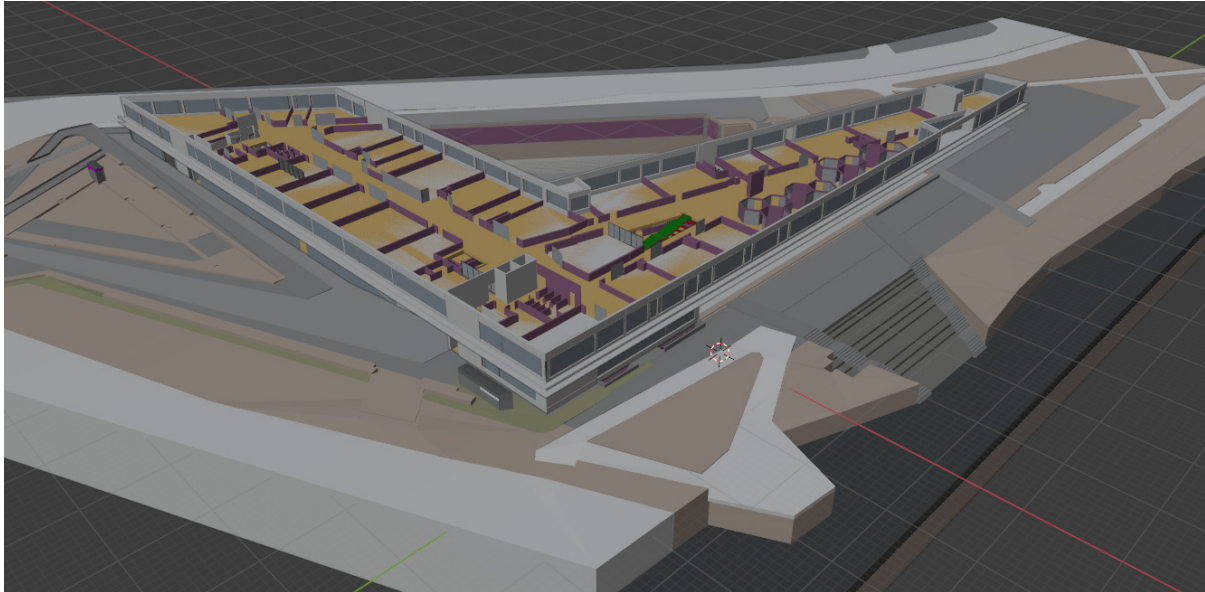
DBV projekti autoritelt saadi alusfail, kus oli suur osa DBV visualiseerimise skripte välja võetud ja jäetud alles projektifaili hoone erinevad jupid. Alusest pidi eemaldama veel üleliigseid skripte. Peale selle oli DBV aluse virtuaalreaalsuses renderdamisel kõrgem kaadriaeg kui lubatud miinimum 11,1 ms (vt Lisa II: Nõuded riistvarale).

Jasoni sõnul võib kõrge kaadriaeg põhjustada küberiiveldust (Jerald, 2015, lk 183-184). Kaadriaeg on kõrge seetõttu, et võrreldes DBV projektiga, peab DeltaVR renderdama elamust virtuaalreaalsuse peaseadmes kahe suurema resolutsiooniga ekraani peale korraga. Lisaks on autori arvuti riistvara nõrgem kui see, mille peal DBV projekti testiti.

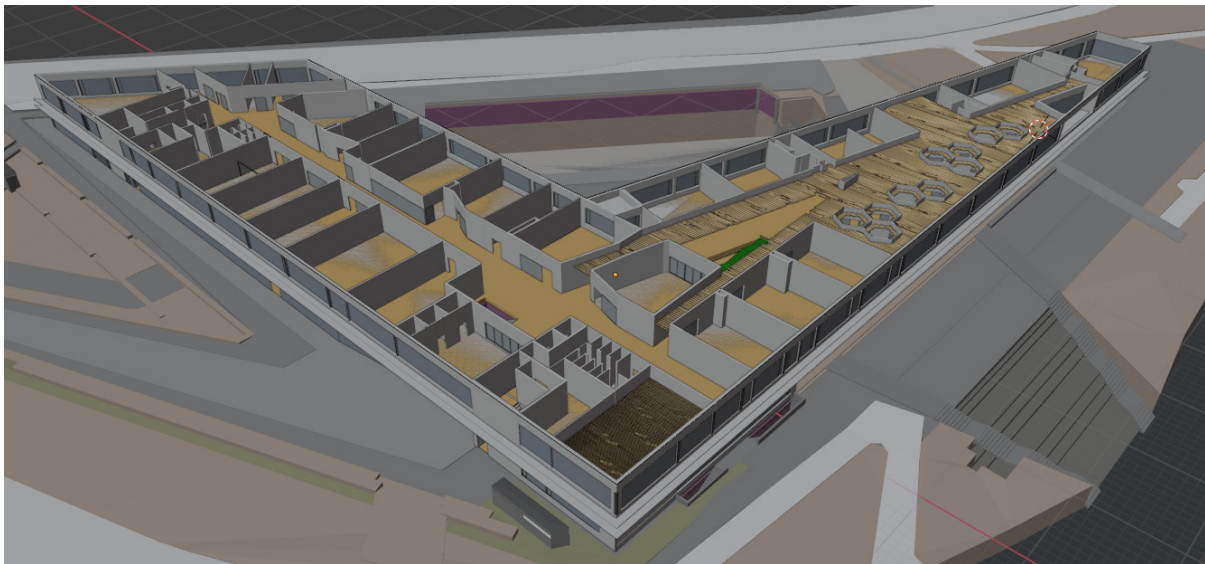
Üks viis vähendada kaadriaega on mitte renderdada seda, mida kasutaja ei näe, näiteks seinatagune mööbel ehk varjatu praakimine (ingl *occlusion culling*). Kuna hoone võrestikus on hästi palju väikeseid vahesid ja auke, on raske seinatagust mööblit korrektselt välja praakida. Lisaks oli empiirilisel näha, et augud halvendavad elamuse kaasatust.

Kuna DBV projektis tehti Delta keskuse mudel programmis Blender ning modelleerimisprogrammi fail tuli projekti alusfailiga kaasa, oli paranduste tegemine lihtne. Esmalt sai lisatud teisele korrusele lagi. Seejärel kõik seinad esimesel ja teisel korrusel said pikendatud korruse laeni

või järgmise korruse põrandani. Samamoodi said täidetud mõlemal korrusel uste kohal olevad seinad. Seejärel suutis mängumootor seinatagust mööblit välja praakida ning saavutati nauditava elamuse jaoks maksimaalne kaadriaeg 11,1 ms.



Joonis 5: DBV teine korrus.



Joonis 6: DeltaVR teine korrus.

Joonis 5 ja Joonis 6 vahel on näha, et on tõstetud kõik seinad kõrgemaks ning lisatud tagasi Delta hoonele omased laes olevad ribad. Kuna 3D mudelil puudub kaugem ümbrus, sai taustaks valitud Unity poest¹⁷ leitud kosmose taust (vt Lisa III: Pildid rakendusest).

3.2 Elamuses liikumine

Virtuaalreaalsuse (VR) elamustes on oluline hoiduda tekitamast kasutajale küberiiveldust, et tagada nauditav katkestusteta kogemus. VR elamustes on kasutusel erinevad liikumisviisid ja nendel on erinevad mõjud kasutaja kogemusele (Jerald, 2015, lk 163). Mayor jt. analüüsisid nelja erinevat liikumisviisi: huvipunktidesse liikumine, puldiga juhtimine, tehishüppeid ja ruumiulatuses liikumine. Nende järgi saavutasid kõige väiksema iivelduse ruumiulatuses liikumine ja tehishüppamine, kuigi viimane on vähem intuitiivsem liikumisviis. Selle järel tulid puldiga juhtimine ja huvipunktidesse liikumine. Mayor jt. mainivad lisaks, et ruumiulatuses navigeerimine ja tehishüppamine sobivad hästi kokku. Nende arvates on ruumiulatuses navigeerimine kõige parem liikumisviis nii elamuse saamise kui ka iivelduse vältimise poolest (Mayor, Raya, & Sanchez, 2019). Clifton jt. said sarnased tulemused. Nad lisasid, et kuigi puldiga juhtimine tekitab rohkem iiveldust, olid kasutajad rohkem mängu süvenenud võrreldes tehishüppamisega. Samuti on inimesed vähem süvenenud elamusse istudes kui seistes. Cliftoni jt. arvates on siiski raske leida universaalset VR liikumisviisi, mis sobiks kõikidele inimestele nii elamuse kui ka iivelduse poolest (Clifton & Palmisano, 2019).

DeltaVR elamus on mõeldud peamiselt virtuaalreaalsuse uutele katsetajatele, seetõttu eelnevast võib järeldada, et parim ja kõige universaalsem kombinatsioon on ruumiulatuses liikumine koos tehishüppamisega. Tehishüppamise jaoks peab olema kasutajal võimalus sihtkohta valida. Selle jaoks sai loodud puuteplaati ülaosa vajutades või juhtkangi üles lükkamisel välja tulev joon koos sihtmärgiga (vt Joonis 7).



Joonis 7: Tehishüppe sihtmärk.

¹⁷ <https://assetstore.unity.com/packages/2d/textures-materials/sky/skybox-volume-2-nebula-3392>

Veel tuleb lisada eraldi elamuses pööramise viis. Mõlemat lõppseadet on võimalik ühendada arvutiga kaabliga, ning liiga palju füüsiliselt pöörates võib kaabel kasutaja jalgade külge kinni jääda. Kuna esimesest lõigust võib järeldada, et kasutaja sujuv liigutamine võib tekitada iiveldust, siis elamusse sai valitud kohene 45-kraadine pööre vasakule või paremale, vastavalt puldi juhtnupu keeramisel või puuteplaadi vajutamisel.

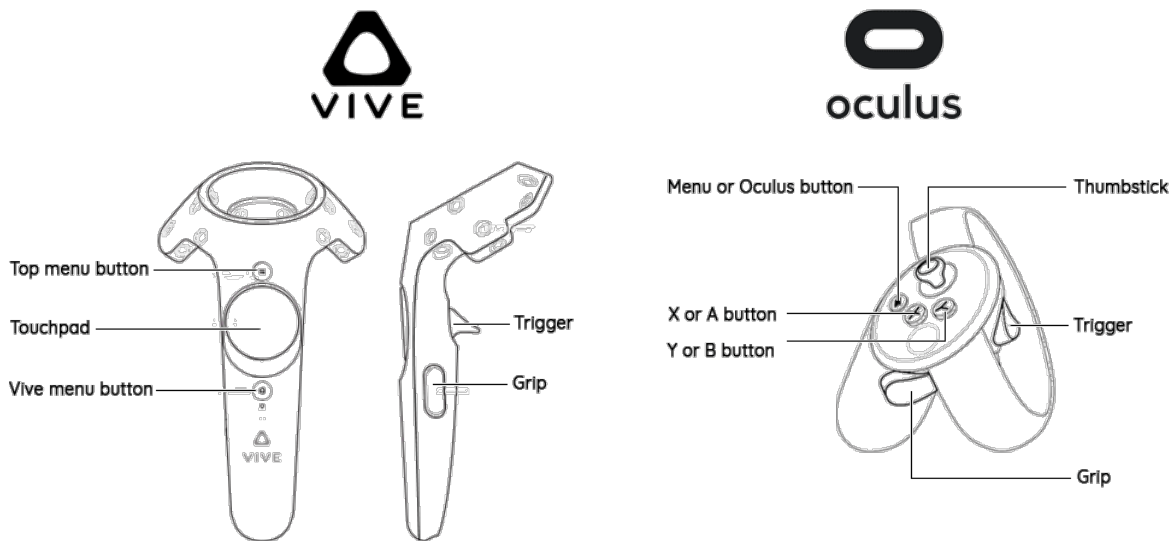
3.3 Virtuaalsed käed

Virtuaalreaalsuse elamuste üks suur oluline osa on kasutaja interaktsioon keskkonnaga. Jerald leidis, et virtuaalsed käed on kõige parim variant loomuliku interaktsiooni jaoks. Ta loetles kolme erinevat viisi, kuidas käsi lisada: 1) realistlik variant koos käte ja käsivartega 2) osati realistlik variant ainult kätega 3) abstraktsed käed, kus kätt saab liigutada väljapoole isiklikku ruumi. Nende hulgast ta ütleb, et osati realistlikke kätega saab kasutaja kõige kergemini ümbrust mõjutada (Jerald, 2015, lk 323-326). Seega DeltaVR elamusse on kõige sobivam lisada osati realistlikud käed. Lisaks on selle realiseerimine variantide hulgast kõige lihtsam.

Kaasatuse suurendamiseks lisati ka kätele erinevad animatsioonid vastavalt sellele, kuidas kasutaja oma käsi nuppudel hoiab (vt Joonis 8). Nendeks sai valitud avatud käsi, rusikas ja näpistamine ehk ok-positsioon. Kuna elamus peab toetama kahte erinevat pulti, peavad mõlema puldi nupud panema käe õigesse positsiooni. Kuna nupud asuvad kontrolleritel piisavalt samades kohtades, võib need panna sama nupu külge (vt Joonis 9). Kui puldil ei ole nuppe alla vajutatud, on käsi avatud positsioonis. Kui puldil on alla vajutatud *trigger* nupp, siis läheb käsi näpistamise positsiooni. Kui puldil on alla vajutatud *grip* nupp, on käsi rusika positsioonis.



Joonis 8: Erinevad käe positsioonid DeltaVR elamuses.



Joonis 9: Vive Wand (vasakul) ja Oculus Touch (paremal) pultide nupud.¹⁸

Algselt sai tehtud nii, et kui kasutaja vajutab *grip* nuppu ning käsi läheb rusikasse, siis võtab virtuaalne käsi objektist kinni ning hoiab seda. Kuna Vive Wand kontrolleriil on autori testimisel väga ebamugav *grip* nuppu all hoida, siis sai muudetud nii, et kasutaja saab ka asju üles võtta kasutades *trigger* nuppu.

¹⁸ <https://vrsketch.eu/docs-drawing.html>

3.4 Uksed

Delta hoones erinevate ruumide liikumisel tuleb läbi liikuda ustest. Selle jaoks on kaks erinevat lahendust. Esimene on kasutaja teisele poole tehishüppamine, mis on kasutuses näiteks sarnases töös CGVR VR Demo. Teine variant on kasutajal ise uks lahti teha ja läbi kõndida. Elamusse sai valitud teine variant, sest see suurendab elamuse kaasatust.



Joonis 10: 1) Pärisklassi uks 2) DBV uks 3) DeltaVR uks.

Lisaks ei klapi DBV baasiga kaasa tulnud uksed päriseluga. Seega algul tuli neid parandada, muutes külgsuurt paksusi ning lisades paremaid materjale (vt Joonis 10). Kui võtta kinni ukse hingid, on võimalik ust kinni ja lahti tõmmata. Seetõttu tuli leida hoonest vanad uksed ning need uuema versiooniga välja vahetada.

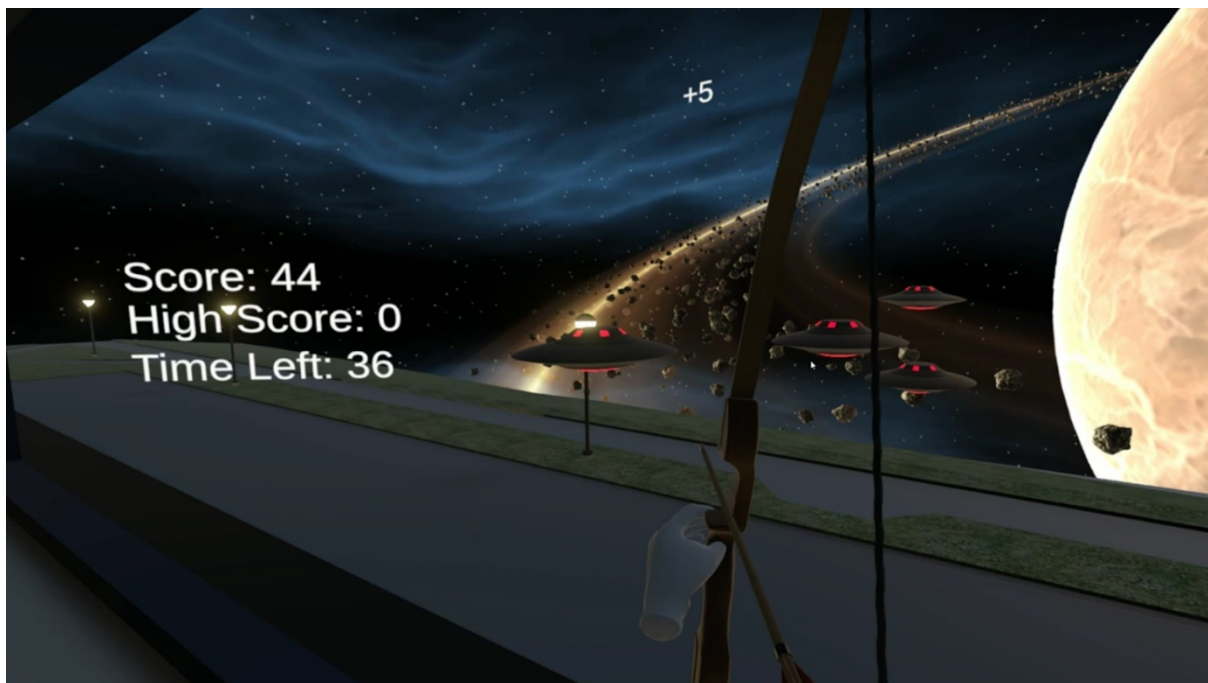
3.5 Mängud

Mängud on väiksemad iseseisvad kogemused Delta hoones. Nende eesmärk on pakkuda kasutajale muud tegevust peale hoonest ringi liikumise. Lisaks näitavad nad, et virtuaalreaalsuses on võimalik teha tegevusi, mida päriseluses on keeruline saavutada. Kuna DeltaVR on ka mõeldud uutele VR kasutajatele, peavad elamused olema mugavuselt lihtsad.

3.5.1 Vibu lasketiir

Esimene mäng on vibu lasketiir, mis sai loodud teise korruse rõdule. Rõdule on asetatud vibu ning lõpmatu kogus nooli. Hoides vibu ühes käes ning pannes noole vibule, saab vibu tagasi tõmmata. Lahti lastes lendab nool välja.

Lasketiirus sihtmärkide tekitamiseks peab mängija laskma paigal seisvat “Start” sihtmärki. See tähendab, et liikuvad sihtmärgid tekivad alles siis, kui kasutaja on õppinud vibu kasutama. Kui kasutaja on Start sihtmärgile pihta saanud, algatab lasketiir üheminutilise taimeri ning hakkab liikuvaid sihtmärke kaugusesse tekitama (vt Joonis 11).



Joonis 11: Pilt vibu lasketiiru elamusest.

Lasketiiru liikuvateks sihtmärkideks on Delta hoonele lähenevad UFO-d (vt Joonis 12), mis valivad suvalise punkti rõdu juures ning liiguvad sinnapoole. Sihtmärkidele pihta laskmine lisab mängijale punkte sõltuvalt sihtmärgi kaugusest.



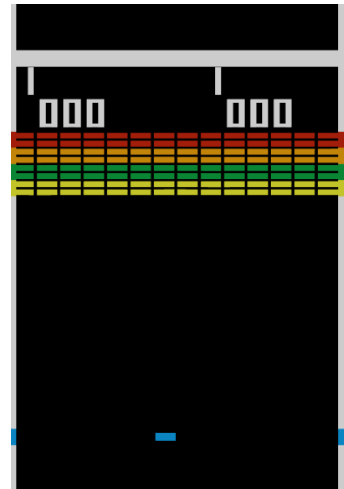
Joonis 12: UFO.

Kui taimer saab täis, näidatakse kasutajale tema skoori, kõige kõrgemat skoori ning tekib tagasi “Start” sihtmärk.

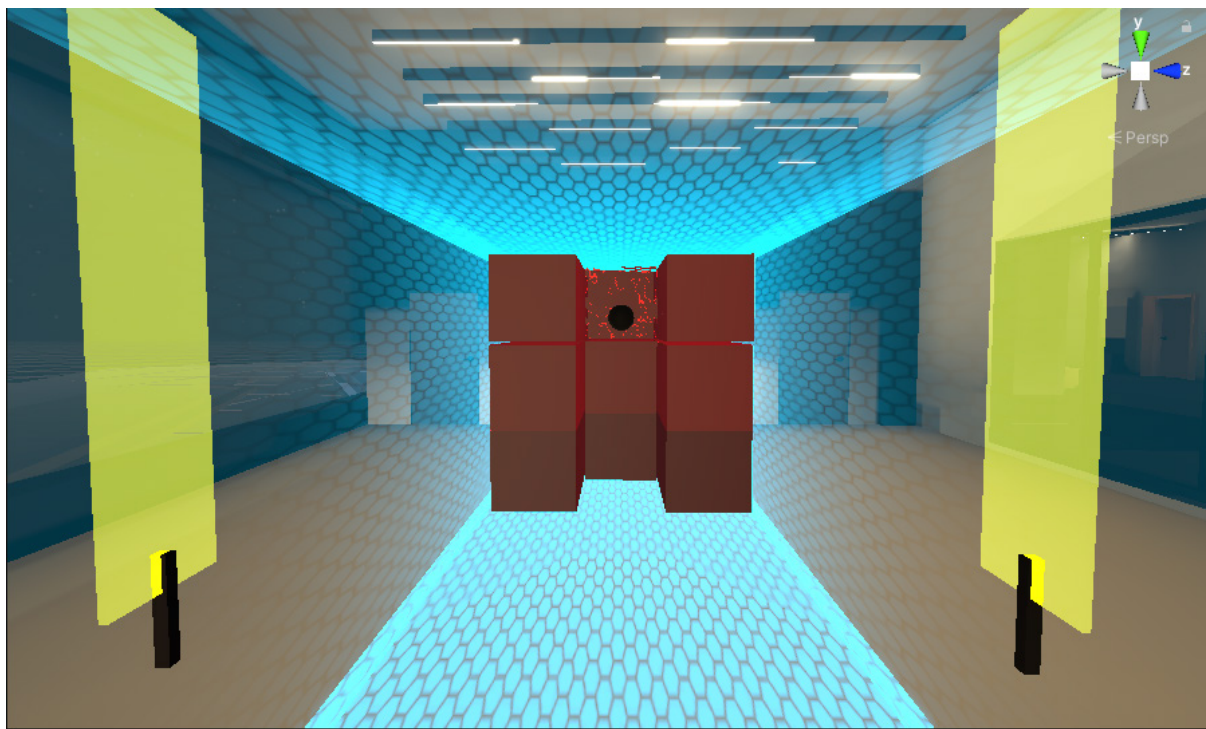
3.5.2 Breakout VR

“Breakout” on klassikaline 2D mäng, kus kasutaja peab labaga palle vastu kaste põrgatama (vt Joonis 13) (Mike, 2016). Breakout VR on inspireeritud kloon sellest mängust, mis kasutab sarnaseid mehaanikaid. Võrreldes 2D variandiga, on kolme põrgatava seina asemel viis, ühe laba asemel on kaks laba, ning kastid on 3D ruumis.

Mängus saab kasutaja hoida mõlemas käes ühe laba, millega palli lüüa (vt Joonis 14). Eesmärk on lõhkuda mängijale lähenevaid kaste. Kui kastid liiga lähedale jõuavad, on mäng läbi. Kui pall läheb mängijast mööda vastu tagumist seina, siis saab mäng samuti läbi.



Joonis 13: Pilt "Breakout" mängust



Joonis 14: Breakout VR mäng

Kaste on kahte tüüpi. Esimene kasti tüüp on tavaline punane kast, mille hävitamisel kaob ta niisama ära. Teine kasti tüüp on roheline kast, mis hävitamisel tekitab ühe palli juurde. Rohkemate pallidega saab kiiremini kaste hävitada, aga peab rohkem palle jälgima. Rohelise kasti tekkimise tõenäosus on 5%.

Kastid tekivad lainetes ning igas laines on 9 kasti. Uus laine tekib iga viie sekundi tagant. Kastid liiguvad mängija poole 0,15 meetrit sekundis. Uute lainete tekkimise ja kastide liikumise kiiruse vahe tõttu on lained üksteisega tihedalt koos. Mida lähemal on lained kasutajale, seda kiiremini saab kasutaja põrgatada palli vastu kaste, ning seda kiiremini lained hävinevad.

4. Implementatsioon

Implementatsiooni peatükis on täpsemalt kirjas, milliseid tehnoloogiaid kasutati elamuse loomiseks ning kuidas erinevaid lahendusi implementeeriti. Kuigi implementeerida tuli palju asju, siis siin peatükis on kirjeldatud ainult huvitavamad ja keerulisemad asjad. Lisaks sai palju korrastatud hoone mudelit, et valguse küpsemine tuleks hea välja (vt Lisa III: Pildid rakendusest).

Alampeatükis 4.1 on kirjas, milliseid tehnoloogiaid kasutati DeltaVR loomisel ja miks. Alampeatükis 4.2 on kirjas, kuidas lisati mängumootorile VR tugi. Alampeatükis 4.3 on kirjas, kuidas implementeeriti elamuses liikumine. Alampeatükis 4.4 on kirjas, kuidas animeeriti virtuaalsed käed. Alampeatükis 4.5 on kirjas, kuidas implementeeriti virtuaalsed ukсед. Alampeatükis 4.5 ja 4.6 on kirjas, kuidas implementeeriti vibu ja breakout mängud.

4.1 Tehnoloogia

Arvutimängude loomiseks on võimalik valida erinevaid mängumootoreid. DeltaVR elamuse jaoks tuli valida mängumootor, millel on VR tugi. VR toega mängumootorid on näiteks Unity, Unreal Engine ja Godot¹⁹. Nende hulgast sai valitud Unity, sest autoril on varasem kogemus tarkvaraga, tarkvara on tasuta, ning peatükis 3.1 oli DBV projekti autoritelt juba saadud Unity mängumootoris Delta hoone alusfail. Lisaks on Unity populaarne mängumootor, seega probleemide esinemisel leidub internetist abi.

Kuna DBV Delta hoone mudelil oli puudujääke, oli vajadus seda redigeerida. Hoone mudeli muutmiseks sai valitud 3D-graafika rakendus Blender, kuna see toetab DBV alusfaili formaati ning on vabavara.

4.2 Virtuaalreaalsuse toe lisamine

Unity mängumootoris on saadaval palju erinevaid virtuaalreaalsuse toe pakette, näiteks Unity XR Interaction Toolkit²⁰, Oculus Integration²¹ ja SteamVR²². Kuna elamus peab toetama erinevaid virtuaalreaalsuse peaseadmeid, siis selleks sobivad paketid on XR Interaction Toolkit ja SteamVR.

¹⁹ https://docs.godotengine.org/en/3.1/tutorials/vr/vr_starter_tutorial.html

²⁰ <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.10/manual/index.html>

²¹ <https://developer.oculus.com/documentation/unity/unity-utilities-overview/>

²² <https://store.steampowered.com/steamvr>

Oculuse seadmetele lisab SteamVR pakett kaadriaega ning ei tööta korralikult Oculuse pakutud silumistööriistadega. Oculuse silumistööriistadega saab Oculuse seadmetel näiteks kaadriaegu mõõta ning *asynchronous spacewarp* meetodit kinni lülitada²³. Seega DeltaVR elamusele sai valitud Unity XR Interaction Toolkit pakett.

Lisaks toetab valitud pakett uuemat OpenXR²⁴ standardit, mis teeb erinevate seadmete toe lisamise kergemaks.

Peale paketi allalaadimist ja projekti lisamist tuleb Unity stseeni lisada paketiga tulnud XR Rig komponent, mis sisaldab oma struktuuris nii pea kui ka käte komponente. Peale seda oli olemas tugi virtuaalreaalsuses ringi vaatamisele ning pea liigutamisele.

4.3 Virtuaalreaalsuses liikumine

Peatükis 3.2 sai otsustatud, et DeltaVR elamuse liikumisviisid on ruumiulatuses liikumine ja tehishüppamine. Ruumiulatuses liikumise tugi on juba olemas tänu eelnevas alapeatükis 4.2 XR Rig mängukomponenti lisamisele. Tehishüppe tugi tuleb eraldi lisada.

Tehishüppe jaoks pidi määrama pinnad, kuhu on võimalik hüppata. Kuna hoone tuli ühe suure võrestikuna, tuli põrandad Blenderis eraldada. Kuna põrandatel on erinev materjal kui seinadel, siis sai kasutada Blenderis *Split by Material* valikut, mis eraldab ühe võrestiku mitmeks erinevaks väiksemaks võrestikuks, kus igal uuel võrestikul on ainult üks materjal.

Seejärel tuli lisada uutele põrandade võrgustikele `Teleport Area` komponent. Kui komponent saab valiku käsu, siis ta liigutab XR Rig objekti valiku asukohta. Paremale käele sai lisatud XR Ray Interactor komponent ning seejärel oli võimalik parema käe teleport nupuga kiirt välja lasta.

Lisaks peab igale `Teleport Area` komponendile määrama sihtmärgi kokkupanek objekt (ingl *prefab*), mis on Unity mootoris erinevate objektide ja nende komponentide korduvkasutatav kogum. Kokkupanek objekt tekib sihitud kohta, kus eelnevalt lisatud kiir ja `Teleport Area` komponendiga võrestik lõikuvad (vt peatükk 3.2). Seejärel kui kasutaja laseb teleport nupust lahti ning on olemas lõikumispunkt, liigutatakse kasutaja lõikumispunktile.

²³ <https://developer.oculus.com/documentation/native/pc/dg-debug-tool/>

²⁴ <https://www.khronos.org/openxr/>

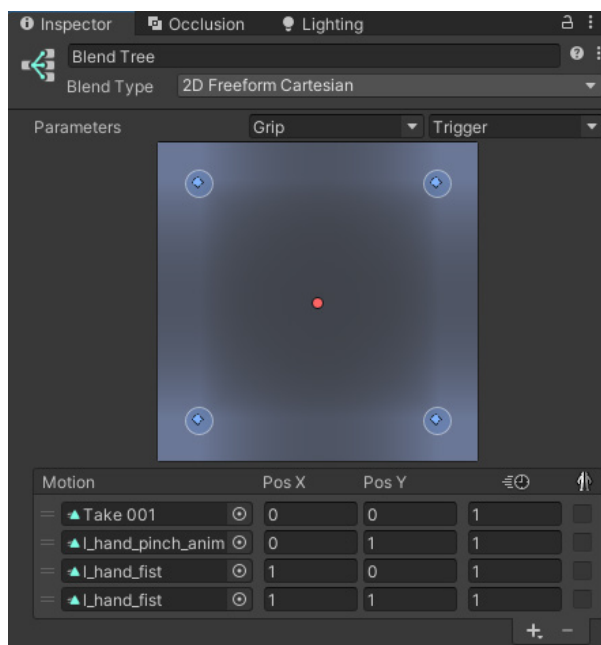
4.4 Virtuaalsed käed

Kasutaja käte positsiooni jälgib esimeses alapeatükis lisatud XR Rig komponent. Aga kuna enamustel VR peaseadmetel on käte jaoks mõeldud puldid, ei ole võimalik täpset näppude positsiooni kätte saada. Siiski on võimalik teha eeldus kasutaja näppude positsiooni kohta kasutades infot puldi nuppudest.

Esiteks tuli leida sobiv käte mudel ja animatsioonid. Kuna XR Interaction Toolkit neid lõputöö tegemise ajal ei sisaldanud, siis sai need võetud Oculus Integration paketest, sest autoril on varasem kogemus selle paketiga. Nagu peatükis 3.3 kirjeldatud, sai kätega kaasa kolm animatsiooni: avatud käsi, rusikas ja näpistamine. Käte mudelitest sai loodud kokkupakitud objekt.

Teine samm oli animatsioonide vahetamine. Selle jaoks on mängumootoris Unity olemas Animator Controller²⁵ komponent. Tuli luua mõlema käe kokkupakitud objektile eraldi Animator Controller ning lisada neile parameetrid *Grip* ja *Trigger*, millele saab hiljem anda puldi nuppude sisendid. Pärast tuli lisada Animator Controller komponendile Blend Tree²⁶, millega saab erinevaid animatsioone kokku sulandada (vt Joonis 15). Blend Tree komponendis saab määrata X ja Y sisendit ning nendeks määrati eelnevalt tehtud parameetrid *Grip* ja *Trigger*.

Blend Tree komponenti lisati veel kolm eelnevalt nimetatud animatsiooni. Pärast nende lisamist saab määrata, kuidas neid kokku sulandada. Sai tehtud nii, et kui *Grip* ja *Trigger* ei ole alla vajutatud, siis on käsi avatud. Kui on *Trigger* alla vajutatud, on käsi näpistamise positsioonis. Kui on kas *Grip* või nii *Grip* kui ka *Trigger* alla vajutatud, siis on käsi rusika positsioonis.



Joonis 15: Blend Tree Unity inspektoris.

²⁵ <https://docs.unity3d.com/2020.2/Documentation/Manual/class-AnimatorController.html>

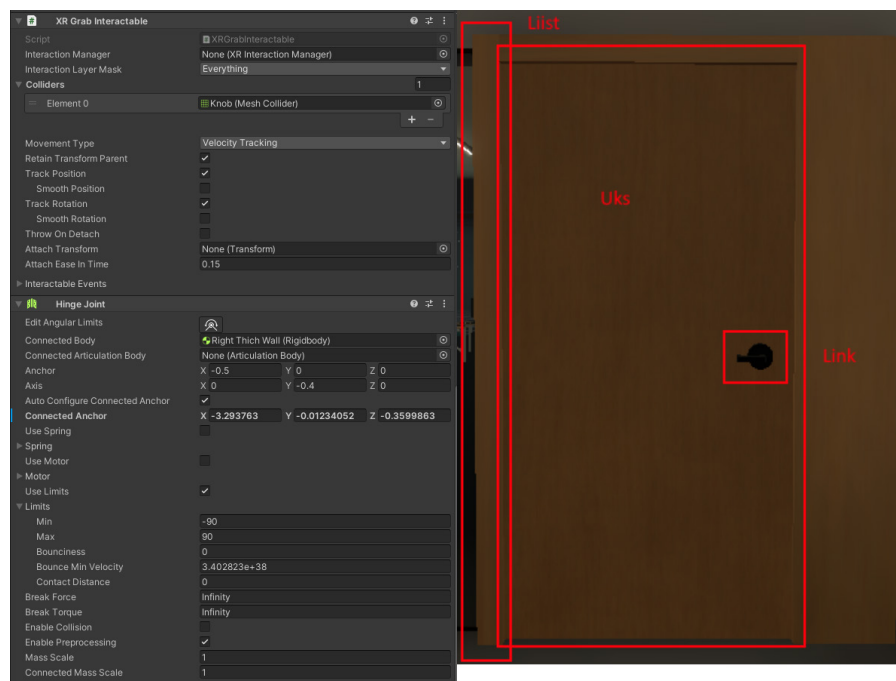
²⁶ <https://docs.unity3d.com/2020.2/Documentation/Manual/class-BlendTree.html>

Kolmandana tuli saata puldi nuppude info Animator Controller komponendile. Selle jaoks tuli luua uus skript mõlemale käele. Skripti sees tuli leida käe Animator Controller komponent, leida sisendseadmete hulgast parem ja vasak käsi ning saata mõlema käe *Grip* ja *Trigger* nappu staatus Animator Controller komponendi *Grip* ja *Trigger* parameetritele.

Viimasena tuli lisada XR Rig komponendile mõlema käe kokkupakitud objekt ning pärast seda olid olemas virtuaalsed käed, mis reageerisid kasutaja näppude positsioonile puldil.

4.5 Uksed

DeltaVR elamuses said uksed tehtud kolmest osast: uks ise, ukse link ja hingega ühendatud ukseliist (vt Joonis 16). Uks ja ukseliist ühendati komponendiga Hinge Joint (vt Joonis 16). Komponendis tuli määrata ühendatud kehaks (ingl *connected body*) ukseliist. Lisaks tuli määrata ukse pööramispiirid (ingl *limits*), milleks said -90 kuni 90 kraadi ning pöörlemistelg paralleelseks maaga. Seejärel oli olemas mängumootori füüsikat kasutatav uks.



Joonis 16: Unity mootoris ukse *inspect* vaade (vasak) ja ukse osad (parem).

Et kasutaja saaks ust tõmmata ja lükata, tuli lisada uksele XR Grab Interactable komponent. Kuna kasutaja peab saama ust ukse lingist tõmmata, tuli üle kirjutada komponendi *collider* ukse lingi omaga. Lisaks tuli määrata, et komponent võtaks käe positsiooni asemel käe kiiruse.

Selle tagajärjel saab ust hooga kinni ja lahti tõmmata, sest ukse liigutamisel antakse uksele positsioonile punktkiirust, ning mängumootor arvutab ukse liikumist ümber hinge.

Kuna võetakse ainult kiirus, ei ole võimalik uksele hingede küljest lahti tulla, sõltumata käe positsioonist.

4.6 Vibu

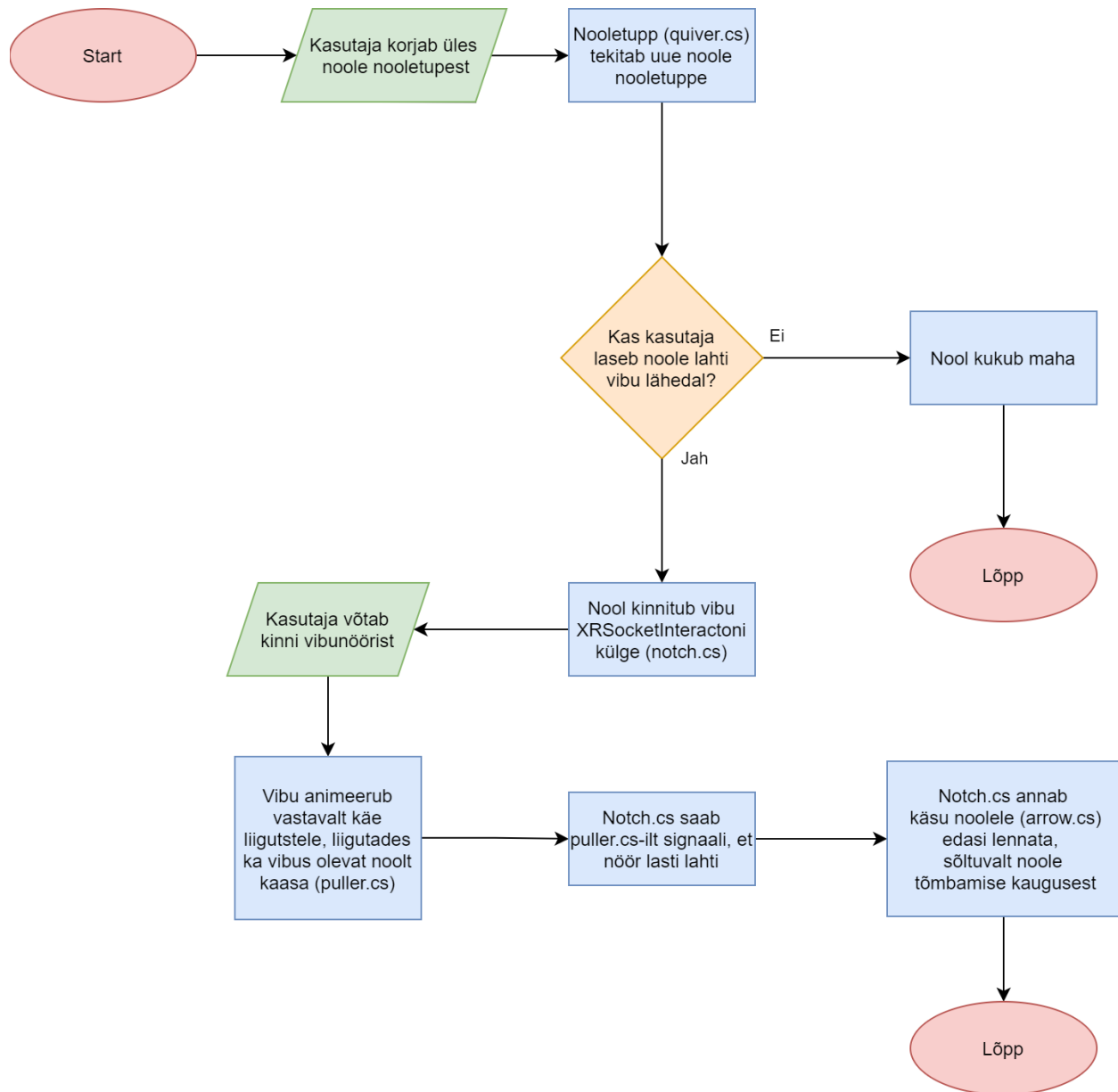
Vibu lasketiir sisaldab kolme tähtsat objekti: vibu, nooled ja sihtmärgid. Nooli saab kasutaja üles võtta nooletupest. Nooletupp on nii mängukomponent kui ka skript, mis laiendab `XRSocketInteractor` skripti. `XRSocketInteractor` skript käitub nagu pesa ning hoiab enda sees kuni ühte mänguobjekti. Kasutaja saab sinna mänguobjekte panna või ära võtta. Seega sobib see hästi noole hoidmiseks.

Nooletupe loomisel tekitab tema skript mänguobjekti koordinaatidele uue noole ning antakse käsk `XRSocketInteractor` skriptile see üles korjata (vt Joonis 17). Kui kasutaja korjab hoitud noole üles, siis loob skript uue noole ning annab uuesti käsu `XRSocketInteractor` skriptile see üles korjata. Niiviisi on võimalik kasutajal üles korjata lõpmatu hulk nooli.

Vibu mänguobjekt koosneb kolmest erinevast skriptist ja nende mängukomponentidest: `notch`, `quiver` ja `puller`. `Puller` skript asub vibunööri keskpunktis ning selle eesmärk on lasta kasutajal tõmmata vibunööri ja jälgida tõmbe kaugust. Skriptis on määratud alg ja lõpp positsioon. Algpositsioon määrab vibunööri nullpositsiooni ning lõpp määrab kõige suurema kauguse, kuhu on võimalik vibunööri tõmmata.

Kui kasutaja võtab kinni `puller` mänguobjektist, siis sõltuvalt käe positsioonist animeerib skript vibunööri, arvutades tõmbamise tugevuse, kasutades alg ja lõpp positsiooni. Lahti lastes muudab ta selle väärtuse nulliks.

`Notch` mängukomponent asub samuti vibunööri keskpunktis ning selle eesmärk on hoida vibus noolt ning seda lasta. Selle jaoks laiendab `notch` `XRSocketInteractor` skripti. `Notch` skript kuulab `puller` skriptilt lahti laskmisi. Kui kasutaja laseb `puller` komponendi lahti, siis annab `notch` skript käsu noolele edasi lennata, koos `puller` skriptilt saadud tõmbamise tugevusega.



Joonis 17: Vibulaskmise loogika diagramm.

Vibu mängukomponent laiendab `XRGrabInteractable` komponenti, ning laseb kasutajal seda kätte võtta. Ta sisaldab ka `Animator` komponenti, mis animeerib vibunööri tõmbamist `puller` skriptist saadud tõmbamise tugevusega.

Noole mängukomponent laiendab `XRGrabInteractable` komponenti, mis laseb samuti kasutajal seda käes hoida. Lisaks on võimalik asetada noolt `XRSocketInteractor`

komponentidesse. Kui nool saab `notch` skriptilt käsu edasi lennata, arvutab ta saadud tugevusega enda uue kiiruse ning liigub edasi, kuni tabab sihtmärki või muud seinu. Skript pöörab ka noole tippu liikumissuunda, et jäljendada realistliku noole lendamist.

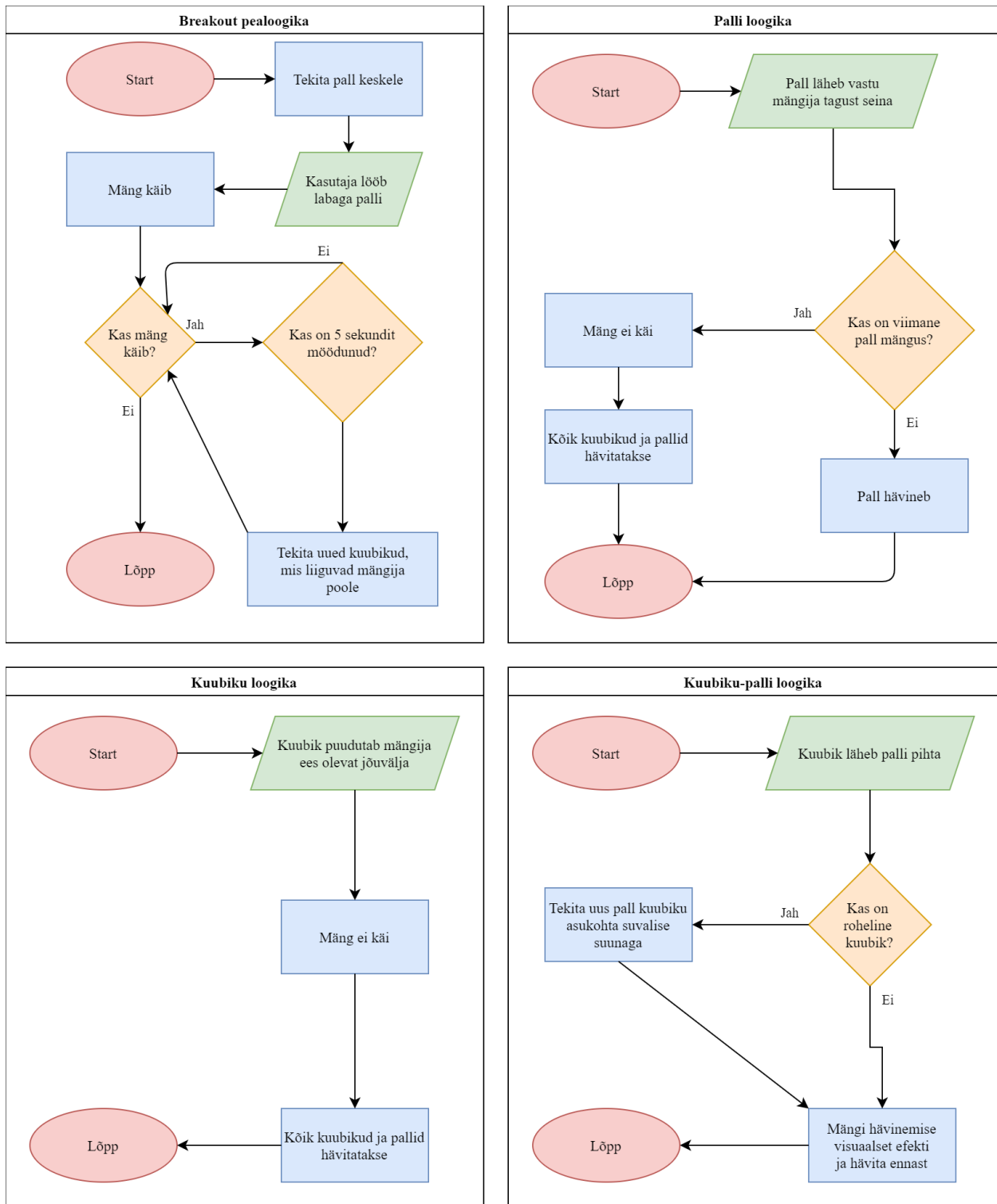
Unity mängumootori füüsikas võivad väikesed kiiresti liikuvad objektid seinadest läbi lennata, kuna objekt liigub kahe füüsika sammu vahel objektist läbi. See juhtub isegi siis, kui määrata *continuous* (pidev) kokkupõrke tuvastuse režiim. Seetõttu kontrollib noole skript veel lisaks iga füüsika tiku vahel, kas jäi mingi objekt eelmise positsiooni ja uue positsiooni vahele. Kui jäi, siis paneb skript noole seisma. Sedaviisi ei saa nool peenikestest tahketest asjadest läbi lennata.

4.7 Breakout

Breakout sisaldab nelja tähtsat mänguobjekti. Nendeks on labu, pall, kast ja sein. Labu koosneb käepidemest ja suuremast tahust, millega palli lüüa. Käepidemel on `XRGrabInteractable` laiendatud skript, mis lubab kasutajal labast kinni võtta. `XRGrabInteractable` on määratud jälgima käte kiirust positsiooni asemel, et anda palli lüües edasi labu kiirus. Kuna XR Interaction Toolkit 1.0.0-pre.1 versioonis oli kiiruse jälgimisega käes hoides labu liikumine ebasujuv, tehti laiendatud skript. Laiendatud skript määrab labu kätte võttes labu vanemaks XR `Rig` komponendi. Sel juhul liigutab mängumootor ise labu inimese keha liikumisega kaasa ning labade liikumine on sujuv.

Teine oluline mänguobjekt on kast. Kastid tekivad 9-s määratud algpunktis ning iga kaader liiguvad lõppseina suunas. Kui kast läheb punasele jõuväljale (mängija ees olev läbipaistev sein) pihta, siis saadab jõuväli mängule käsu mäng lõpetada (vt Joonis 18). Kui kast puudutab palli, siis ta hävitab ennast ära. Kui kast on roheline kast, siis ta tekitab uue palli ning annab talle suvalise suuna. Lisaks annab ta mängu peaskriptile teada, et nüüd on üks pall rohkem mängus. Enne ära hävimist mängivad kastid ära hävitamise visuaalset efekti.

Kolmas oluline mänguobjekt on pall. Pall hoiab enda kiirust piirides, et vältida liiga kiiret või aeglast palli liikumist. Kui pall läheb mängija taha seinu pihta, hävitab ta ennast ning annab mängule teada, et mängus on üks pall vähem (vt Joonis 18). Kui pallide arv on 0, siis mäng saab läbi.



Joonis 18: Breakout elamuse mängu erinevate osade käik.

Implementatsiooni osade kohta on veel kirjeldamata osi, näiteks vibu sihtmärkide liikumine, skoori lugemine, palli seinade peal põrkamine jms. Kuna need on lihtsad osad

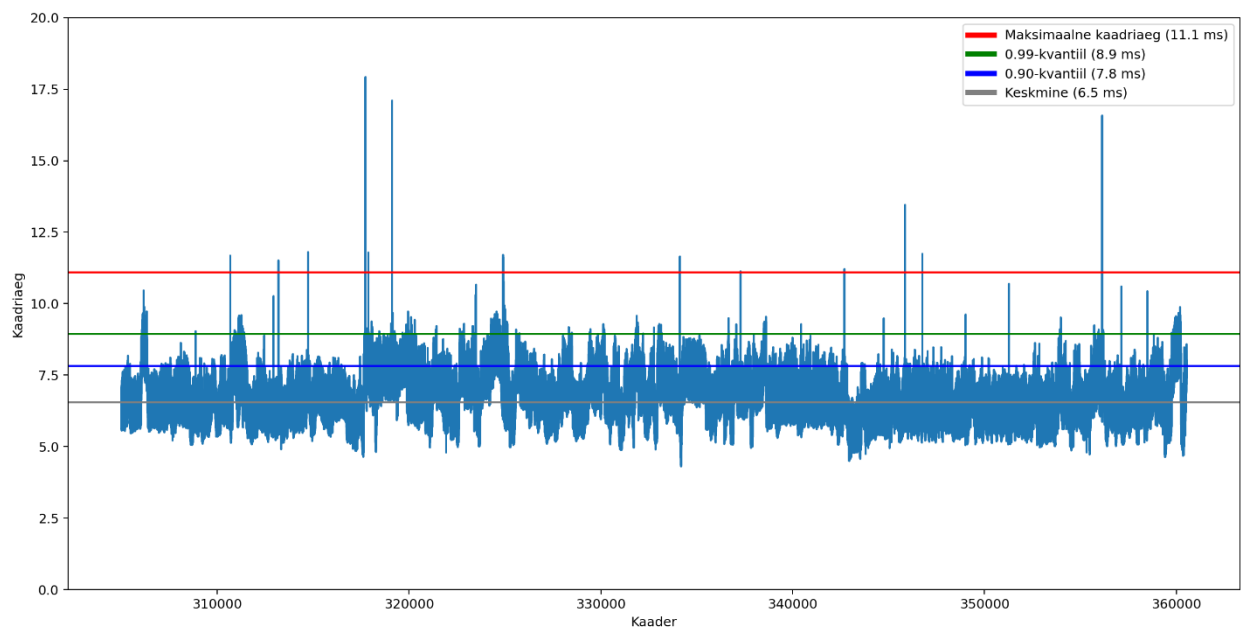
implementatsioonist, ei ole nende töötamist siin kirjeldatud. Edasi tuleb veel testida, kas elamusse implementeeritud mängud on kasutajatele nauditavad ning mida oleks vaja nende puhul parandada.

5. Testimine

Testimine on tarkvaraarenduse oluline osa. See aitab määrata, kas töö loomise ajal tehtud otsused on head või halvad ning mida saaks parandada. Alampeatükis 5.1 analüüsiti elamuse jõudlust. Alampeatükis 5.2 kirjeldatakse DeltaVR elamuse kasutajatestimist eelneva virtuaalreaalsuse kogemusega kasutajate peal ning alampeatükis 5.3 analüüsiti testimise tulemusi.

5.1 Elamuse jõudlus

Jasoni sõnul on üks suurim küberiivelduse tekitaja kõrge latentsus. Ta lisab, et kõrge latentsus põhjustab veel kasutaja halvenenud visuaalset täpsust ja pause virtuaalreaalsuse kohalolekus (Jerald, 2015, lk 183-184). Elamuse tekitatud latentsust saab mõõta läbi kaadriaja. Ideaalis saavutab elamus sellise maksimaalse kaadriaja, mis vastab HMD ekraani sagedusele. Kuna DeltaVR elamus on peamiselt mõeldud AVGR laboris oleva HTC Vive Pro seadme peal kasutamiseks, on see kaadriaeg 11,1 ms (vt Lisa 2B).



Joonis 19: Kaadriaja graafik DeltaVR kasutajatestimisel.

Joonis 19 näitab kaadriaega DeltaVR kasutajatestimise ajal. Testimise ajal kõndis kasutaja virtuaalses elamuses ringi, läbides enamuse hoonest ja selle ümbrusest, ning proovis läbi kõik mängud. Kuigi kaadriaeg on reaalse rakenduse jaoks suure variatsiooniga, jääb enamus kaadriaegadest alla maksimaalse kaadriaja. Seega elamus suudab luua rohkem kaadreid, kui

peaseade suudab näidata. Kuna VR peaseadmehel on üle 99% kordadel järgmine kaader olemas, siis kasutaja variatsiooni ei märka.

Joonisel 19 on ka kirjas keskmine kaadriaeg ja 0.9-kvantiil ja 0.99-kvantiil. 0.99-kvantiil tähendab seda, et 99% kordadest oli suurim kaadriaeg 8,94 ms. Seega üle 99% kaadriaegadest on alla maksimaalse kaadriaega (vt Lisa II: Nõuded riistvarale). Seega MF1 on täidetud ning elamuse jõudlus on sobiv.

5.2 Kasutajatestimine

Eraldi kasutajatestimine on oluline, et aru saada, kuidas kasutajad tegelikult tarkvara kasutavad. Lisaks võivad nad leida vigu ja anda ideid, mille peale arendaja ise ei tule. Schnelli sõnul piisab, kui teha kasutajatestimist kolme kuni viie inimesega (Nielsen, 2000).

Seega kasutajatestimiseks leiti kolm inimest. Inimeste otsimisel eelistati neid, kellel on varem kogemus virtuaalreaalsusega. Varasema kogemusega inimesed peaksid hindama elamust objektiivsemalt, sest nad on rohkem teadlikud, milline on hea virtuaalreaalsus elamus.

Testimist viidi läbi Tartu Ülikooli Delta hoones AVGR Laboris ruum 2007 (vt Joonis 20). Testijad kasutasid mängimiseks HTC Vive Pro peaseadet (vt Lisa II, Joonis 21), mis oli ka peamiseks sihtseadmeks elamusele.



Joonis 20: AVGR Labor Ruum 2007, testimise läbiviimise asukoht.



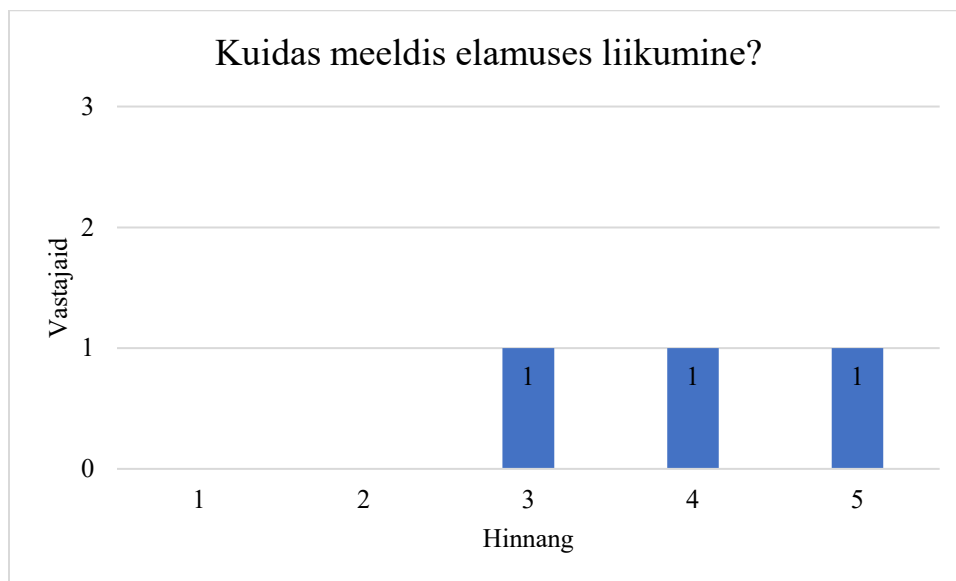
Joonis 21: HTC Vive Pro virtuaalne peaseade.

Testijatele selgitati enne testi algust, kuidas elamuses liikuda ning mida puldil nupud teevad. Pärast selgitust lubati kasutajatel oma valikul elamuses ringi liikuda. Kohustuslik osa oli läbida mõlemad suurimad elamused hoones (vt peatükk 3.5). Kui testijad lõpuks ise elamuseeni ei jõudnud, juhutati nad elamuse juurde.

Testi lõpus anti testijatele küsitlus täita. Küsitluses küsiti kasutaja eelneva VR kogemuse kohta ning arvamust elamuste kohta. Küsimused olid nii vabavormis kui ka hinnang 1-5 palli süsteemis.

5.3 Kasutajatestimise tulemused

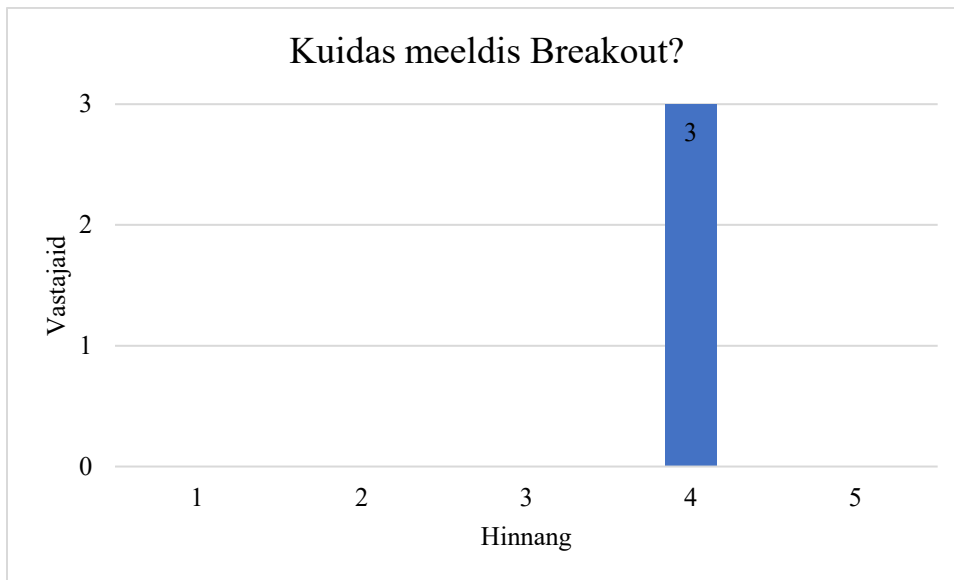
Peatükis 3.2 ja 3.3 kirjutati elamuses liikumise kohta. Testijatelt sai küsitud, kuidas nad liikumist hindaksid ning mis neile selle puhul meeldis või ei meeldinud (vt Joonis 23). Kõik vastajad hindasid elamuses liikumist keskmise hindegga 4,0 viiest ja leidsid, et elamuses liikumisega on probleeme. Probleemidena toodi välja, et oli võimalik läbi seina kõndida ning selle tõttu esines hoones ka kohti, kuhu oli võimalik kinni jääda. Lisati, et tehishüppamine võiks toimida ka siis, kui ei ole sõrme täiesti puuteplaadilt lahti lasknud.



Joonis 22: Elamuses liikumise meeldivus.

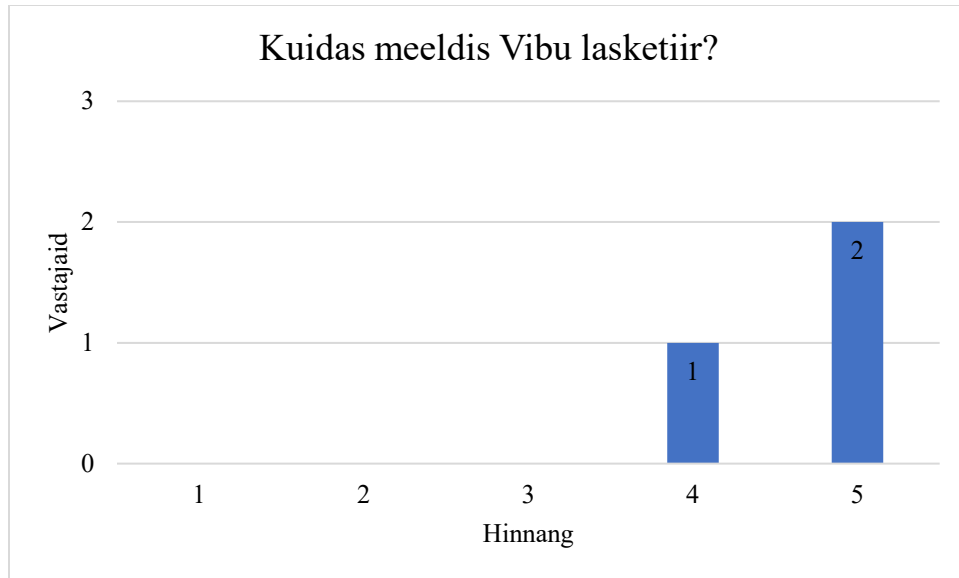
Peatükis 3.5.2 ja 4.6 kirjutati elamuse Breakout loomisest. Testijatelt sai küsitud, kuidas nad Breakout VR elamust hindaksid ning mis neile selle puhul meeldis või ei meeldinud (vt Joonis 24). Kõik vastajad hindasid Breakout mängu keskmise hindegga 4,0 viiest ja leidsid, et seda on veel

võimalik täiendada. Probleemidena toodi välja, et vahel oli raske ennustada palli liikumist ning võiks olla rohkem tagasisidet mängides, näiteks pultide vibreerimine ja rohkem heliefekte. Lisaks ei olnud kohe ilmne, et on võimalik läbi jõuvälja mänguväljale tehishüpata.



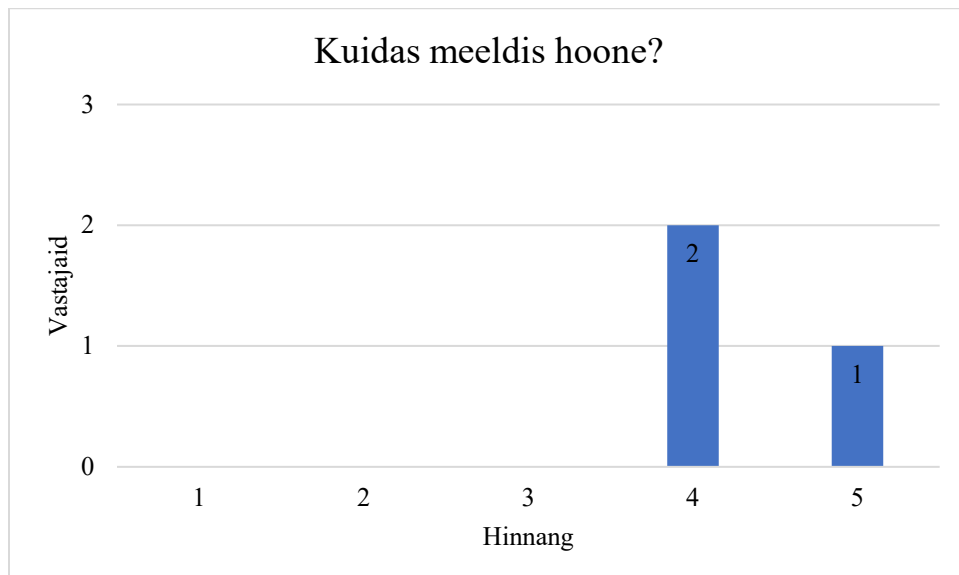
Joonis 23: Breakout mängu meeldivus.

Peatükis 3.5.1 ja 4.5 kirjutati elamuse vibu lasketiiru loomisest. Testijatelt sai küsitud, kuidas neile vibu lasketiir meeldib ning mis neile selle puhul meeldis või ei meeldinud (vt Joonis 25). Kõik vastajad hindasid vibu lasketiiru mängu keskmise hindega 4.6(6) viiest ja leidsid, et seda on veel võimalik täiendada. Vahel oli raske noole trajektoori aimata. Lisaks taheti lasta ka muid asju, kui UFOsid, näiteks robotika ruumis liikuvat robotit. Ühel kasutajal oli päriselus kogemust vibu laskmisega, ning ta soovis sihtida vibu kahe käega. Mängu testimisel tuli välja ka viga, kus sai panna muid objekte peale noolte vibusse, näiteks Breakout mängu labasid.



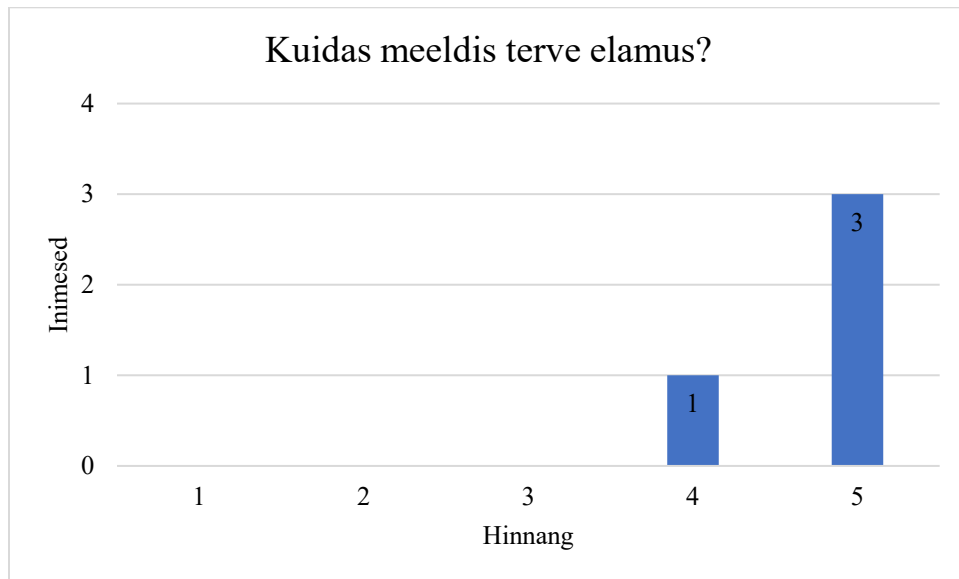
Joonis 24: Vibu lasketiiru meeldivus.

Peatükis 3 kirjutati Delta hoone digitaalse kaksiku loomisest. Testijatelt sai küsitud, kuidas nad hoonet hindaksid ning mis neile selle puhul meeldis või ei meeldinud (vt Joonis 26). Kõik vastajad hindasid hoonet keskmise hindega 4.3(3) viiest ja leidsid, et seda on veel võimalik täiendada. Testijatele ei meeldinud mõnes hoone kohas esinevad graafilised vead ning üldiselt sooviksid näha rohkem korruseid. Kasutajatele meeldis öine keskkond.



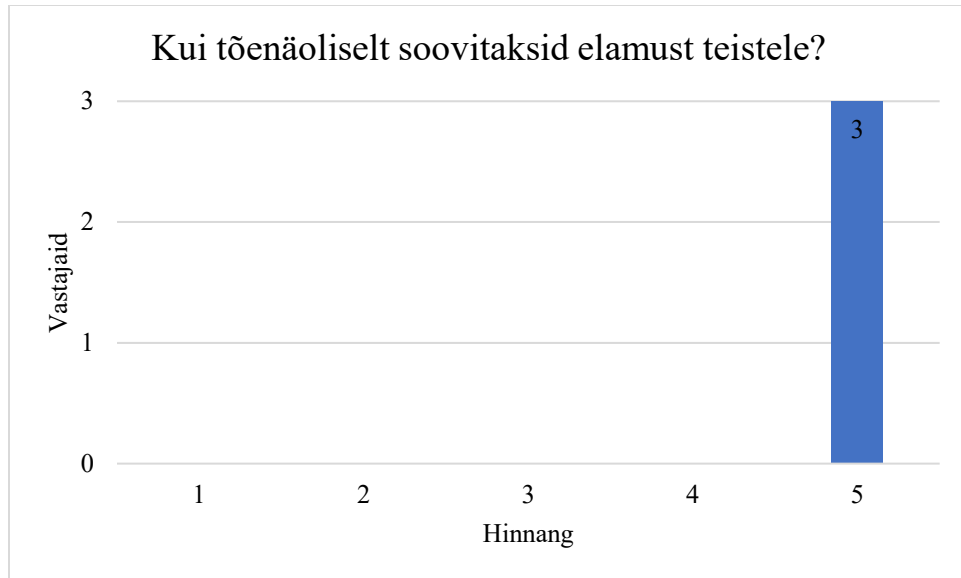
Joonis 25: Hoone meeldivus.

Testijatelt sai veel küsitud, kuidas neile elamus tervikuna ning mis neile selle puhul meeldis või ei meeldinud (vt Joonis 27). Kõik vastajad hindasid tervet elamust keskmise hindega 4.6(6) viiest ja leidsid, et elamus neil meeldis. Testijatele meeldis hoones ringi liikuda ja asjadega interakteeruda. Hoone ja keskkonna välimus oli parem kui algselt oodatu. Veel meeldis neile leida erinevusi hoone ja digitaalse kaksiku vahel.



Joonis 26: Terve elamuse meeldivus.

Viimasena sai küsitud testijatelt, kui tõenäoliselt nad soovitaksid elamust teistele (vt Joonis 28). Kõik vastajad hindasid 5,0. Hinnang anti peamiselt selle tõttu, et nad üldiselt nautisid elamust.



Joonis 27: Elamuse soovitamine.

DeltaVR elamus vajab täiendusi vaatamata üldisele positiivsele meeldivusele. Kõige suurem probleem oli liikumisega. Liikumine on ka oluline aspekt hoone uurimisel, seega see võib ka muid hinnanguid negatiivselt mõjutada. Lisaks tuleks edasi viimistleda elamusi ning Delta hoone mudelit. Mudelil on vaja vähendada visuaalseid vigu ning lisada ka ülejäänud korrused.

Kokkuvõte

Bakalaurusetöö tulemusena valmis virtuaalreaalsuse elamus DeltaVR. Algul analüüsiiti 3 olemasolevat digitaalset kaksikut (PaabelVR, DBV Projekt, CGVR VR Demo) ning leiti, et pole mõistlikku Delta õppehoone ülest VR rakendust, mida AVGR laboris näidata.

Töös sai analüüsitud erinevaid varinate, kuidas saada Delta hoonest mängumootoris kasutatav 3D mudel (originaalne IFC fail, DBV projekti 3D mudel) ning valiti nende hulgast DBV projekti 3D mudel, mis sisaldas Delta õppe- ja teadushoone 1. ja 2. korrust ning selle ümbrust. 3D mudel ei kõblanud kohe virtuaalreaalsuse elamuse jaoks kasutamiseks, sest kaadriaeg oli üle maksimaalse kaadriaaja. Seega mudelis sai lisatud kõrgemad seinad ning Unity mängumootoris konfigureeriti varjatu praakimine. Pärast seda oli kaadriaeg alla maksimaalse lubatud kaadriaaja.

Mudelit sai veel täiendatud, lisades puuduvad asjad; laed ja hoonele omased ribid. Mudelit sai veel optimeeritud, jagades hoone mudeli võrestiku väiksemateks võrestikeks. Mudelis sai parandatud paljusid kohti, kus valguse küpsetamine ei töötanud korralikult ning sai eemaldatud palju graafilisi vigu.

Unity mängumootorisse VR toe lisamiseks analüüsiiti erinevaid variante (SteamVR, Oculus Integration, XR Interaction Toolkit) ning nende hulgast valiti XR Interaction Toolkit. XR Interaction Toolkit võrreldes teiste Unity pakettidega, pakkus tuge ka erinevatele VR raamistikele (Oculus, SteamVR). Implementatsiooni käigus lisati tugi elamuses ruumiulatuses liikumiseks ja tehishüppamiseks. Tugi lisati veel virtuaalsetele kätele ning lisati käe animatsioone (neutraalne, rusikas, ok-positsioon). Implementeeriti virtuaalsed ukсед, mida saab käega kinni võttes kinni ja lahti tõmmata ja lükata. Elamusse sai veel lisatud mängu, mis olid vibu lasketiir ja Breakout VR.

Elamuse peal tehti nii jõudlustestimist kui ka kasutajatestimist. Jõudlustestimise tulemusena leiti, et elamus on alla lubatud maksimaalse kaadriaaja ning inimestel ei tohiks tekkida jõudluse puudumise tõttu küberiiveldust. Kasutajatestimisel testiti kolme inimesega, kelle otsimisel eelistati inimesi eelneva VR kogemusega. Kasutajatega sai testitud erinevaid elamuse osi. Elamuses liikumisega oli suurim probleem, et tehishüppamiseks pidi näpu täiesti puuteplaadilt ära võtma. Kasutajad oleks eelistanud näppu puuteplaadile jätta. Breakout VR elamus kasutajatele meeldis, aga mõnikord oli palli liikumine kasutajale mitte ennustatav. Vibu lasketiir oli kõige meeldivam elamus, aga inimesed tahtsid rohkem asju virtuaalse hoone peal vibuga teha. Hoone digitaalne

jäljendus inimestele meeldis, kuigi nad leidsid veel graafilisi vigu ning oleks soovinud ka rohkem korruseid näha. Elamus kasutajatele meeldis ning soovitaksid ka seda sõpradele, seega on olemas huvi projekti edasiarenduse ja kasutamise kohta.

Töö autor tänab juhendajat Raimond-Hendrik Tunnelit abi ja soovitude eest ning AVGR laborit riistvara ja kastete keskkonna pakkumise eest.

Viidatud allikad

- Clifton, J., & Palmisano, S. (2019). Effects of steering locomotion and teleporting on cybersickness and presence in HMD-based virtual reality. *Springer Nature*, 461-464.
- Jerald, J. (2015). *The VR Book: Human-Centered Design for Virtual Reality*. Morgan & Claypool Publishers. doi:10.1145/2792790
- Leesment, A. (2020). *Delta õppehoone visualisatsioon – hoone ja keskkonna realistlikkus*. Bakalaureusetöö, Arvutiteaduse Instituut, Tartu Ülikool. Allikas: https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=69794&year=2020
- Linde, E. (2019). *Delta õppehoone visualisatsioon - visuaalsed efektid*. Bakalaurusetöö, Arvutiteaduse Instituut, Tartu Ülikool. Allikas: https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=66396&year=2019
- Mayor, J., Raya, L., & Sanchez, A. (2019). A comparative study of virtual reality methods of interaction and locomotion based on presence, cybersickness and usability. *IEEE Transactions on Emerging Topics in Computing*, 10-11. doi:10.1109/TETC.2019.2915287
- Mike, S. (2016). *Breakout*. Kasutamise kuupäev: 28. 04 2021. a., allikas The American Classic Arcade Museum: <https://www.classicarcademuseum.org/breakout>
- Nielsen, J. (18. 03 2000. a.). *Why You Only Need to Test with 5 Users*. Kasutamise kuupäev: 28. 04 2021. a., allikas Nielsen Norman Group: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- Nikolajev, A. (2018). *Delta Building Visualisation and Optimisation*. Bakalaurusetöö, Arvutiteaduse Instituut, Tartu Ülikool. Allikas: https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=61825&year=2018&language=en
- Zimmerman, E. (2003). Play as Research: The Iterative Design Process. rmt: B. Laurel, & P. Lunenfeld, *Design Research: Methods and Perspectives* (lk 176-184).
- Voitenko, A. (2018). *Delta õppehoone keskkonna visualiseerimine*. Bakalaurusetöö, Arvutiteaduse Instituut, Tartu Ülikool. Allikas: https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=61937&year=2018&language=et

Lisad

I. Sõnastik

1. **Tipp** (ingl vertex) on arvutigraafikas kahe või enama külje ühine punkt.²⁷
2. **Tehishüpe** (ingl *teleport*) on kasutaja ühest punktist teisse liikumine hetkega.²⁸
3. **Ruumiulatus** (ingl *room-scale*) on virtuaalreaalsuse liikumisviis, kus kasutaja positsioon muutub samamoodi, nagu päriselus. Peab olema vähemalt 2 x 1.5 meetri suurune ruum.²⁸
4. **SteamVR** on tarkvara raamistik, mis lisab toe erinevatele peas kantavate prillidele.²⁹
5. **Võrestik** (ingl *mesh*) on hulknurkade tippudest, külgedes ja tahkudest koosnev kogum.³⁰
6. **Digitaalne kaksik** (ingl *digital twin*) on virtuaalne esindus objektist või süsteemist.³¹
7. **Huvipunkt** (ingl *point of interest*) on kindel asukoht, kus saab midagi teha.
8. **Asynchronous spacewarp** on kaadriaaja optimeerimise tehnika, kus tekitatakse uus kaader kasutades infot eelmisest kaadrist ning kasutaja positsiooni muutusest. Kui elamus ei saavuta sobivat kaadriaega, siis tekitatakse vahepealsed kaadrid sellega.³²
9. **Varjatu praakimine** (ingl *occlusion culling*) on protsess, mille käigus leitakse objektid, mis on muude objektide poolt varjatud. Leitud objekte ei renderdada.³³
10. **Paus virtuaalreaalsuse kohalolekus** (ingl *break-in-presence*) on hetk, kui isik saab aru, et ta tegelikkuses ei ole virtuaalses maailmas.
11. **Collider** on kuju, mida kasutatakse füüsika kokkupõrgete jaoks. Collider on nähtamatu ning ei pea klappima objekti võrestikuga.³⁴
12. **Valguse küpsetamine** (ingl *lightmapping*) on pindade heleduse ettearvutamine ning faili salvestamine.³⁵
13. **Füüsika samm** (ingl *physics step*) on Unity mängumootoris füüsika interaktsioonide ja muutuste arvutamise hetk. Iga sekund toimub kindel arv füüsika samme.

²⁷ <https://cgvr.cs.ut.ee/wp/index.php/arvutigraafika-terminid/>

²⁸ <https://eevr.ee/dictionary/>

²⁹ <https://store.steampowered.com/steamvr>

³⁰ <https://cgvr.cs.ut.ee/wp/index.php/arvutigraafika-terminid/>

³¹ <https://www.ibm.com/blogs/internet-of-things/iot-cheat-sheet-digital-twin/>

³² <https://developer.oculus.com/blog/asynchronous-spacewarp/>

³³ <https://docs.unity3d.com/Manual/OcclusionCulling.html>

³⁴ <https://docs.unity3d.com/Manual/CollidersOverview.html>

³⁵ <https://docs.unity3d.com/Manual/Lightmappers.html>

II. Nõuded riistvarale

A. Autori arvuti

CPU: AMD Ryzen 5 2600 6c/12t @ 3950MHz

GPU: NVIDIA GTX 1660 6GB

RAM: 4xDDR4 8GB 3000MHz

SSD: ADATA SX8200PNP 1TB NVMe

VR: Oculus Rift S 2,560 x 1,440 @ 80Hz

B. CGVR-Torrance (AVGR labori arvuti)

CPU: AMD Ryzen R9-3900X 12c/24t @ 4.6GHz

GPU: NVIDIA RTX 2080S 8GB

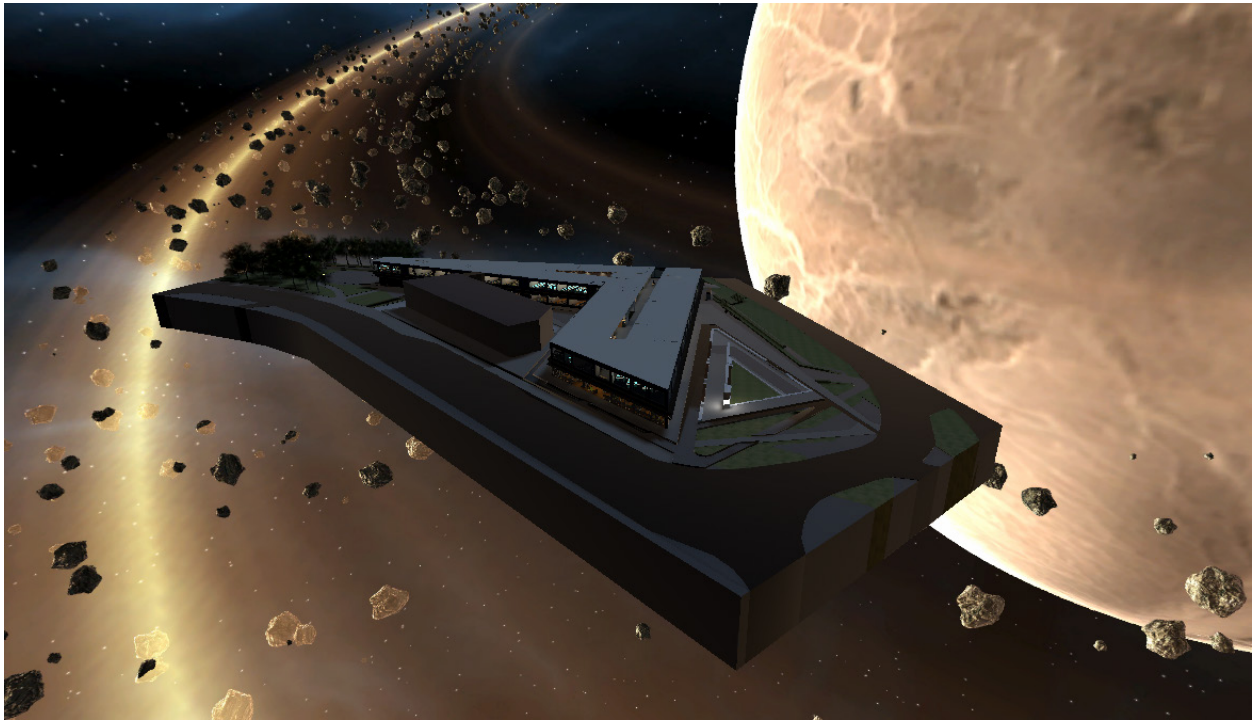
RAM: 2xDDR4 16GB 3200Mhz

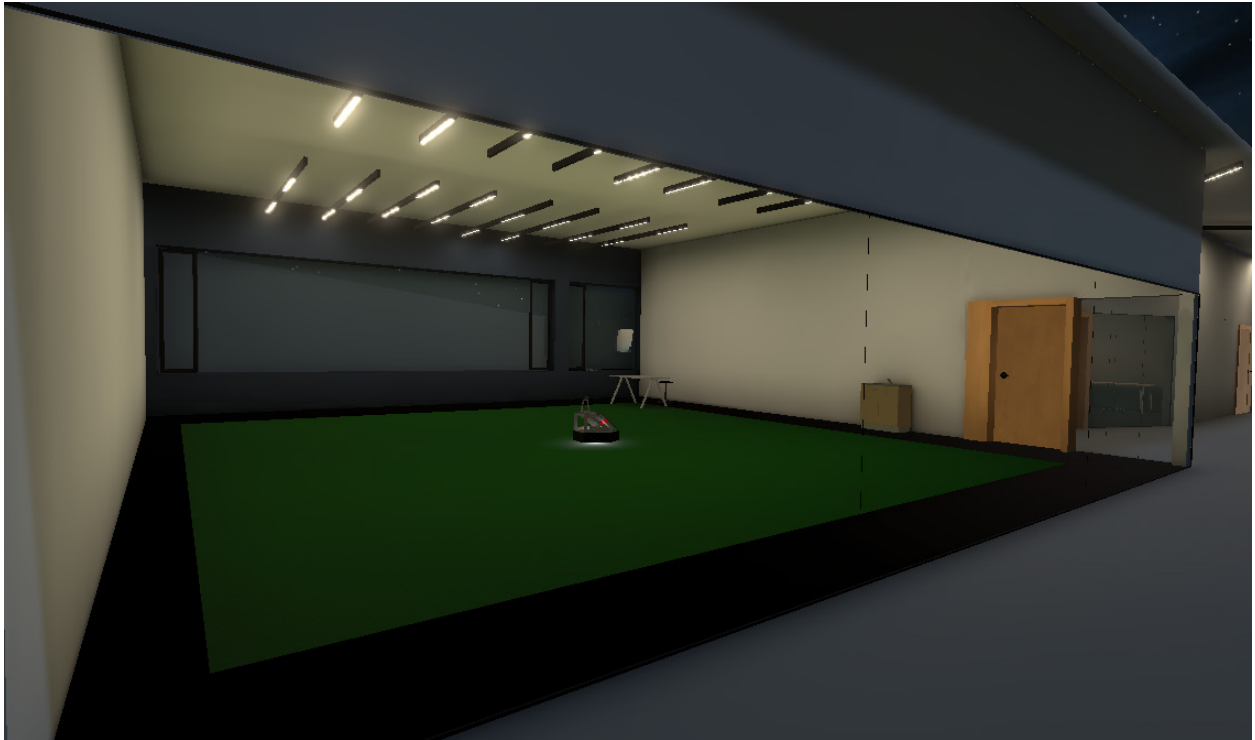
SSD: 1TB Intel 660p M.2 NVMe

VR: HTC Vive Pro 2880 x 1600 @ 90Hz

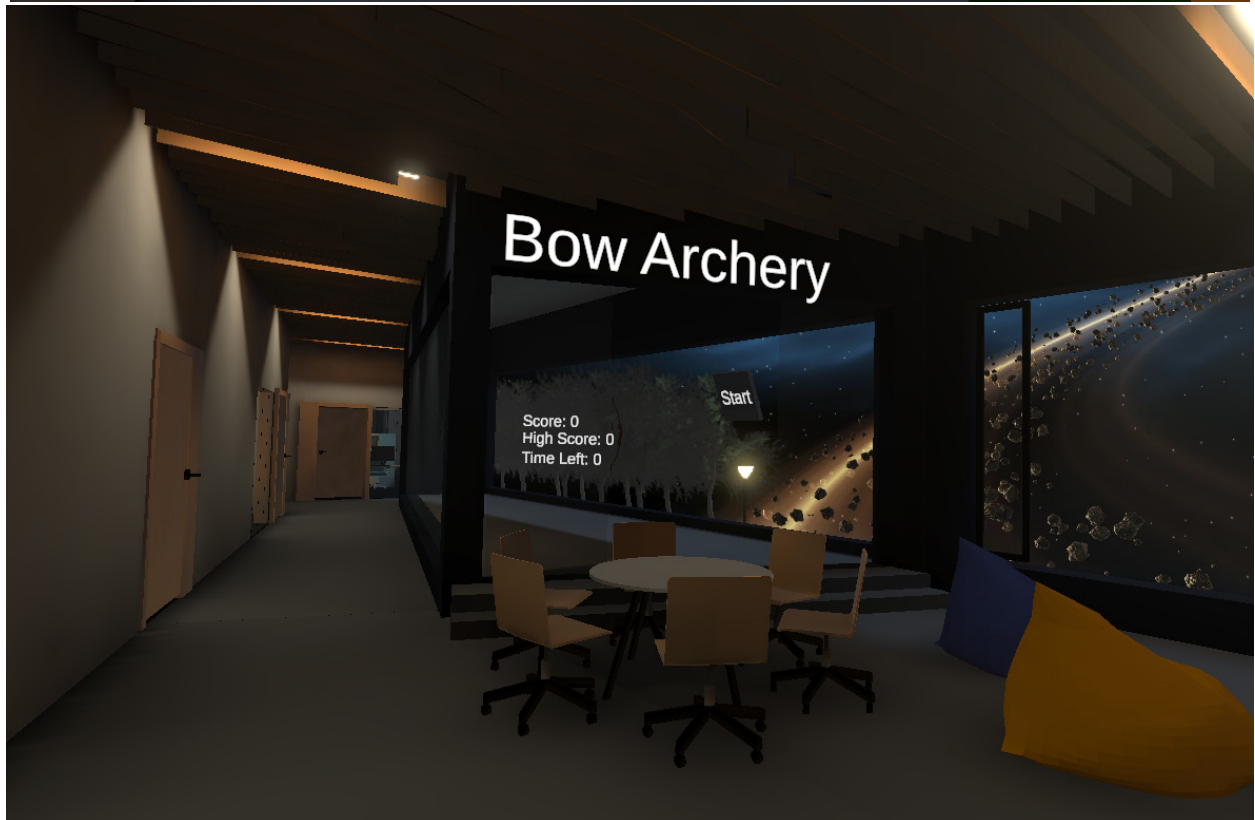
MF1. Maksimaalne 99-kvantiil kaadriaeg 11,1ms Vive Pro peal.

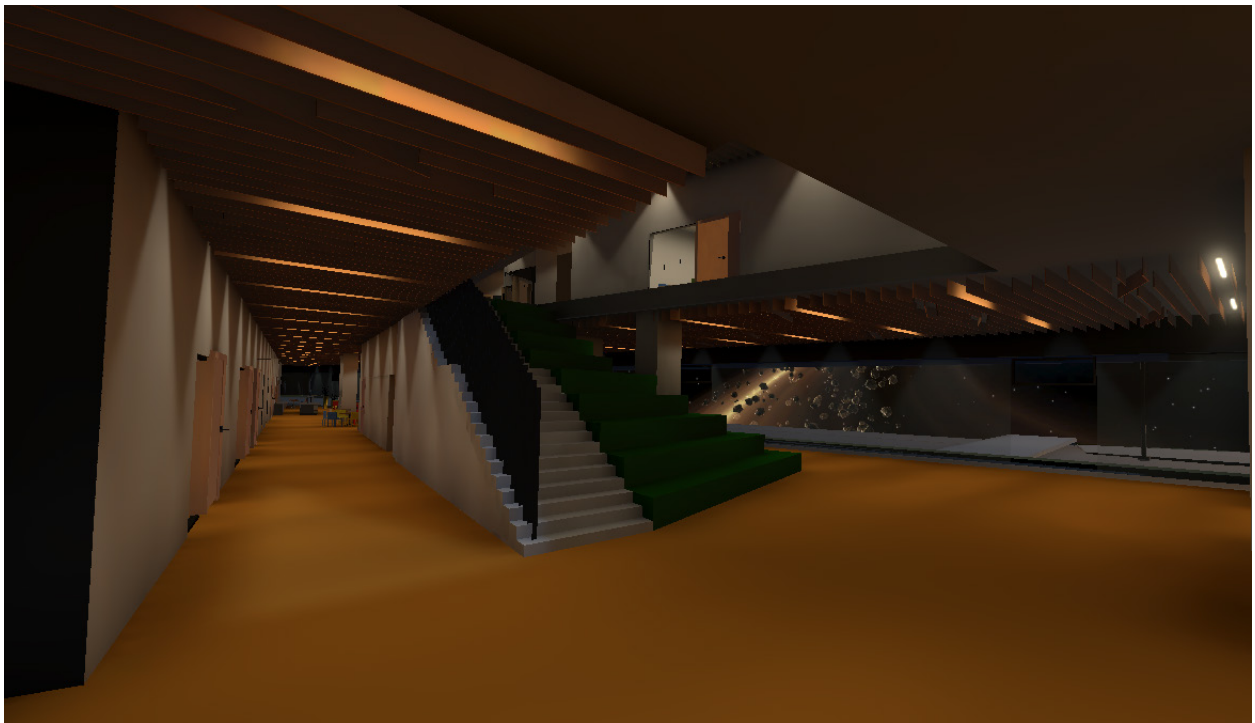
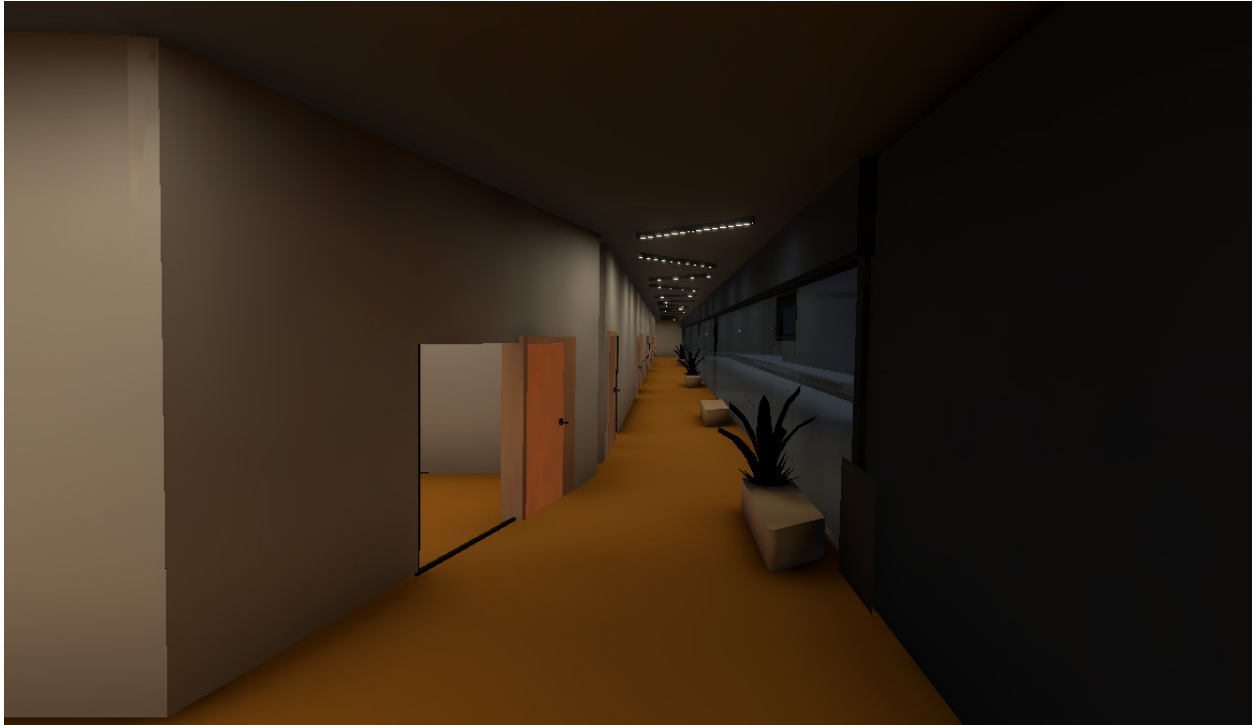
III. Pildid rakendusest

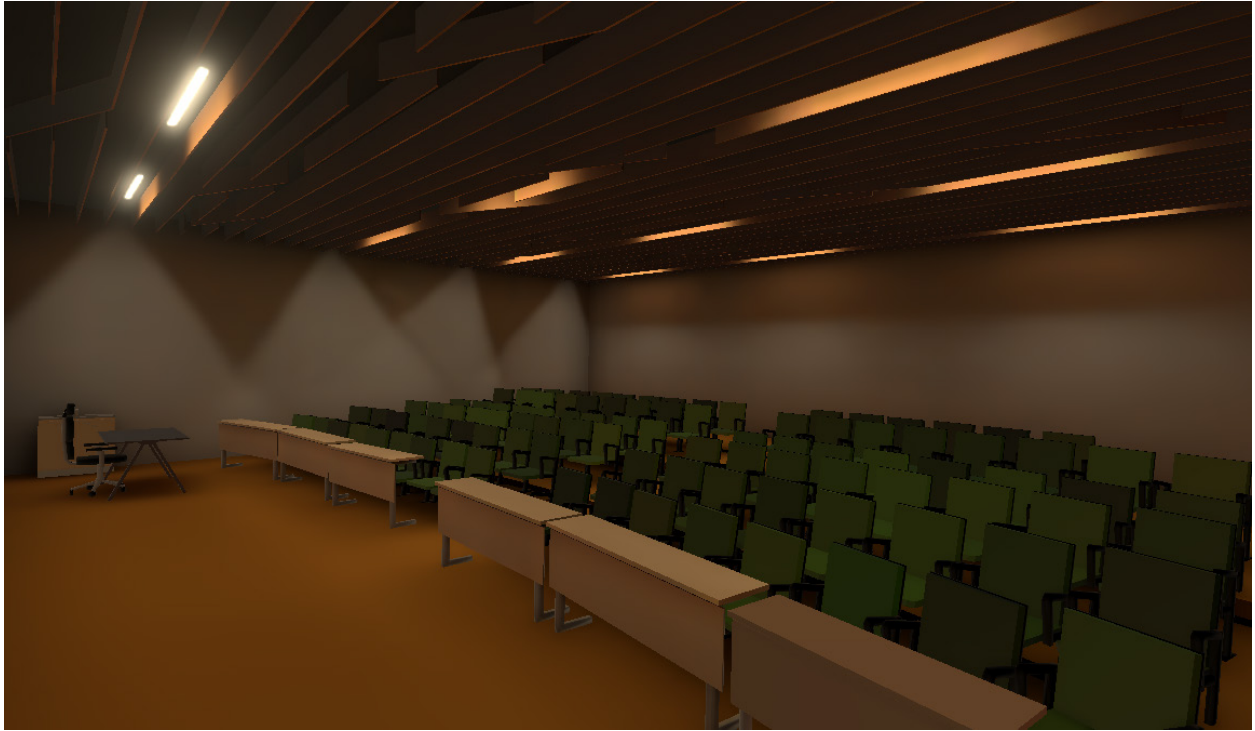








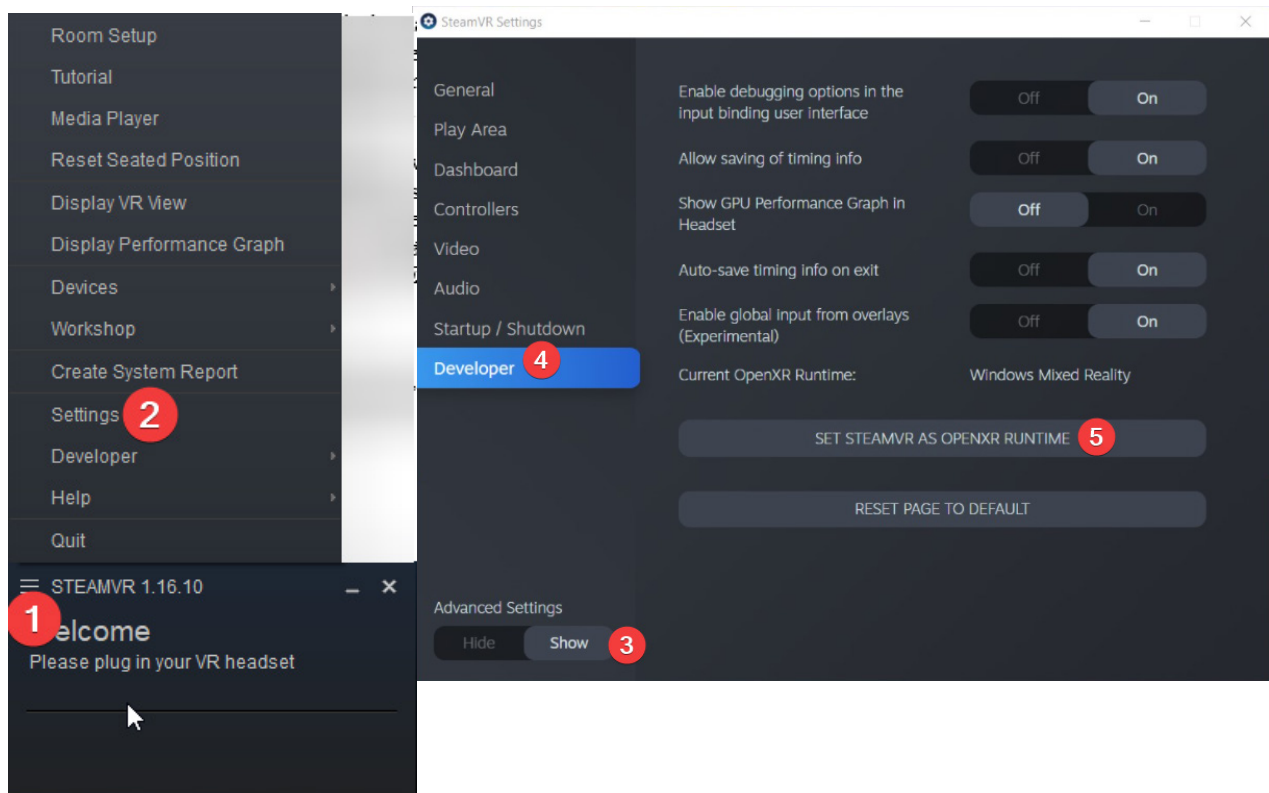




IV. Kasutajajuhend

DeltaVR elamus toetab mitmeid erinevaid virtuaalreaalsuse prille ja raamistike. DeltaVR on ainult testitud SteamVR ja Oculuse tarkvaraga. Teised tarkvarad võivad ka töötada, kuigi need on testimata ning tugi ei ole garanteeritud. Ideaalne kombinatsioon on SteamVR ja HTC Vive Pro, millest mõlemad on saadaval AVGR laboris.

Kui arvutis on saadaval mitu erinevat raamistikku, valib elamus automaatselt OpenXR standardi alusel määratud vaikimisi raamistiku. Vajadusel saab vaikimisi raamistiku määrata SteamVR raamistikuks läbi SteamVR *Settings* menüü. Selle jaoks tuleb käima panna SteamVR. Avada hamburgeri menüü (☰). Valida *Settings*. Määrata *Advanced Settings Showiks*. Valida *Developer* alammenüü, ning sealt valida *SET STEAMVR AS OPENXR RUNTIME*.



Elamuse saab käima panna läbi DeltaVR.exe faili, mis asub kaustas „DeltaVR Build“. Samuti on võimalik elamus ise kompileerida, kasutades lähtekoodi Unity 2020.3.3f LTS versiooni. Lähtekood on kaustas „DeltaVR Source“.



Joonis 28: Parema käe HTC Vive Pro pult.

Elamuse kontrollideks HTC Pro Vive prillidel on (vt Joonis 28):

- 1) Paremat puuteplaati otse vajutada – tehishüppamine.
- 2) Paremat puuteplaati vasakule või paremale vajutada – pööramine 45 kraadi vast
- 3) avalt vajutuse suunale.
- 4) Trigger/Grip nupp – Asjade kätte võtmine.

V. Failid

Tööga kaasatulevas arhiivi struktuur on järgnev:

- /DeltaVR Build/ - sisaldab kompileeritud versiooni mängust.
- /DeltaVR Graafid/ - sisaldab nii kasutajatestimise kui ka jõudlustestimise tulemusi.
- /DeltaVR Graafid/Performance Testing/- sisaldab jõudlustestimise tulemusi ja andmeid.
- /DeltaVR Graafid/Usability Testing/- sisaldab kasutajatestimise tulemusi ja andmeid.
- /DeltaVR Source/ - sisaldab elamuse lähtekoodi.
- /DeltaVR Videod/ - sisaldab videot elamusest.

VI. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Toomas Tamm,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose DeltaVR, mille juhendaja on Raimond-Hendrik Tunnel, 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.

Toomas Tamm

07.05.2021