

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Infotehnoloogia Kolledz

Kristjan Oliver Kruus 175655IDDR

**INIMKEHA LIIGUTUSTE JÄRGIMINE VIRTUAALREAALSUSE SÜSTEEMIDES
KASUTADES HTC VIVE TRACKER SEADMEID**

bakalaureusetöö

Juhendaja: Aleksei Tepljakov
PhD

Tallinn 2017

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kristjan Oliver Kruus

27.12.2017

Annotatsioon

Lõputöö käigus keskenduti HTC Vive Tracker seadmete kasutamisele inimkeha liikumise järgimiseks virtuaalreaalsuses kasutades Unity3D mängumootorit. HTC Vive on üks levinumaid virtuaalreaalsuse platvorme ning seda on võimalik laiendada kasutades Tracker seadmeid pärismaailmas asuvate objektide asukoha järgimiseks. Unity3D on laialt levinud mängumootor, mida on kasutatud paljude virtuaalreaalsuse rakenduste loomiseks. Inimkeha liikumise järgimine virtuaalreaalsuses võimaldab kasutaja keha siduda virtuaalse tegelase omaga, muutes virtuaalse keskkonna realistlikumaks ning pakkudes erinevaid võimalusi mängu- ja filmitööstusele uue sisu loomisel. Diplomitöö käigus loodi lahendus, mis võimaldab kasutajal kontrollida virtuaalse tegelase liigutusi erineva arvu Tracker seadmete abil. Seadmete asukohtade põhjal ruumis seotakse need virtuaalse tegelase kehaosadega ning kasutatakse seejärel pöördkinemaatikat tegelase keha liigutamiseks võimalikult realistlikul viisil. Kasutajal on võimalik vaadelda ennast peegelpildis ning jälgida kuidas tegelase keha paindub erinevate kehapooside korral. Rakendus võimaldab ka salvestada iga kasutaja puhul seadmete asukohtade põhjal kasutajaprofiile, mida saab kasutada virtuaalse tegelase keha sidumisel kasutaja omaga rakenduse taaskäivitamisel. Kokkuvõtvalt võib öelda, et loodud rakendust on võimalik kasutada erinevates tehnoloogiaharudes, näiteks animatsioonide salvestamisel filmitööstuses ning sotsiaalsete virtuaalreaalsuse rakenduste korral.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 27 leheküljel, 5 peatükki, 13 joonist, 2 tabelit.

Abstract

Full Body Human Motion Tracking in Virtual Reality Systems Using HTC Vive Trackers

The topic of this thesis is concerned with using the HTC Vive virtual reality platform for full body human motion tracking in virtual reality applications using Unity3D game engine. HTC Vive is currently one of the most widely used virtual reality platforms, which can be extended by using Tracker devices for tracking real world objects. Unity3D is a game engine that has gained widespread use for developing virtual reality applications. Full body motion tracking in virtual reality allows users to connect their body movement with virtual avatars making the virtual experience more realistic and also offers many advantages for content creation in film and game industry. In the scope of the work presented in this thesis, an application for controlling the movement of virtual characters using different number of Vive Trackers was created. Using the position in space the Tracker devices are connected with the virtual avatar and an inverse kinematics based method is used to calculate missing joint positions and angles. Connected joints can be further adjusted afterwards to make the tracking more exact and thereby more realistic. The user is able to see his/her reflection in the mirror image and take different poses to test the model accuracy. Different number of Tracker devices can be used, so the user is not restricted to having exact number of devices. The model accuracy improves incrementally when more devices are added. Missing tracking points are calculated with inverse kinematics or animations where possible. The application allows users to save their calibrated profiles and load them without recalibrating with the virtual body, simplifying application development and adding possibility for quick setup for multiple users. There are numerous ways to further expand the created solution for different technical industries for example saving animations for film industry and social VR experiences.

The thesis is in Estonian and contains 27 pages of text, 5 chapters, 13 figures, 2 tables.

Lühendite ja mõistete sõnastik

VR	Virtuaalreaalsus
peaseade	<i>Head Mounted Display(HMD)</i> , seade, mis võimaldab vaadelda 3D-maailma kumeral, inimese silma perspektiivis
HTC Vive	Tarkvarafirma HTC poolt toodetav virtuaalreaalsuse platvorm
Oculus	Tarkvarafirma, mis toodab omanimelist virtuaalreaalsuse peaseadet
Tracker	HTC Vive lisaseade, mis võimaldab järgida seadmete asukohta ruumis
Unity3D	Levinud mängumootor
Xbox	Microsofti poolt 2001. aastal loodud videomängude bränd, mis toodab ka mängukonsoole
Kinect	Xbox mängukonsooli lisaseade, mis võimaldab jälgida keha liikumist
Drive4VR	Draiver, mis võimaldab kasutada Kinecti koos HTV Vive või Oculus Rift platvormidega
OptiTrack	Visuaalsetel markeritel põhinev jälgimistehnoloogia
MoCap	<i>Motion Capture</i> , füüsilise keha liikumise salvestamine optiliste seadmete või sensorite abil
POGO-klemm	Klemm, mida kasutatakse jooteta ühenduse loomiseks elektroonikakomponentide vahel
API	<i>Application Programming Interface</i> , rakenduse programmeerimisliides
Perception Neuron	Magnetiliseid sensoreid kasutav inimkeha asukoha järgimise süsteem
Lighthouse	Ameerika korporatsiooni Valve poolt välja töötatud lahendus esemete asukoha järgimiseks ruumis
Baasjaam	<i>Base Station</i> , „Lighthouse“ asukoha tuvastamise süsteemis infrapunakiirte allikas
Pöörkinemaatika	<i>Inverse Kinematics</i> , liigete asukoha leidmine matemaatilise mudeli abil
Pöörkinemaatika sihtmärk	<i>Target</i> , ruumipunkt, mida kasutatakse pöörkinemaatika mudeli poolt seotud liigete asukoha ning nurga leidmiseks
JSON	<i>JavaScript Object Notation</i>

Sisukord

Autorideklaratsioon	2
Annotatsioon.....	3
Abstract Full Body Human Motion Tracking in Virtual Reality Systems Using HTC Vive Trackers	4
Lühendite ja mõistete sõnastik	5
Sisukord.....	6
Jooniste loetelu	8
Tabelite loetelu	9
1 Sissejuhatus	10
2 Olemasolevad lahendused inimkeha jälgimiseks virtuaalreaalsuses.....	11
2.1 Kinect.....	11
2.2 OptiTrack.....	12
2.3 Perception Neuron	13
2.4 HTC Vive Trackerid.....	14
2.5 Autori hinnang inimkeha jälgimise süsteemidele virtuaalreaalsuses	15
3 Vive Trackereid kasutavad tarkvaralised lahendused	17
3.1 iKinema Orion	17
3.2 FinalIK VRIK	18
4 HTC Vive Trackerite kinnitamine.....	19
4.1 Vahendid Trackerite kinnitamiseks	19
4.2 Vive Trackerite jaoks sobivad kinnituspunktid	20
5 Tehniline teostus.....	23
5.1 3D-stseeni loomine	23
5.2 Ühendatud seadmete tuvastamine	27
5.3 Seadmetele rolli määramine	28
5.4 Pöördkinemaatika algoritmi valik	31
5.5 Pöördkinemaatika komponentide lisamine virtuaalsele tegelasele	33
5.6 Virtuaalse tegelase keha sidumine seadmetega	33
5.7 Tegelase keha ning jäsemete pikkuse kohandamine	34

5.8 Kasutajaprofiilide laadimine ning salvestamine.....	35
6 Kokkuvõte	37
7 Kasutatud kirjandus	38
Lisa 1 - SteamVR teegi abil seadmete tuvastamine	40
Lisa 2 – Kalibreerimispunktite kaugus põhikaamerast.....	41
Lisa 3 – Alakeha jaoks vajalike kalibreerimispunktide määramine	42

Jooniste loetelu

Joonis 1. Kinect Xbox One seadmele.....	12
Joonis 2. OptiTrack inimkeha jälgimissüsteem.....	13
Joonis 3. Perception Neuron inimkeha jälgimissüsteem	14
Joonis 4. HTC Vive Tracker seade koos Trackstrap velcro-kinnitusega	15
Joonis 5. SteamVR komponendid projekti kaustas	24
Joonis 6. Tekstuuri seadistused	25
Joonis 7. Avatari peegelpildis kuvamiseks kasutatava kaamera seadistused	25
Joonis 8. Peegli jaoks kasutatava tasapinna asukoha ning suuruse seadistused	26
Joonis 9. Peegli jaoks kasutatava tasapinna varjutaja seadistused	26
Joonis 10. Kasutajaliidese paneeli seadistused.....	26
Joonis 11. Näide läbi tegelase pea renderdatud vaateväljast.....	29
Joonis 12. Pöördkinemaatika sihtmärgid enne (a) ning pärast (b) sidumist virtuaalse tegelasega.....	34
Joonis 13. Tegelase jäsemete pikkuse ning nurga kohandamine peale automaatset kalibreerimist	35

Tabelite loetelu

Tabel 1. Virtuaalreaalsuse rakendustes inimkeha jälgimise jaoks kasutatavate süsteemide võrdlus	16
Tabel 2. Pöördkinemaatika mudel	32

1 Sissejuhatus

Inimkeha liigutuste jälgimist on filmitööstuses tegelaste animeerimiseks kasutatud juba aastakümneid. Virtuaalreaalsuse kiire areng viimastel aastatel on loonud võimaluse kasutada sama tehnoloogiat ka kasutaja poolt kogu kehaga kontrollitavate tegelaste ehk avataride puhul. Käesolevas diplomitöös kasutatakse avataride juhtimiseks HTC Vive [1] virtuaalreaalsuse süsteemi ning selle lisaseadmeid mida nimetatakse „Trackeriteks“ [2]. Võrreldakse olemasolevaid seadmeid ja tarkvaralisi lahendusi, mida on kasutatud inimkeha liikumise järgimiseks virtuaalreaalsuses. Tuuakse hetkel kasutatavate süsteemide puudused, mis raskendavad avataride kasutamist virtuaalreaalsuse rakendustes ning püütakse leida neile lahendusi. Töö eesmärgiks on arendajate jaoks Vive Trackeritele rakenduste loomist lihtsustada ning muuta lõppkasutaja jaoks avatari kalibreerimine võimalikult lihtsaks ning arusaadavaks. Lõpptulemusena luuakse avatud lähtekoodiga rakendus, mis toetab erinevat arvu Trackereid kinnitatuna erinevate kehaosade külge. Arendustöökäsitatakse Unity3D mängumootorit. Kasutatakse pöördkinemaatikat, mis võimaldab simuleerida jäsemete paindumist ka juhul kui kasutatakse väiksemat hulka Trackereid. Põhiline fookus antud lõputöös on HTC Vive virtuaalreaalsuse platvormil ning selle platvormiga ühilduval seadmetel ning programmidel.

2 Olemasolevad lahendused inimkeha jälgimiseks virtuaalreaalsuses

Inimkeha liigutuste jälgimine virtuaalreaalsuses on uudne lähenemine keha liikumise salvestamisele ning analüüsimisele ning töötlemisele, mida on varasemalt kasutatud filmi- ning mängutööstuses. Paljud kasutusel olevad liikumise jälgimise süsteemid ei toeta reaalses liikumise jälgimist, seega pole need virtuaalreaalsuses toimivate rakenduste jaoks piisavalt võimekad. Alljärgnevalt toob autor välja olemasolevad riistvaralised lahendused, mida kasutatakse inimkeha jälgimiseks virtuaalreaalsuses.

2.1 Kinect

Kinect on Microsofti poolt Xbox konsoolidele loodud optiline seade, mis on võimaline tuvastama inimeha kehaosi ning nende liikumist kasutades infrapuna kaamerat. Esimene Kinecti versioon tuli turule 2010. aastal Xbox 360 konsoolidele ning teine versioon 2013. aasta lõpus Xbox One konsoolidele. Kinecti teise versiooni sügavuskaamera vaatenurgaga 70x60 kraadi ning on võimeline tuvastama keha liikumist kuni 4.5 meetri kauguselt kaadrisagedusega 30Hz [3]. Inimkeha liikumise jälgimiseks ei ole soovitatav kasutada mitut Kinecti seadet korraga, sest mitmest allikast tulenev infrapunakiirus tekitab valgulainete interferentsi, mis vähendab jälgimise täpsust. Kinect töötab ainult juhul kui kasutaja vaatab otse kaamera suunas. Juhul kui kasutaja seisab küljega kaamera poole, ei suuda Kinect tihti tuvastada kasutaja poosi [4]. Kinecti on kasutatud ka HTC Vive virtuaalreaalsuse süsteemiga. Arendajate poolt on loodud mitmeid niiõelda hübriidlahendusi, mis kasutavad VR-peaseadet ning kontrollereid pea ja käte jälgimiseks ning ülejäänud keha jälgimiseks kasutatakse Kinecti. Üks tuntumaid seesuguseid lahendusi on „Drive4VR“, mida saab kasutada nii Vive kui ka Oculus platvormidel [5]. Tegu on tasulise rakendusega, mida hetkeseisuga saab 10 minutit proovida ning seejärel on vajalik litsentsi ostmine. Kinecti kasutamine ei vaja inimese keha külge kinnitatud seadmeid, kuid jälgimise kvaliteet ei ole kahjuks võrreldav Vive Trackeritega. Kinecti puhul peab kasutaja olema kogu aeg suunaga kaamera poole ning ei saa ruumis vabalt ringi liikuda. Samuti tekivad jälgimisel tihti häired ning animatsioonid ei ole piisavalt

kiired ning tekitavad tihti hüplemist, mis on kasutaja jaoks häiriv ning võib tekitada iiveldustunnet. Kinecti hinnaks on 2017.aasta novembri seisuga 99 dollarit [6]. Kinecti Xbox One seadmele on kujutatud Joonisel 1.



Joonis 1. Kinect Xbox One seadmele

2.2 OptiTrack

„OptiTrack“ on optilistel markeritel põhinev liikumise jälgimise süsteem. Optilisi markereid on selle jälgimissüsteemi puhul kahte liiki – aktiivsed ning passiivsed [7]. Passiivmarkerid on väikesed keermeauguga plastkuulid, mis on kaetud valgust hästi peegeldava kilega. Plastkuulid kinnituvad keerme abil väikese plastikust jala külge, mis omakorda liimitakse kasutaja riiete või naha külge. Tootja pakub ka spetsiaalseid riidest kombinisooni, mis peegeldavad vähesel määral valgust, parandades jälgimise kvaliteeti. Aktiivmarkeriteks on rütmiselt vilkuvad LED-tuled, mis on ühendatud toiteallikaga. Toiteallikas suhtleb raadiolainete teel vastuvõtjaga, mis on ühendatud internetikaabli abil arvutiga. Aktiivmarkerite eeliseks on asjaolu, et igal markeril on enda vilkumissagedus ning igale markerile on määratud talle vastav identifikaator [8]. See võimaldab eseme asukohta täpsemalt määrata olukorras, kui suurem osa esemest on kaamerate eest varjatud. OptiTracki süsteemi kasutab markerite jälgimiseks infrapunakaameraid kaadrisagedusega kuni 240 kaadrit sekundis [9]. OptiTracki puhul ei ole võimalik osta ühte baaskomplekti, vaid komponendid tuleb osta eraldi. Virtuaalreaalsuse rakendustega ühildub OptiTrack „Slim 13E“ kaamera [7] hinnaga 1499 dollarit [10], millele lisanduvad markerid hinnaga 50 dollarit tükk [11]. OptiTrack süsteemi koos kehale kinnitatavate passiivmarkeritega on kujutatud Joonisel 2.

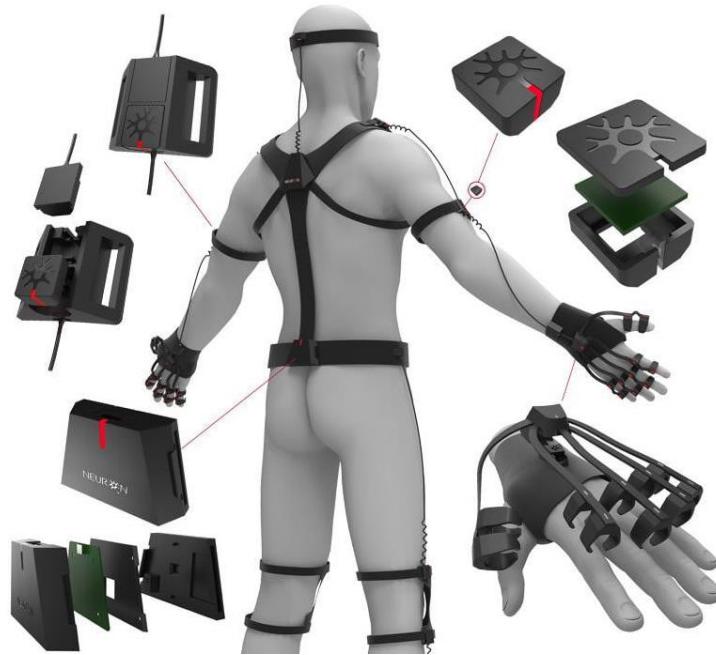


Joonis 2. OptiTrack inimkeha jälgimissüsteem

2.3 Perception Neuron

„Perception Neuron“ on USA firma Noitom poolt välja töötatud magnetilistel sensoritel põhinev inimkeha jälgimise süsteem, mis kasutab kehaosade asukoha tuvastamiseks güroskoobist, akseleomeetrist ja magnetomeetrist koosnevaid sensoreid [12]. Perception Neuron koosneb 32-st või 18-st sensorist, mida on võimalik keha külge velcro-paeltega kinnitada [13]. Optiliste seadmete puuduseks on sensorite varju sattumine või kaamera vaateväljast välja jäämine. Magnetiliste seadmete puhul seda probleemi ei esine, seadmed toimivad autonoomselt ning ei vaja asukoha määramiseks väliseid seadmeid. Probleemiks magnetiliste seadmete puhul on järgmise kadu, kui seade satub liiga lähedale metallist esemetele või teistele elektroonilistele seadmetele. Selle põhjuseks on see, et ferromagnetilised materjalid sensorite läheduses häirivad sensori lokaalset magnetvälja ning moonutavad nurga määramiseks olulisi mõõteandmeid [14]. Perception Neuroni hinnaks on 18-ne jälgimispunkti korral 2017.aasta novembri seisuga 999 dollarit [15].

Perception Neuron jälgimissüsteemi koos suurendatud komponentidega on kujutatud Joonisel 3.



Joonis 3. Perception Neuron inimkeha jälgimissüsteem

2.4 HTC Vive Trackerid

Trackerid on Taiwani firma HTC poolt Vive virtuaalreaalsuse peaseademele mõeldud lisaseadmed, mis võimaldavad järgida pärismaailma esemeid virtuaalreaalsuses. Välimuselt meenutab Tracker ketast, mille äärtes on kolm 120 kraadise nurga all asuvat trapetsikujulist „sarve“. Trackerite tagaküljel asub $\frac{1}{4}$ tollise keermega poltliides, mis võimaldab kinnitada Trackerit esemete külge, ning kuus POGO-klemmi. POGO-klemmide kaudu saab anda sisend-väljundsignaale erinevatele Trackereid kasutavatele lisaseadmetele, nagu näiteks püstolid või tennisereketid [16]. Trackerid töötavad kasutades sama „Lighthouse“ tehnoloogiat nagu Vive peaseade ning kontrollid. Lighthouse kasutab kahte baasjaama või „majakat“, mis kiirgavad kindla rütmi alusel välja infrapunakiirgust [17]. Lighthouse asukoha järgimise süsteemis on seadmete pind kaetud valgussensoritega, mis baasjaamadest väljuvate infrapunakiirte langemisnurka kasutades määravad ära seadme asukoha ruumis. Trackerite lisamiseks Lighthouse süsteemi kasutava süsteemi mängualasse tuleb ühendada arvutiga Trackeriga kaaasasolev USB-adapter ning vajadusel tuvastada Tracker manuaalselt. Trackereid on võimalik kasutada ka teiste virtuaalreaalsuse peaseadmetega, asendades nende asukoha määramise sisendi peaseadme külge kinnitatud Trackeri asukohaga. Näitena võib välja tuua Ivan

Mathy, kes 2017.aasta aprillis kasutas Vive Trackereid koos Oculus Rift peaseadmega [18]. Antud süsteem oli võimeline jälgima nii Oculus Touch kontrollereid kui ka peaseadme ning jalgade külge kinnitatud Trackerite asukohta ruumis. Trackerid kuuluvad võrreldes teiste inimkeha jälgimise süsteemidega keskmisesse hinnaklassi, 2017.aasta novembri seisuga on ühe Trackeri hinnaks 99 dollarit [19]. HTC Vive Trackerit koos Trackstrap velcro-kinnitusega on kujutatud Joonisel 4.



Joonis 4. HTC Vive Tracker seade koos Trackstrap velcro-kinnitusega

2.5 Autori hinnang inimkeha jälgimise süsteemidele virtuaalreaalsuses

Autori hinnangul tuleks olemasolevate inimkeha jälgimise süsteemide jaoks, mis on kasutatavad ka virtuaalreaalsuses, võtta arvesse seadmete ning vajaliku tarkvara hinda ja integreeritavust HTC Vive virtuaalreaalsuse süsteemi ning Unity3D mängumootoriga.

Kinect on autori hinnangul praeguseks hetkeks vananenud tehnoloogia, mis ei ole piisav tagamaks virtuaalreaalsuses piisavalt kvaliteetset jälgimist. Kinecti tootmine on Microsofti poolt lõpetatud [20] ning tulevikus võib osutada selle integreerimine teiste rakendustega keeruliseks.

OptiTrack ning Perception Neuron on mõeldud pigem äriklientidele, seda just nende kõrge hinna tõttu. Samas on võimalik antud süsteeme kasutada küllalt suurel alal nagu näiteks messidel või VR-mängukeskustes.

Vive Trackerid on autori hinnangul valitud süsteemide seast parima hinna ja kvaliteedi suhtega. Trackerid on kallimad kui Kinect, kuid keha järgimine on oluliselt parem ning annab suurema liikumisvabaduse. Trackerid jäävad kasutava mänguala suuruse osas alla

OptiTracki ning Perception Neuron süsteemidele, kuid toimiva täiskeha jälgimist toetava süsteemi hind on praegusel hetkel Trackerite puhul kordades odavam. Trackerite ühildamine antud lõputöös kasutatavate HTC Vive peaseadme ning Unity3D mängumootoriga ei vaja täiendavaid lisaprogramme ega -teeke. Alljärgnevalt on esitatud võrreldavad tegurid tabeli kujul.

Tabel 1. Virtuaalreaalsuse rakendustes inimkeha jälgimise jaoks kasutatavate süsteemide võrdlus

Seade	Tegurid, mis võivad põhjustada keha asukoha määramises anomaaliaid	Integreeritavus HTC Vive VR-süsteemiga	Hind
Kinect	Keha ning ruumis olevate esemete varju sattumine, küljega kaamera poole seismine	Kasutades kolmanda osapoole teeke	Madal
OptiTrack	Keha ning ruumis olevate esemete varju sattumine	Kasutades API-d	Kõrge
Perception Neuron	Metallist esemete ning elektroonika lähedus	Kasutades API-d	Kõrge
Vive Tracker	Keha ning ruumis olevate esemete varju sattumine	Ei vaja täiendavaid lisaprogramme	Keskmine

3 Vive Trackereid kasutavad tarkvaralised lahendused

Inimkeha jälgimiseks on HTC Vive platvormil hetkel võimalik kasutada mitmeid olemasolevaid pöördkinemaatika lahendusi, mis toimivad nii Vive baaskomplektiga kui ka Trackeritega. Vive baaskomplekti all on mõeldud seadistust, kus kasutatakse ainult peaseadet ning kahte kontrolleri. Järgnevalt tuuakse välja olulisemate virtuaalreaalsuse jaoks kohandatud pöördkinemaatika lahenduste eripärad ning puudused ja tugevused.

3.1 iKinema Orion

„iKinema Orion“ on programm, mis kasutab virtuaalsele tegelasele kasutaja liigutuste ülekandmiseks Vive Trackereid. Tegu on tasulise rakendusega, mis nõuab iga installatsiooni kohta aastast litsentsi, kuid testimiseks on võimalik kasutada ka 14-päevast demo [21]. iKinema Orion vajab töötamiseks vähemalt kuute Vive seadet ning toetab maksimaalselt kuni seitset Vive seadet. Orioni puhul on võimalik kasutada erinevaid seadistusi ning asendada ka peaseade ning kontrolleriid Trackeritega. Toetatud on käte, jalgade ning küünarnukkide asukoha jälgimine [22]. Kalibreerimiseks ehk seadmete tuvastamiseks ning sidumiseks virtuaalse tegelasega kasutab Orion automaatset kalibreerimist ning käte, jalgade, selgroo ning kaela pikkuse manuaalset määramist kasutajaliidese abil [23]. Automaatse kalibreerimise korral on peaseadme suhtes defineeritud ristküliku kujulised alad, seadmetele määratakse roll vastavalt sellele, millises ristkülikus tema koordinaadid asuvad [24]. Mänguarenduse seisukohast on Orion mängumootorist eraldiseisev programm, mis analüüsib Trackeritest tulevat andmevoogu ning ennustab selle põhjal kontrollitava tegelase poosi. Nagu enamik teisi inimkeha jälgimise süsteeme kasutab Orion puuduvate liigete imiteerimiseks pöördkinemaatikat. Orioni puhul on tegu ennustusliku mudeliga, seega ei keskenduta kasutaja ning virtuaalse avatari jäsemete asukoha võimalikult täpsele kokkuviiamisele, vaid pigem korrektse poosi ning jäsemete realistliku paindumise imiteerimisele. Seejuures suudetakse vältida jäsemete deformeerumist, mis teistes pöördkinemaatika lahendustes võib liigete pöörderaadiuse ületamisel tekkida. Samas ei järgi virtuaalse tegelase keha sajaprotsendiliselt kasutaja keha, seetõttu võib esineda kuni viiesentimeetrilisi erinevusi

virtuaalse keha ja kasutaja keha vahel. Antud tarkvara ei ole ka võimeline ka kohanduma erineva pikkusega jäsemetega, avatari käte-jalgade pikkused ei vasta kasutaja omadele.

3.2 FinalIK VRIK

FinalIK on Eesti ettevõtte Rootmotion OÜ poolt välja töötatud pöördkinemaatika lahendus. Tegu on üks arvustuste põhjal üks enimkasutatavaid pöördkinemaatika teeke Unity3D mängumootorile, mis kasutab erinevaid pöördkinemaatika algoritme [23]. Üks osa FinalIK teegist, mille nimi on „VRIK“, on suunatud pöördkinemaatika lahendustele virtuaalreaalsuses. VRIK töötati välja koostöös „Gunfire Games“ mängustuudioga VR-mängu „Dead and Buried“ jaoks [24]. VRIK töötab nii Vive baaskomplekti kui ka Trackeritega ning võimaldab asendada mittejärgitavaid kehaosi animatsioonidega (näiteks jalgade IK, kui kasutada ainult Vive baaskomplekti). HTC enda poolt Trackerite jaoks loodud demorakendus, kasutab samuti pöördkinemaatika jaoks FinalIK teeki [25]. Demo võimaldab kuni viie jälgitava punkti kasutamist (2 kontrolleri + 3 Trackerit) ning avatar tuleb iga rakenduse käiviatmise korral uuesti kalibreerida. Kalibreerimise alguses seisab avatar T-posis, vaadates otse, jalad õlgade laiuselt ning käed külgedele välja sirutatud. Kasutaja asetab enda käed ning jalad samadele positsioonidele kui avatari omad ning vajutades kontrolleri nupule seotakse virtuaalne keha seadmete asukohtadega.

4 HTC Vive Trackerite kinnitamine

Vive Trackerite kinnitamisel inimkehale tuleks teha kindlaks, milliseid kehaosi Trackerite abil on võimalik ning otstarbekas järgida ning milliseid kinnitusvahendeid seejuures kasutada. Kinnituspunktide valikust oleneb, milliseid kehaosi saab otseselt siduda järgitavate seadmetega ning milliste jaoks tuleb kasutada pöördkinemaatikat. Kvaliteetse ning stabiilse järgimise tagamiseks on oluline sobivate kinnitusvahendite valik.

4.1 Vahendid Trackerite kinnitamiseks

Trackeritel on esemete külge kinnitamiseks tagaküljel ¼" keermega poltliides. Antud keermega liideseid on kergesti kättesaadavad, neid kasutatakse tihti fotokaamerate kinnitamiseks. Trackerite keha külge kinnitamiseks on kasutatud erinevaid vahendeid nagu näiteks:

1. Velcro- ehk takjapaelad;
2. Küünarnuki- ja põlvekaitsmed;
3. Pingutatavad rihmad;
4. 3D-prinditud kinnitused;
5. Kummipaelad.

Kummipaelad on üks kättesaadavamaid vahendeid, millega Trackereid keha külge kinnitada. Kummipaeltega saab Trackereid kinnitada käte ning jalalabade külge, asetades paelad Trackeri kolme „sarve“ ümber. Piisavalt stabiilse kinnituse saavutamise võib võtta aega ning kinnitused ei ole kummipaelte puhul kahjuks kuigi stabiilsed. Paelad võivad kergesti puruneda või välja venida, mistõttu on vaja neid pidevalt asendada. Liiga tugevate paelte puhul võib hakata pael liigselt soonima ning tekitab kasutajale ebamugavust.

Rihmade kasutamisel torgatakse enamasti rihmast läbi sobiva suurusega auk ning asetatakse sealt läbi sobiv keermekinnitus. Rihmasid on kasutatud koos 3D-prinditud kinnitustega [26], mis muudavad Trackeri kinnitamise lihtsamaks ning lisavad stabiilsust. Seda kinnitusmeetodit kasutatakse enamasti alaselja järgimiseks, kasutades püksirihma. Võimalik on kasutada ka lühema pikkusega rihmasid jäsemete järgimiseks, kuid tihti kasutatakse nende asemel velcro-paelasid, mida on kergem eemaldada.

Küünarnuki- ja põlvekaitsmeid on kasutajamugavuse seisukohast üks paremaid lahendusi, nende kinnitamine kehale on kiire ning kaitsmed enamasti on tootjate poolt ergonoomiliseks disainitud. Selleks, et kaitsmeid ühendada Trackeritega puuritakse neisse augud ning asetatakse neist läbi sobiva keermega kruvid. Kaitsmete välispinnad on kumerad, mistõttu võidakse kasutada ka stabiliseerivaid 3D-prinditud plaate või kõrgendusi. Kaitsmete kasutamisel osutuvad piiranguks tulistamismängud, kus mängija peab varjumiseks võtma sisse põlvelt või lamades laskeasendi. Nõjatudes keharaskusega põlvekaitsme küljes oleva Trackeri peale võib kasutaja seadet tõsiselt kahjustada.

Takjapaelad on MoCap-süsteemides üks enimlevinud kinnitusviise, millega sensoreid või markereid inimkehale kinnitada. Trackerite kinnitamine velcro-paelte abil on tihti kiirem kui rihmade abil ning neid on kergem reguleerida. Takjapaelete kasutamisel aitab stabiilsust lisada plastikust alusplaadi või papitüki kinnitamine paela ning Trackeri vahele. Vive Trackerite kinnitamiseks on USA firma Rebuff Reality töötanud välja „Trackstrap“ velcro-kinnitused, millele on sobiva keermekinnitusega alusplaat sisse õmmeldud. Trackstrap kinnitused on paela pikkuse tõttu kõige sobivamad just jäsemete külge kinnitamiseks.

4.2 Vive Trackerite jaoks sobivad kinnituspunktid

Kinnituspunktides oleneb, milliseid kehaosi on võimalik Trackerite abil jälgida ning avatarile üle kanda. Vive Trackerite puhul on tegu optilise süsteemiga, mis tähendab, et kinnituspunktide valikul tuleb arvestada seadmete keha varju sattumisega. Liigutuste tegemisel võib mõni kehaosa täielikult varjata mõne Trackeri ning seadme asukohaga seotud virtuaalse tegelase kehaosa asukohta pole enam võimalik määrata. Antud olukordade esinemise vähendamiseks tuleb kinnitada Trackerid keha külge sellisel viisil, et Trackerid satuks liigutuste tegemisel võimalikult vähe teiste kehaosade varju.

Kinnituspunktide valikul olulised punktid on lähtuvalt inimese anatoomiast ning autori hinnangust järgmised:

1. Käelabad;
2. Randmed;
3. Küünarnukid;
4. Alaselg;
5. Põlved;
6. Jalalabad;
7. Pöiad.

Pea asukoha järgimiseks kasutatakse enamasti peaseadme koordinaate, kuid animatsioonide salvestamiseks või segareaalsuse rakenduste loomiseks võib peaseadme asendada ka Trackeriga. Heaks näiteks selle kohta on iKinema demo, kus salvestatakse tantsiva naisterahva liigutusi.

Käelabade asukoha järgimiseks kasutatakse enamasti VR-süsteemiga kaasa tulevaid kontrollereid, kuid need võib samuti asendada Trackeritega. Osade virtuaalreaalsuse kinnaste, näiteks „Noitom Hi5 VR Glove“ või „ManusVR“ puhul, kus käelaba asukoha määramiseks kasutatakse randmele kinnitatud Trackerit. Randmele kinnitatud Trackerite kinnitamisel on autori hinnangul oluline kinnitada seade randme välisküljele, sest käeliigutuste tegemisel randme sisekülj satub liigutuste tegemisel rohkem varju.

Küünarnukkide ning põlvede asukoha järgimine Vive seadmete abil muudab autori kogemusest lähtuvalt virtuaalse tegelase käte ning jalgade liigutused oluliselt realistlikumaks, sest nende asukoht sõltub füüsilise seadme asukohast mitte pöördkinemaatika mudelist. Sellisel juhul tuleb pöördkinemaatikat kasutades arvutada välja ainult puusa ning õlgade asukoht ning nurk ruumis.

Alaselja asukoha järgimine on kasulik ka juhul, kui kasutatakse ainult ühte Vive Trackerit. Vive baaskomplekti kasutades pöörab virtuaalne tegelane tavaliselt samas suunas kuhu kasutaja vaatab. Alaseljale kinnitatud Trackeri abil saab täpsemalt määrata

virtuaalse tegelase asukohta ning pöördenurka ruumis võimaldades kasutajal pöörata pead ilma et keha kaasa liiguks.

Sääre ning jalalabade asukoha järgimiseks võib kinnitada Trackerid näiteks takjapaelte abil ümber jalalaba või sääre alaosa ehk põia. Trackerite põia ümber kinnitamisel on keeruline määrata jalalaba nurka ruumis, mistõttu jalalaba ei pruugi mõjuda piisavalt realistlikuna. Jalalaba külge kinnitatud Trackerite abil on võimalik autori hinnangul küllaltki realistlikult tagada virtuaalse tegelase jalalabade ning sääre nurk ning asukoht ruumis.

5 Tehniline teostus

Eesmärgiks on luua 3D-stseen, kus kasutaja saaks kalibreerida ning siduda enda keha virtuaalse tegelase omaga. Seejärel saaks kasutaja testida kuivõrd realistlikult tegelase keha järgib kasutaja keha liikumist ning erinevaid kehapoose. Kasutajale peaks saama paremaks testimiseks näha ennast peegelpildis, vaadata kasutajaliidesest logi süsteemi sündmuste kohta, salvestada ning laadida varasemaid kalibreerimisi ehk profile.

5.1 3D-stseeni loomine

Tuleb luua mängumaailm, milles on võimalik vaadelda tegelast esimeses isikus ning ka peegelpildis. Samuti on oluline logi kuvamine vajalike tegevuste kohta nagu leitud seadmete arv ning seadmetele määratud rollid.

Stseeni loomisel vajalikud sammud on järgnevad:

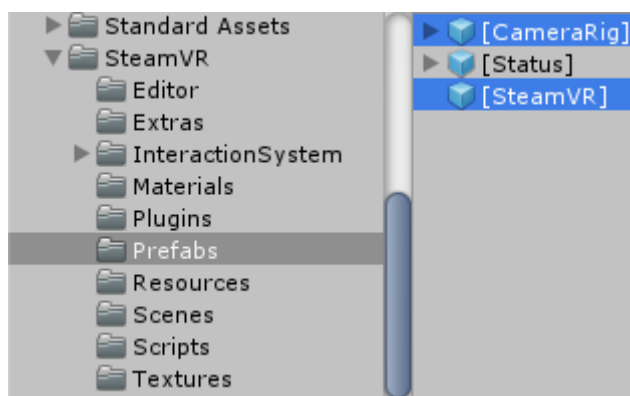
1. SteamVR teegi import ning vajalike komponentide lisamine stseeni;
2. Virtuaalse tegelase lisamine stseeni;
3. Peegel, millest kasutaja saab avatari liikumist vaadelda;
4. Kasutajaliides logi kuvamiseks.

Peale uue Unity3D projekti loomist on vaja lisada projektile SteamVR teek, mis võimaldab stseeni vaadelda läbi HTC Vive peaseadme. SteamVR teek on tasuta ning seda on võimalik importida „Unity Asset Store“ veebipoest. Veebipoe avamiseks otse Unity3D töölauarakenduses tuleb valida ülemisest valikumenüüst „Window“ ja seejärel rippmenüüst „Asset Store“. Sisestades avanenud akna otsingukasti märksõna „SteamVR“ kuvatakse otsingutulemused, mille hulgast tuleb valida „SteamVR Plugin“, mille avaldajaks on Valve Corporation.

Järgneval lehel tuleb vajutada nuppu „Download“, mille peale teek laaditakse Unity3D mängumootori juurkausta. Juurkausta laadimine võimaldab kasutada mõne teise

virtuaalreaalsuse projekti loomisel kasutada juba arvuti kõvakettal olevat versiooni ning seetõttu pole vajalik enam teegi failide allalaadimine veebipoe serverist. Seejärel tuleb vajutada nuppu „Import“, mis avab hüpikakna, kus on välja toodud kõik projekti imporditavad skriptid, 3D-mudelid ning muud komponendid. Hüpikakna alaosas oleva nupu „Import“ vajutamisel imporditakse projekti kausta kõik valitud komponendid. Seejärel avatakse uus hüpikaken, mis pakub võimalusi Unity3D seadete kohandamiseks virtuaalreaalsuse rakenduste jaoks. Kõik väljatoodud seaded on vajalikud loodava VR-rakenduse toimimiseks seega tuleb vajutada nupul „Accept All“.

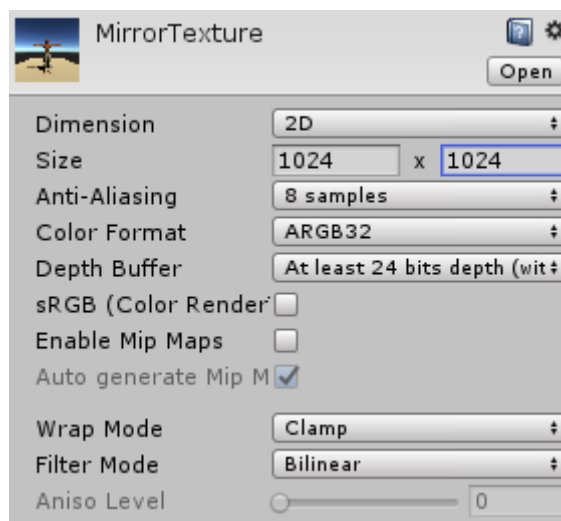
Tühja stseeni tuleb lisada kaks komponenti importimisel loodud „Prefab“ kaustast, milleks on „[CameraRig]“ ning „[SteamVR]“ ning stseenist tuleb kustutada automaatselt lisatud „Main Camera“, vt Joonis 5.



Joonis 5. SteamVR komponendid projekti kaustas

Virtuaalse tegelase valikult kasutab autor Unity3D veebipoes saadaolevaid tasuta 3D-mudeleid. Üheks selliseks „Morph3D“ poolt loodud „MCS Male“, mis võimaldab tegelast muuta eriilmelisemaks kasutades erinevaid riideesemeid ning näo- ning kehakuju muutmist. Kasutatava tegelase import toimub samalaadselt SteamVR teegi impordiga. Seejärel tuleb valida sobiv komponent asukohast „MORPH3D“->“Content“->“M3Dmale“->“Figure“->“M3Dmale“ ning lohistada see stseeni vaatesse.

Kasutaja jaoks on oluline näha avatari keha peegelpildis, sest peegel abistab kalibreerimisel ning avatari keha paindumise testimisel. Peegli jaoks vajaliku kujutise saamiseks tuleb lisada stseeni hierarhia paneeli uus kaamera, valides paremklikiga „Camera“, mille nimeks paneme „Mirror Camera“. Samuti loome projekti kataloogi alla uue tekstuuri, mille nimeks paneme „MirrorTexture“. Tekstuuri seadetes määrame tekstuuri suuruseks 1024x1024 pikslit, et peeglikujutis oleks piisavalt kõrge resolutsiooniga. Tekstuuri seaded on kujutatud Joonisel 6.



Joonis 6. Tekstuuri seadistused

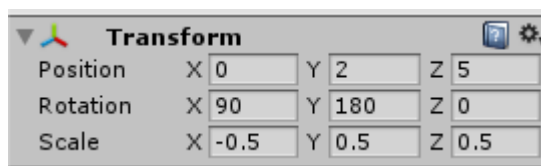
Seejärel valime varem loodud kaamera ning määrame seadetes „Target Texture“ valides hüplikaknast eelnevalt tekitatud tekstuuri. Samuti määrame kaamera asukohaks 1.2 ühikut Y-teljel ning 3 ühikut Z-teljel ning pöörame kaamerat 180 kraadi võrra Y-teljel, et see osutaks tegelase poole. Kaamera seadistust on kujutatud Joonisel 7.



Joonis 7. Avatari peegelpildis kuvamiseks kasutatava kaamera seadistused

Peegli tasapinna lisamiseks tuleb teha paremkliik stseeni hierarhia paneelil ning valida „3D Object“ -> „Plane“. Loodud tasapinna puhul tuleb määrata sobiv asukoht, pöördenurk ning suurus. Tasapinna asukohaks määrame 2 ühikut Y-teljel ning 5 ühikut Z-teljel ning pöörame tasapinda 90 kraadi võrra X-teljel ning 180 kraadi võrra Y-teljel. Tasapinna suuruseks määrame 0.5 ühikut, et peegelpilt ei oleks kasutaja jaoks liialt suur. Seejuures

määrame X-telje suuruse negatiivseks, et kaamerapilt ei oleks pöördvaates. Seadistust on kujutatud Joonisel 8.



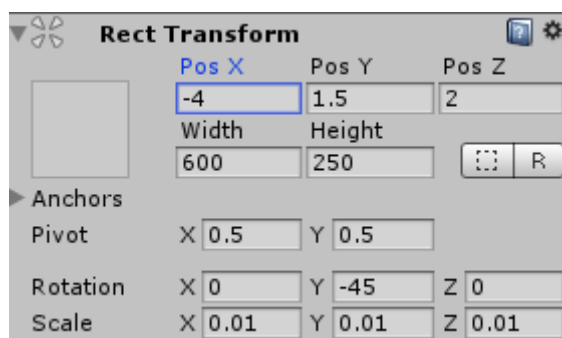
Joonis 8. Peegli jaoks kasutatava tasapinna asukohta ning suuruse seadistused

Määrame ka tasapinna varjutaja valides varjutaja tüübiks „Unlit/Texture“ ning määrame tekstuuriks „MirrorTexture“, misjärel tasapinnal kuvatakse kaamera poolt renderdatav pilt, vt Joonis 9.



Joonis 9. Peegli jaoks kasutatava tasapinna varjutaja seadistused

Logi kuvamiseks tuleb luua kasutajaliides, milles kuvatakse keritavat teksti ning süsteemi poolt leitud seadmete arvu. Selleks loome uue „Canvas“ elemendi tehes stseeni hierarhia paneelil paremkliki ning valides „UI“->„Canvas“. Määrame Canvas elemendile sobiva suuruse ning asukohta, et see oleks kasutajale loetav, kuid ei varjaks liigselt vaatevälja. Antud seadistust on kujutatud Joonisel 10.



Joonis 10. Kasutajaliidese paneeli seadistused

Seejärel lisame Canvase alamelemendiks „Panel“ tüüpi elemendi, millest saab kasutajaliidese taust. Paneli alamelemendiks lisame omakorda „Scroll View“, mis võimaldab keritavate elementide loomist. Eemaldame linnukese Scroll View komponendi

„Scroll Rect“ seadistuse „Horizontal“ puhul, sest paneel on piisavalt suur ka pikemate logitekstide kuvamiseks. Samuti eemaldame Scroll View alamelemendi „Horizontal Scrollbar“. Seejärel lisame Scroll View alamkomponendile „Content“ komponendi „Text“, mille väärtust uuendame hiljem skriptide abil.

5.2 Ühendatud seadmete tuvastamine

Inimkeha järgimise jaoks on vajalik tuvastada, millised seadmed hetkel Vive virtuaalreaalsuse süsteemis töötavad ning milline on nende olek. Seejärel on võimalik määrata igale seadmele tema asukoha alusel millise avatari kehaosaga seade peaks olema seotud. Vajaliku informatsioon hetkel Vive virtuaalreaalsuse süsteemis töötavate seadmete kohta on kättesaadav SteamVR teegi kaudu. Olulised parameetrid, mida on vajalik tuvastada:

1. Indeks seadmete nimekirjas;
2. Seadme klass;
3. Seadme olek.

Indeks seadmete nimekirjas võimaldab hiljem seadme poole pöörduda seadme asukoha määramiseks ruumis. Vive virtuaalreaalsuse süsteemi puhul on igale seadmele määratud vastav klass. Arvestamata täiendavaid lisaseadmeid, võib ühendatud seade kuuluda ühte järgnevatest klassidest:

1. HMD (peaseade);
2. Controller (kontroller);
3. GenericTracker (Tracker seade);
4. TrackingReference (baasjaam).

Meid huvitavad seadmed on hetkel kontrollerid ning Trackerid, baasjaamade asukohad ei ole hetkel olulised ning peaseadme asukoht on leitav põhikaamera asukoha järgi. SteamVR teegi kaudu on võimalik tuvastada ka seadme olek ning välistada edasisest töötlusest seadmed, mis ei ole aktiivsed või mille järgimine on häiritud. Tulemusena

saame teatmik tüüpi objekti, mida on võimalik kasutada kehaosasid kirjeldavate rollide määramisel. Antud tegevust esitav programmikood asub Lisas 1.

Vive seadmete puhul võib tekkida häiritus, mistõttu seade hangub ning ei liigu enam virtuaalses ruumis või tekib suur erinevus seadme tegeliku asukoha ning virtuaalse asukoha vahel. Häirituse põhjuseid on mitmeid, enamasti on selleks seadme varju sattumine või liialt valgust peegeldavate esemete olemasolu. Seadmete arvu kahanemisel tühjendatakse SteamVR teegi siseselt kogu eelnev seadmete nimekiri ning seadmete indeksid võivad sellest muutuda. Kohandamiseks seadmete arvu muutumisega kontrollime seadmete arvu teatud ajavahemiku järel. Võrdleme seadmete arvu praegusel ajahetkel eelneva kontrollimise käigus saadud tulemustega. Juhul kui seadmete arv on vähenenud, tuvastame mõjutatud seadmed, logime seadmete info ja määrame seadmete rollid uuesti.

5.3 Seadmetele rolli määramine

Selleks, et Vive seadmeid saaks siduda pöördkinemaatika mudeliga ning kasutada neid virtuaalse tegelase keha liigutamiseks, tuleb määrata igale seadmele sobiv roll. Autori poolt valitud rollid on järgnevad:

1. Head (Pea);
2. LowerBack (Alaselg);
3. RightHand (Parem käelaba);
4. LeftHand (Vasak käelaba);
5. RightElbow (Parem küünarnukk);
6. LeftElbow (Vasak küünarnukk);
7. RightFoot (Parem jalalaba);
8. LeftFoot (Vasak jalalaba);
9. RightKnee (Parem põlv);
10. LeftKnee (Vasak põlv).

Inimkeha külge kinnitatud Vive seadmetele rolli määramiseks peab kasutaja seisma sobivas poosis. Täiskeha jälgimise puhul kasutatakse enamasti kahte levinud kalibreerimispoosi, niinimetatud T-poosi ning A-poosi. T-poosis seisab kasutaja sirgelt, jalad õlgade laiuselt, pilk otse ette suunatud ning käed on õlgade kõrguselt välja sirutatud. A-poos erineb T-poosist sellepoolest, et käed ei ole õlgade kõrgusel vaid on suunatud umbes 40 kraadi alla [27].

Seadmetele korrektsete rollide määramiseks peab kasutaja astuma samale positsioonile kui virtuaalne tegelane ning sisse võtma kalibreerimispoosi ja vajutama kontrolleri päästikule. Seejärel kasutatakse kontrolleri ning Trackerite asukohta rollide määramiseks.

Rollide määramisel on oluline seadme piki- ja põikikaugus peaseadme asukohast ning seadme kõrgus ehk asukoht vertikaalteljel. Pea asukoha ning nurga saab määrata Vive peaseadme järgi. Vajadusel tuleb tegelase pea asukohta peaseadme suhtes veidi muuta, lisades pikiteljel 5-10 sentimeetrise nihke. Vastasel korral renderdatakse vaateväli läbi tegelase pea, mistõttu vaateväli on kasutaja jaoks oluliselt piiratud. Näide läbi tegelase pea renderdatud vaateväljast on kujutatud Joonisel 11.



Joonis 11. Näide läbi tegelase pea renderdatud vaateväljast

Ülejäänud kehaosade asukoha määramisel on oluline teada, millised seadmed on seotud parema või vasaku kehapoolega. Loogiliselt järeldades on vastava kehapoolega seotud need seadmed, mis asuvad peaseadmes vasakul või paremal. Tuvastamiseks, milline on kasutaja vasak või parem pool kasutame lineaarvõrrandit.

Selleks määrame kaks punkti, esimeseks neist on põhikaamera asukoht ning teiseks punktiks määrame punkti, mis asub kaamera asukohast ühe meetri kaugusel kaamera vaatesuunas. Arvutame võrrandi tõusu jagades saadud punktide piki- ja põikikoordinaadid omavahel. Seejärel kasutame valemit

$$y = x - x_1 - m(z - z_1)$$

kus x_1 ja z_1 on põhikaamera piki- ja põikikoordinaadid, m on võrrandi tõus ning x ja z on Vive seadme piki- ja põikikoordinaadid. Tulemuseks on märgiga kaugus peaseadmest, negatiivse kaugusega seadmed asuvad peaseadmest vasakul ning positiivse märgiga seadmed paremal. Seadme vasakul või paremal kehapoolel asumise tuvastamist esitab programmikood Lisas 2.

Järgnevalt määrame igale seadmele konkreetse rolli. Käsitleme rollide määramisel alakeha ning ülakeha kinnituspunkte eraldi, sest ülakeha puhul on vajalik kinnituspunktide asukohtade põhjalikum omavaheline võrdlus.

Alakeha puhul määrame jalalabade ning põlvede rolli sõltuvalt nende asukohast vertikaalteljel ning eelnevalt arvatud kaugusest peaseadmest. Leiame seadme ning põhikaamera asukoha suhte vertikaalteljel ning võrdleme seda kahe konstantse väärtusega. Need väärtused on määratud autori poolt võttes arvesse Austraalias läbi viidud uuringut [28], milles leiti keha pikkuse ning põlvekõrguse suhteks 26.4...33.9 protsenti. Võttes arvesse neid tulemusi määrame, et kõigi seadmete puhul, mille seadme ning peaseadme suhe on väiksem kui 0.2, on tegu jalalabadega ning mis jäävad vahemikku 0.2...0.35 on tegu põlvedega. Antud tegevust esitab programmikood Lisas 3.

Ülakeha rollide määramisel tuleb arvestada et samal kõrgusevahemikus võib asuda mitu erineva rolliga seadet. Kõigepealt teeme kindlaks, millise seadme puhul on tegu vöökohaga. Inimesed võivad kanda Trackerit vöökohal erineval viisil, nii seljal, kõhul kui ka puusal. Korrekse kalibreerimispoosi korral saame siiski leida õige seadme, milleks on kõige madalamal ja peaseadmele kõige lähemal asuv seade. Eemaldades vaadeldavate seadmete hulgast vöökoha rolliga seadme, saame järgnevalt määrata rolli küünarnukkidele, mis on peaseadmele vasakul ning paremal kauguselt lähimad. Seejärel ülejäänud seadmetel saab olla vaid üks võimalik roll, milleks on vasak või parem käelaba. Leitud rollide põhjal on võimalik järgnevalt määrata, millised seadmed tuleb siduda vastavate virtuaalse tegelase kehaosadega.

5.4 Pöördkinemaatika algoritmi valik

Unity3D mängumootori puhul on olemas kaks põhilist pöördkinemaatika lahendust, mis on piisavalt arenenud ning mida on küllaltki lihtsalt virtuaalsete tegelaste animeerimiseks kasutada. Nendeks on Unity3D sisseehitatud „Mecanim IK“ ning Rootmotion OÜ poolt väljatöötatud FinalIK, mis on tasuline. Need lahendused toimivad hästi tavaliselt kasutatavate pöördkinemaatika probleemide puhul, kuid nende puhul esineb mitmeid puuduseid, mis ei võimalda autori poolt väljapakutud lahendust nende abil rakendada. Nendeks puudusteks on:

1. Olemasolevat lahendust ei ole võimalik laiendada või laiendamine on keerukas;
2. Puudub võimalus põlvedele või küünarnukkidele pöördkinemaatika sihtmärgi lisamiseks;
3. Tasulised lahendused.

Antud puudused tulenevad autori hinnangul asjaolust, et olemasolevad lahendused on suunatud pigem üldlevinud pöördkinemaatika mudelitele, kus kasutatakse sihtmärgina ainult käsi ning jalalabasid. Pöördkinemaatika sihtmärk on positsioon, kuhu pöördkinemaatika ahela viimane lüli üritab jõuda, enamasti on viimaseks lüliks inimeselaadsete tegelaste puhul randmed ning põiad. Autori poolt väljapakutud mudel on üldlevinud mudelist mõnevõrra erinev, sest kasutab ka põlvede ning küünarnukkide puhul pöördkinemaatika sihtmärke.

Nii Mecanim IK kui ka FinalIK puhul on põhiliseks probleemiks põlvede ning küünarnukkide jaoks pöördkinemaatika sihtmärkide lisamine. Unity3D sisseehitatud Mecanim IK puhul ei ole võimalik lisada uusi sihtmärke, sest sihtmärgid sõltuvad sisseehitatud tüübist „AvatarIKGoal“, mida pole võimalik laiendada [29]. FinalIK puhul on virtuaalreaalsuse rakenduste jaoks loodud eraldi komponent VRIK, mille puhul on võimalik põlvede ja küünarnukkide puhul kasutada „Bend Goal“ nimelisi lisasihtmärke, mis muudavad küünarnukkide ja põlvede paindenurka, kuid ei taga, et küünarnuki asukoht on alati õige. Veebipoest allalaadimisel tuleb kaasa kogu FinalIK teegi lähtekood, seega on küünarnukkidele ning põlvedele sihtmärkide lisamine võimalik, kuid tähendaks autori hinnangul liialt suure osa olemasoleva koodi ümberkirjutamist.

Lähtudes eelnevalt nimetatud pöördkinemaatika süsteemide puudustest on autor otsustanud luua virtuaalse tegelase kehaosade liigutamiseks enda pöördkinemaatika mudeli. Inimese jäsemete liigutamiseks modelleerimisprogrammides ning teistes mängumootorites kasutatakse tihti kolmest punktist koosnevate pöördkinemaatika mudelite kombinatsioone. Kolmest punktist koosnev pöördkinemaatika mudel on kõige lihtsam pöördkinemaatika mudel, mida kasutatakse tihti robotikas kolmest teljest koosnevate tööstusrobotite juhtimiseks. Lähtudes asjaolust, et mitmetes modelleerimisprogrammides kasutatakse tegelaste animeerimiseks kolmest punktist koosnevaid pöördkinemaatika mudelite kombinatsioone, on autori valik kasutada sama lähenemist.

Virtuaalse tegelase keha liigutamiseks vajalik pöördkinemaatika mudel koosneb seega viiest komponendist, mis vastutaksid käte, jalgade ning selgroo liigutamise eest. Iga komponendi liigutamiseks Vive kontrolleri või Tracker seadme abil on vajalik pöördkinemaatika sihtmärgi olemasolu. Sihtmärk on enamasti viimane lüli pöördkinemaatika ahelas, mille positsioonist lähtuvalt arvutatakse ka teiste ahela lülide nurk ning asukoht. Inimkeha puhul oleksid viimasteks lülideks randmed, põiad ning pea. Selleks, et täpselt järgida küünarnukkide asukohta, on vaja lisada sihtmärgid ka küünarnukkidele ning põlvedele. Selgroo puhul on oluline ka vöökoht, sest sellest oleneb tegelase keha paindenurk ning asukoht.

Autori poolt välja pakutud pöördkinemaatika mudel on esitatud järgnevalt tabeli kujul.

Tabel 2. Pöördkinemaatika mudel

Pöördkinemaatika komponent	Jälgitavad kehapunktid	avatari	Komponendi sihtmärgid
Selgroog	Pea, rind, vöökoht		Pea, vöökoht
Vasak käsi	Vasak õlg, vasak küünarnukk, vasak ranne		Vasak küünarnukk, vasak ranne
Parem käsi	Parem õlg, parem küünarnukk, parem ranne		Parem küünarnukk, parem ranne

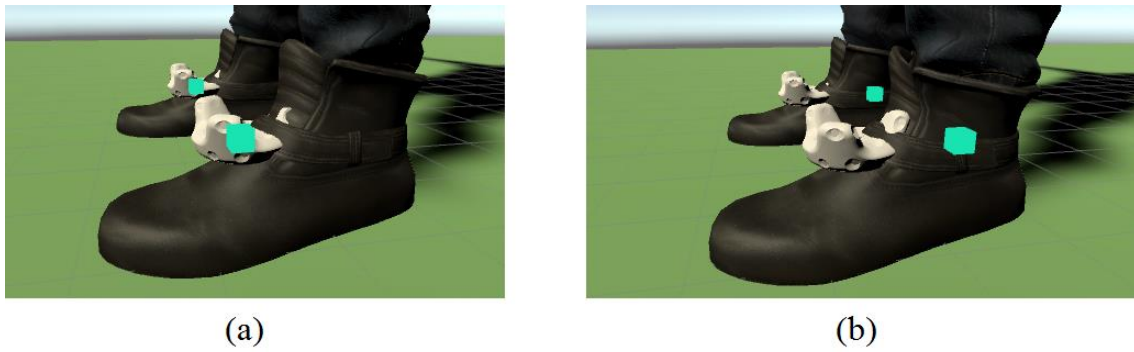
Vasak jalg	Vasaku reie ülaosa, vasak põlv, vasak põid	Vasak põid, vasak põlv
Parem jalg	Parema reie ülaosa, parem põlv, parem põid	Parem põid, parem põlv

5.5 Pöördkinemaatika komponentide lisamine virtuaalsele tegelasele

Autori valik on kasutada virtuaalse tegelase kehaosade liigutamiseks viite pöördkinemaatika komponenti, mis koosnevad kolmest tegelase kehaosast ning kahest sihtmärgist. Komponente saab lisada virtuaalse tegelase külge käsitsi või kasutada ära Unity3D automaatset luustruktuuri tuvastamist, kui tegelase skeleti tüübiks on valitud „Humanoid“. Automaatne tuvastamine pole võimalik kui tegelase luustruktuur on mõnevõrra erinev tavapärasest, näiteks on tegelaseks kentaur. Kasutades ära Unity3D mängumootori poolt tuvastatud tegelase luid, saame uuele mängutegelasele automaatselt lisada vajalikud komponendid korrektsete luude ning sihtmärkidega. Autori poolt valitud luude tuvastamise meetod on sõltuv Unity3D vahenditest ning programmi töötamise ajal uusi mudeleid lisada pole võimalik. Selle võimaldamiseks tuleks luua võimalus ise automaatselt tegelase luid tuvastada. Automaatne pöördkinemaatika komponentide lisamine toimub programmi käivitamisel ning selle tulemusena lisatakse virtuaalsele tegelasele viis komponenti, mis esialgu on mitteaktiivsed. Komponentid muudetakse aktiivseks virtuaalse tegelase keha sidumisel Vive seadmetega.

5.6 Virtuaalse tegelase keha sidumine seadmetega

Esimesel kalibreerimisel on vajalik astuda tegelasega samale positsioonile, et oleks võimalik määrata seadmete ning tegelase liigete vaheline kaugus. Seejärel liigutame eelnevalt loodud pöördkinemaatika sihtmärgid vastavate liigete positsioonidele ning määrame nad vastavate seadmete alamelementideks säilitades nende nurga ning positsiooni. Muutes pöördkinemaatika komponendid aktiivseks, hakkab avatari keha järgima Vive seadmete asukohta ruumis. Pöördkinemaatika sihtmärkide sidumist tegelase kehaga on visualiseeritud Joonisel 12 enne (a) ning pärast (b) sidumist.



Joonis 12. Pöördkinemaatika sihtmärgid enne (a) ning pärast (b) sidumist virtuaalse tegelasega

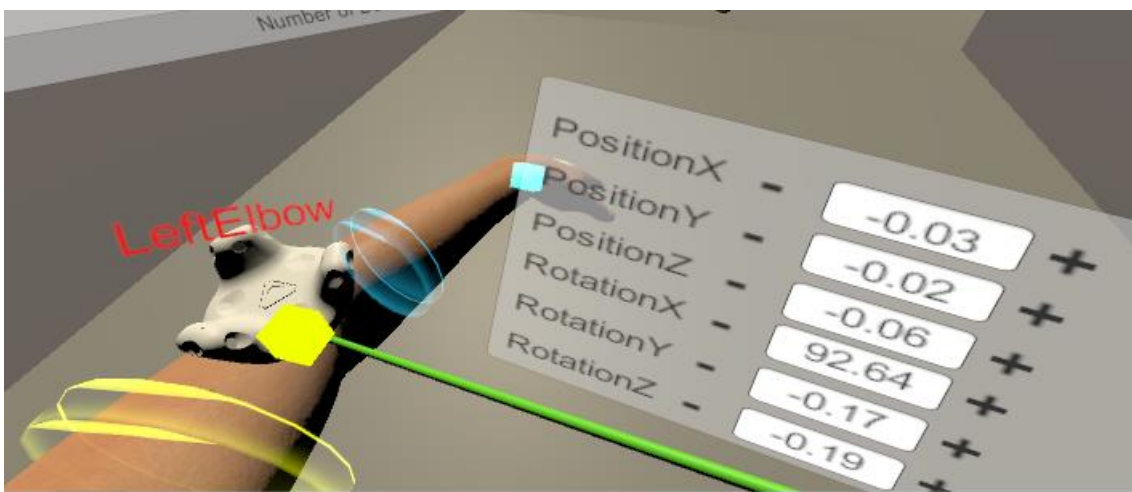
Esmase kalibreerimise järel leitud pöördkinemaatika sihtmärkide nihet Vive seadmete suhtes on võimalik kasutada avataride vahetamisel ning programmi taaskäivitamisel koheseks tegelase keha sidumiseks seadmetega.

Igal Vive seadmel on enda tehase poolt antud identifikaator, mis võimaldab salvestatud info põhjal koheselt määrata seadmele rolli ning seega siduda seadmed virtuaalse tegelasega. Selle meetodi puuduseks on asjaolu, et kasutaja võib kasutuskordade vahel seadmeid omavahel vahetada, mistõttu tegelane võtab programmi käivitamisel sisse ebaloomuliku poosi. Võimalus oleks luua valikuline süsteemiseadistus, mis iga käivitamise või kasutajaprofiili vahetamise korral paluks kasutajal sisse võtta kalibreerimispoos vabalt valitud suunas, mis võimaldaks seadmetele uuesti rollid määrata. Autori hinnagul oleks lahenduseks ka seadmete füüsiline markeerimine vastavate siltidega või kleebistega.

5.7 Tegelase keha ning jäsemete pikkuse kohandamine

Igal kasutajal on erinevad kehaproportsioonid ning virtuaalsete avataride puhul on märgatav erinevus avatari ning kasutaja keha ning jäsemete pikkuse vahel. Realistlikult mõjuva avatari saavutamiseks peaks avatari keha kohanduma kasutaja keha omaga. Keha pikkuse kohandamiseks kasutame tegelase silmade kõrgust, sest sellisel kõrgusel peaks asuma põhikaamera. Leides kaamera kõrguse ning tegelase esialgse silmade kõrguse suhte saame kindlaks teha kui palju tuleks tegelase mudelit suurendada. Kõrguse arvutamine on kalibreerimise faasis pidev, see tähendab, et kui kasutaja kükitab muutub ka tegelase kõrgus. Käte pikkuse kohandamiseks kalibreerimise ajal peab kasutaja käed maksimaalselt välja sirutama, misjärel tegelase jäsemeid pikendatakse.

Peale Vive seadmete sidumist virtuaalse tegelasega võib kasutaja soovida täpsemalt määrata, kus asuvad tema kehaosad kontrollrite ning Trackerite suhtes. Selleks saab muuta kasutaja pöördkinemaatika sihtmärkide asukohtasid ning nurkasid, valides kontrollriga vastava sihtmärgi ning liigutades seda ruumis. Sihtmärgi nurga määramiseks tekitatakse sihtmärgi lähedusse ketas, mille abil saab pöörata liigest ümber oma telje. Vajadusel saab sihtmärgi asukohta ning nurka kontrollrite ning Tracker seadmete suhtes muuta ka kasutajaliidese abil, mis kuvatakse kasutaja vaateväljas sihtmärgi valimisel. Pöördkinemaatika sihtmärkide asukoha kohandamist on kujutatud Joonisel 13.



Joonis 13. Tegelase jäsemete pikkuse ning nurga kohandamine peale automaatset kalibreerimist

5.8 Kasutajaprofiilide laadimine ning salvestamine

Kasutaja jaoks on tülikas igal programmi käivitamise korral uuesti keha kalibreerida. Selle lahenduseks oleks kasutajal lubada salvestada kalibreerimise tulemusi ning valida sobiv seadistus varasemate kalibreerimiste hulgast. Nimetame edaspidi kasutaja keha kalibreerimise tulemust kasutaja profiiliks, sest tegu on kasutaja kehamõõtmest tuleneva seadistusega. Kasutajate profiilid salvestame tekstifaili JSON formaadis. JSON on laialt levinud failiformaat, mida kasutatakse andmete hoidmiseks kompaktsel, kuid inimloetaval kujul. Unity3D mängumootori sisseehitatud funktsioonid andmete JSON formaadis serialiseerimiseks ei toeta kahjuks Unity3D kohavektorite ega kvaternionide serialiseerimist, seega kasutame andmete serialiseerimiseks Unity Asset Store veebipoes saadaolevat tasuta laiendust nimega „Json & MessagePack Serialization”. Salvestatavad väärtused on:

1. pöördkinemaatika sihtmärkide asukoht ning nurk Vive seadmete suhtes;
2. kasutaja nimi;
3. salvestamise kuupäev ning kellaeg.

Profiili salvestamine toimub esimesel kalibreerimisel või sihtmärkide asukoha muutmisel Vive seadmete suhtes. Rakenduse käivitamisel vastavalt valitud seadetele laaditakse ajaliselt viimati salvestatud profiil või palutakse kasutajal valida sobiv profiil vastavast nimekirjast.

6 Kokkuvõte

Antud lõputöö eesmärgiks oli luua programm HTC Vive virtuaalreaalsuse süsteemile kasutades Unity3D mängumootorit milles kasutaja saaks kasutada HTC Vive Tracker seadmeid, et kontrollida virtuaalset tegelast. Võrreldi erinevaid olemasolevaid süsteeme inimkeha liikumise jälgimiseks virtuaalreaalsuses, sealhulgas ka Vive Trackereid. Töö tulemusena valmis rakendus, mis võimaldab kanda üle kasutaja keha külge kinnitatud HTC Vive kontrollerite ning Tracker seadmete asukoha ning liikumise üle virtuaalsele tegelasele. Rakendus töötab erineva arvu Vive seadmete korral ning võimaldab seadmeid inkrementaalselt juurde lisada. Seadmete asukohtade põhjal ruumis seotakse need virtuaalse tegelase kehaosadega ning kasutatakse seejärel pöördkinemaatikat tegelase keha liigutamiseks võimalikult realistlikul viisil. Kasutajal on võimalik vaadelda ennast peegelpildis ning jälgida kuidas tegelase keha paindub erinevate kehapooside korral. Rakendus võimaldab ka salvestada iga kasutaja puhul seadmete asukohtade põhjal kasutajaprofiile, mida saab kasutada virtuaalse tegelase keha sidumisel kasutaja omaga rakenduse taaskäivitamisel. Kokkuvõtvalt võib öelda, et loodud rakendust on võimalik kasutada erinevates tehnoloogiaharudes, näiteks animatsioonide salvestamisel filmitööstuses ning sotsiaalsete virtuaalreaalsuse rakenduste korral.

7 Kasutatud kirjandus

- [1] V. hardware. [Võrgumaterjal]. Available: <https://www.vive.com>. [Kasutatud 27.11.2017].
- [2] HTC. [Võrgumaterjal]. Available: <https://www.vive.com/eu/vive-tracker/>. [Kasutatud 27.11.2017].
- [3] Microsoft. [Võrgumaterjal]. Available: <https://developer.microsoft.com/en-us/windows/kinect/hardware>. [Kasutatud 27.11.2017].
- [4] Microsoft. [Võrgumaterjal]. Available: <https://msdn.microsoft.com/en-us/library/hh973074.aspx>. [Kasutatud 27.11.2017].
- [5] RoadtoVR. [Võrgumaterjal]. Available: <https://www.roadtovr.com/driver4vr-allows-kinect-emulate-vive-trackers-cheap-body-tracking/>. [Kasutatud 27.11.2017].
- [6] Microsoft. [Võrgumaterjal]. Available: <https://www.xbox.com/en-US/xbox-one/accessories?xr=shellnav>. [Kasutatud 26.11.2017].
- [7] OptiTrack. [Võrgumaterjal]. Available: <http://optitrack.com/motion-capture-virtual-reality/>. [Kasutatud 27.11.2017].
- [8] OptiTrack. [Võrgumaterjal]. Available: https://v20.wiki.optitrack.com/index.php?title=Active_Marker_Tracking:_Firmware_v1.0. [Kasutatud 28.11.2017].
- [9] OptiTrack. [Võrgumaterjal]. Available: <https://www.optitrack.com/products/slim-13e/indepth.html>. [Kasutatud 27.11.2017].
- [10] OptiTrack. [Võrgumaterjal]. Available: <http://optitrack.com/products/slim-13e/>. [Kasutatud 27.11.2017].
- [11] OptiTrack. [Võrgumaterjal]. Available: <http://optitrack.com/products/motion-capture-markers/>. [Kasutatud 28.11.2017].
- [12] Noitom. [Võrgumaterjal]. Available: https://www.neuronmocap.com/products/perception_neuron. [Kasutatud 27.11.2017].
- [13] UploadVR. [Võrgumaterjal]. Available: <https://uploadvr.com/perception-neuron-review/>. [Kasutatud 27.11.2017].
- [14] H. L. P. V. Daniel Roetenberg. [Võrgumaterjal]. Available: <https://www.computer.org/csdl/proceedings/ismar/2003/2006/00/20060268.pdf>. [Kasutatud 27.11.2017].
- [15] Noitom. [Võrgumaterjal]. Available: <https://neuronmocap.com/content/product/18-neuron-alum-edition>. [Kasutatud 27.11.2017].
- [16] HTC. [Võrgumaterjal]. Available: <https://blog.vive.com/us/2017/11/16/vive-tracker-bundles-time-holidays/>. [Kasutatud 27.11.2017].

- [17] RoadToVR. [Võrgumaterjal]. Available: <https://www.roadtovr.com/analysis-of-valves-lighthouse-tracking-system-reveals-accuracy/>. [Kasutatud 27.11.2017].
- [18] UploadVR. [Võrgumaterjal]. Available: <https://uploadvr.com/developer-sticks-vive-tracker-oculus-rift-enables-foot-tracking/>. [Kasutatud 28.11.2017].
- [19] HTC. [Võrgumaterjal]. Available: <https://www.vive.com/us/accessory/>. [Kasutatud 28.11.2017].
- [20] FastCoDesign. [Võrgumaterjal]. Available: <https://www.fastcodesign.com/90147868/exclusive-microsoft-has-stopped-manufacturing-the-kinect>. [Kasutatud 28.11.2017].
- [21] iKinema. [Võrgumaterjal]. Available: <https://ikinema.com/orion>. [Kasutatud 27.11.2017].
- [22] iKinema. [Võrgumaterjal]. Available: <https://ikinema.com/index.php?mod=documentation&show=317&id=351>. [Kasutatud 27.11.2017].
- [23] iKinema. [Võrgumaterjal]. Available: <https://ikinema.com/index.php?mod=documentation&show=317&id=371>. [Kasutatud 08.12.2017].
- [24] iKinema. [Võrgumaterjal]. Available: <https://ikinema.com/index.php?mod=documentation&show=317&id=361>. [Kasutatud 08.12.2017].
- [25] R. OÜ. [Võrgumaterjal]. Available: <https://www.assetstore.unity3d.com/en/#!/content/14290>. [Kasutatud 27.11.2017].
- [26] Facebook. [Võrgumaterjal]. Available: <https://developer.oculus.com/blog/developer-perspectives-character-animation-in-dead-and-buried/>. [Kasutatud 27.11.2017].
- [27] J. Bear. [Online]. Available: https://github.com/JamesBear/vive_ik_demo/. [Accessed 27.11.2017].
- [28] Thingiverse. [Võrgumaterjal]. Available: <https://www.thingiverse.com/thing:2213567>. [Kasutatud 27.11.2017].
- [29] OptiTrack, „OptiTrack Documentation,“ [Võrgumaterjal]. Available: https://v20.wiki.optitrack.com/index.php?title=Skeleton_Tracking. [Kasutatud 05.12.2017].
- [30] A. J. Teichtahl, „BMC Musculoskeletal Disorders,“ BioMed Central Ltd, 15 02 2012. [Võrgumaterjal]. Available: <https://bmcmusculoskeletdisord.biomedcentral.com/articles/10.1186/1471-2474-13-19>. [Kasutatud 05.12.2017].
- [31] U. Technologies, „Unity Documentation,“ Unity Technologies, 01.12.2017. [Võrgumaterjal]. Available: <https://docs.unity3d.com/ScriptReference/AvatarIKGoal.html>. [Kasutatud 22.12.2017].
- [32] OptiTrack. [Võrgumaterjal]. Available: <https://www.optitrack.com/products/slim-13e/indepth.html>. [Kasutatud 27.11.2017].

Lisa 1 - SteamVR teegi abil seadmete tuvastamine

```
bool triggerClicked = false;
bool gripClicked = false;
bool touchClicked = false;
int currentTrackedDeviceCount = 0;
Dictionary<int, SteamVR_Controller.Device> trackedDevices = new
Dictionary<int, SteamVR_Controller.Device>();
Text logText;
Text numberOfDevicesText;

for (int i = 1; i <= numDevices; i++)
{
    SteamVR_Controller.Device device = SteamVR_Controller.Input(i);
    if (device != null && device.hasTracking && device.connected &&
device.valid)
    {
        ETrackedDeviceClass deviceClass =
OpenVR.System.GetTrackedDeviceClass((uint)i);
        if (deviceClass == ETrackedDeviceClass.Controller ||
deviceClass == ETrackedDeviceClass.GenericTracker)
        {
            if (!trackedDevices.ContainsKey(i))
            {
                trackedDevices.Add(i, device);
                logText.text = string.Format("{0}\n Found device with
number {1}, type {2}", logText.text, i, deviceClass);
            }
            currentTrackedDeviceCount++;
            triggerClicked |=
device.GetPressUp(SteamVR_Controller.ButtonMask.Trigger);
            touchClicked |=
device.GetPressUp(SteamVR_Controller.ButtonMask.Touchpad);
            gripClicked |=
device.GetPressUp(SteamVR_Controller.ButtonMask.Grip);
        }
    }
    numberOfDevicesText.text = string.Format("Number of devices: {0}",
currentTrackedDeviceCount);
}
```


Lisa 2 – Kalibreerimispunktite kaugus põhikaamerast.

```
Vector3 cameraPosition = Camera.main.transform.position;
Vector3 targetForward = Camera.main.transform.rotation * Vector3.forward;

foreach (var trackedDevice in trackedDevices)
{
    var distanceFromLine = Extensions.LeftOrRightFromLine(cameraPosition,
targetForward, trackedDevice.Value.transform.pos);
    _calibrationPoints.Add(new CalibrationPoint()
    {
        Position = trackedDevice.Value.transform.pos,
        Rotation = trackedDevice.Value.transform.rot,
        DeviceClass =
OpenVR.System.GetTrackedDeviceClass((uint)trackedDevice.Key),
        DeviceIndex = trackedDevice.Key,
        DistanceFromLine = distanceFromLine
    });
}

//Extensions.cs
//Linear Equation  $x-x_1 = m(z-z_1)$ 
public static float LeftOrRightFromLine(Vector3 origin, Vector3 target,
Vector3 point)
{
    var slope = origin.x - target.x / origin.z - target.z;
    return point.x - origin.x - slope * (point.z - origin.z);
}
```

Lisa 3 – Alakeha jaoks vajalike kalibreerimispunktide määramine

```
float footMax = 0.2f;
float kneeMax = 0.35f;

for (int i = 0; i < calibrationPoints.Count; i++)
{
    var calibrationPoint = calibrationPoints[i];
    var trackerHeightPercentage = calibrationPoint.Position.y /
Camera.main.transform.position.y;

    if (trackerHeightPercentage < footMax {
        var leftFoot = calibrationPoints.FirstOrDefault(x => x.Type ==
CalibrationPointType.LeftFoot);

        if (leftFoot == null) {
            calibrationPoint.Type = CalibrationPointType.LeftFoot;
        }
        else {
            if (calibrationPoint.DistanceFromLine <
leftFoot.DistanceFromLine) {
                calibrationPoint.Type = CalibrationPointType.LeftFoot;
                leftFoot.Type = CalibrationPointType.RightFoot;
            }
        }
    }

    if (trackerHeightPercentage > footMax && trackerHeightPercentage <
kneeMax) {
        var leftKnee = calibrationPoints.FirstOrDefault(x => x.Type ==
CalibrationPointType.LeftKnee);
        if (leftKnee == null) {
            calibrationPoint.Type = CalibrationPointType.LeftKnee;
        }
        else {
            if (calibrationPoint.DistanceFromLine <
leftKnee.DistanceFromLine) {
                calibrationPoint.Type = CalibrationPointType.LeftKnee;
                leftKnee.Type = CalibrationPointType.RightKnee;
            }
        }
    }
}
```