



Tommi Juutilainen

Käsien seuranta virtuaalitodellisuu- dessa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikan tutkinto-ohjelma

Insinöörityö

25.5.2023

Tiivistelmä

Tekijä: Tommi Juutilainen
Otsikko: Käsien seuranta virtuaalitodellisuudessa
Sivumäärä: 33 sivua
Aika: 25.5.2023

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Tieto- ja viestintätekniikka
Ammatillinen pääaine: Pelisovellukset
Ohjaaja: Lehtori Miikka Mäki-Uuro

Insinööriyössä perehdyttiin virtuaalitodellisuuden nykytilanteeseen. Tarkoituksena oli selvittää, kuinka helposti pelimoottorilla pääsee sovelluskehityksessä alkuun, kun päätavoitteena on virtuaalilasien käsienseurantaominaisuuksien käyttö.

Käsienseurannassa edellytyksenä ovat sisältä ulospäin tapahtuvan seurannan mahdollistavat kamerat, riittävä prosessointiteho ja toimiva ohjelmisto. Ohjaimet tarvitsevat IMU-sensorin (inertial measurement unit eli inertiaalinen mittaussyksikkö) lisäksi IR-ledit (infrared eli infrapuna) tai oman sisältä ulospäin seurannan toteutuksen. Käsien ja ohjainten seurantatehokkuuteen vaikuttavat valoisuus, käsien etäisyys sekä toisistaan että virtuaalilaseista tai tukiasemista, käsien nopeus ja laitteiston ja ohjelmiston tehokkuus. Työssä käytettiin mallilaitteena virtuaalilaseja, joissa on käytössä sekä ohjaimet että käsienseurannan mahdollistava sisältä ulospäin seuranta.

Työssä käytettiin Metan ja Unityn tarjoamia kehityspaketteja, ja niiden avulla luotiin pelillisiä ominaisuuksia, kuten tarttumista, heittämistä ja käsieleitä. Tuloksena oli pelinäkö, jossa toimii sekä käsienseuranta että ohjainten seuranta: molemmilla pake-teilla voi tarttua esineeseen, myös etäisyydeltä, ja laatikkoon, jonka kokoa voi käsi-tellä. Objekteilla on myös fysiikat, joten niitä voi heittää. Metan paketilla, kun kädet ovat käsienseurantatilassa tietyssä asennossa, tulee käsistä salamoita.

Työssä vertailtiin kahta ohjelmistokehityspakettia: Oculus Integration SDK- ja Unity XR Interaction Toolkit -pakettia. Oculus Integration SDK -paketti, joka on tarkoitettu Metan laitteille, on ominaisuuksiltaan laaja, ja kehittäjää varten on tehty paljon val-miiksi, joten koodaustaitoja ei vaadita. Valmiita objekteja viedään pelinäkömään ja toistensa komponenttien referensseiksi. Unity XR Interaction Toolkit -paketti puoles-taan tukee useampaakin laitevalmistajaa, mutta esimerkiksi käsienseurannassa sillä menettää usein otteen objektista, vaikka käsi on vielä tartunta-asennossa. Käsienseurantaominaisuus on paketissa melko tuore, joten se todennäköisesti vielä para-nee.

Avainsanat: virtuaalitodellisuus, laajennettu todellisuus, käsienseu-ranta, ohjainten seuranta, Unity-pelimoottori

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author: Tommi Juutilainen
Title: Hand tracking in virtual reality
Number of Pages: 33 pages
Date: 25 May 2023

Degree: Bachelor of Engineering
Degree Programme: Information and Communications Technology
Professional Major: Game Applications
Supervisor: Miikka Mäki-Uuro, Senior Lecturer

The purpose of the project was to investigate the current trends in virtual reality and popular devices in the field, and thereafter to focus on hand and controller tracking. The circumstances that affect tracking quality were also analyzed. Lastly, the idea of how easy it is to set up a virtual reality project with hand tracking and to implement basic game features, such as grabbing, throwing, and hand signs, was investigated.

During the project, a virtual reality headset with inside-out tracking and hand-tracking ability was used. Inside-out tracking means that the headset can track its position and orientation without needing to mount additional trackers, called base station light-houses, for example to a wall. Hand-tracking requires inside-out tracking and appropriate cameras, hardware, and software to work.

The Unity packages used and compared were Oculus Integration SDK and Unity XR Interaction Toolkit. The result, with each package, was a game scene in which both controller and hand tracking work, and objects that have physics can be grabbed, even from a distance, and thrown. Also, with the Oculus Integration SDK, lightning comes out of the hands once they are in a correct pose. The Unity XR Interaction Toolkit only recently got hand-tracking support, and it seems to fail to keep the grab interaction on from time to time. The feature will most likely improve when it gets updated in the future. Neither of the packages needed coding to use and especially with the Oculus Integration SDK one could continue to develop a game with these now-implemented basic features.

Keywords: virtual reality, extended reality, hand-tracking, inside-out tracking, Unity game engine

Sisällys

Lyhenteet

1	Johdanto	1
2	Merkittävimmät XR-yritykset	2
2.1	Meta Quest 2 -virtuaalilasit	3
2.2	Valve Index -virtuaalilasit ja -ohjaimet	4
3	Ohjainten seuranta	5
3.1	IMU-sensori	6
3.2	Optinen seuranta	8
4	Käsienseuranta	13
4.1	Ultraleapin käsienseuranta	14
4.2	Metan käsienseuranta	15
5	Seurantaa haittaavat tekijät	16
5.1	Käsienseuranta	16
5.2	Ohjaimet	17
6	Käsienseurannan toteutuksia	17
6.1	Oculus Integration -ohjelmistokehityspaketti	20
6.1.1	Käyttöönotto	21
6.1.2	Tarttuminen	22
6.1.3	Käsieleiden hyödyntäminen	26
6.2	Unity XR Interaction Toolkit -paketti	30
6.2.1	Käyttöönotto	30
6.2.2	Tarttuminen	31
7	Yhteenveto	32
	Lähteet	34

Lyhenteet

- AR: *Augmented reality*. Lisätystä todellisuudesta puhutaan, kun nykyiseen todellisuuteen lisätään asioita, jotka eivät fyysisesti siellä sijaitse. Pokémon Go esimerkiksi mielletään AR-peliksi, sillä siinä seikkaillaan oikeassa maailmassa ja puhelimen kameran kuvaamaan taustaan lisätään muun muassa Pokémoneja [1].
- IMU: *Inertial measurement unit*. Inertiaalinen mittausyksikkö on elektroninen laite, joka mittaa ominaisvoimaa, kulmanopeutta ja suuntaa kiihtyvyyssanturin, gyroskoopin ja magnetometrin avulla [2].
- IR: *Infra red*. Infrapunäsäteily on elektromagneettista säteilyä, joka on aallonpituudeltaan suurempaa kuin näkyvä valo ja täten näkymätöntä ihmissilmälle [3]. Ihminen näkee 400–700 nanometrin väliset aallot, kun taas infrapuna säteilee 0,7–300 mikrometrin (μm) välillä [3].
- MR: *Mixed reality*. Tehostettu todellisuus yhdistää lisätyn todellisuuden (AR) ja virtuaalitodellisuuden (VR) ominaisuudet ja luo elämyksen, jossa sekä virtuaalisen maailman että fyysisen maailman asiat yhdistyvät vielä mukaansatempaavammin kuin pelkästään lisätyssä todellisuudessa [1].
- VR: *Virtual reality*. Virtuaalitodellisuus on nykyisestä todellisuudesta kokonaan eroava todellisuus, joka luodaan esimerkiksi virtuaalilasien kanssa [1].
- XR: *Extended reality*. Laajennettu todellisuus on käsite, johon sisällytetään lisätty todellisuus (AR), virtuaalitodellisuus (VR) ja tehostettu todellisuus (MR) [1].

1 Johdanto

Insinööriyön tarkoituksena on perehtyä virtuaalitodellisuuden nykytilaan ja tutustua käsien ja ohjainten seurantaan. Työssä pohditaan myös, mitkä seikat vaikeuttavat molempien seurantaan. Keskeisenä tavoitteena on myös selvittää, kuinka helppoa on aloittaa sovelluskehitys virtuaalilaseille Unity-pelimootorilla. Unity-pelimootorilla käytetään kahta valmispakettia: Oculus Integration SDK ja Unity XR Interaction Toolkit. Oculus Integration SDK -paketti on tarkoitettu vain Metan laitteille, mutta Unity XR Interaction Toolkit -paketti tukee useampaa laitevalmistajaa. Tarkoituksena on saada käsienseuranta toimimaan molemmilla paketeilla ja toteuttaa pelillisiä ominaisuuksia, kuten tarttumista, heittämistä ja käsieleitä.

Virtuaalilasien näyttö on yleensä jaettu kahdelle silmälle niin, että kullekin silmälle näytetään eri näkymää stereoskooppisen 3D-elämyksen luomiseksi. Edistyneellä laitteistolla ja tarkkaan renderöidyllä grafiikalla luodaan tietokonesimuloitu 3D-maailma, jossa käyttäjä voi esimerkiksi liikkua ja olla vuorovaikutuksessa siinä esiintyvien asioiden kanssa. Virtuaalisen maailman todellisuuden illuusiota pyritään tehostamaan tarjoamalla aistiärsyksiä mahdollisimman realistisesti. Näköaistin lisäksi olennainen aisti on ääni, jota pyritään virtuaalimaailmassa tuottamaan niin, että ääni kuuluu käyttäjään nähden todenmukaisesta suunnasta. [4.]

Ohjaimet on suunniteltu niin, että niiden napit ja värinä auttavat tuntemisen simuloimisessa: useimmat ohjaimet tunnistavat painallusten lisäksi myös kosketuksen. On myös kehitetty puettavia tuntoaistiin perustuvia (engl. haptic) laitteita, jotka aiheuttavat sähköistä lihasstimulaatiota simuloidakseen tuntemista virtuaalimaailmassa [5]. Esimerkiksi taistelupelissä pelaaja kykenee tuntemaan iskut ja luodit. Hajua simuloimaan on myös olemassa laite nimeltään ION, joka simuloi VR-maailmassa esiintyviä hajuja yhdeksällä erilaisella kemiallisella yhdisteellä [6]. Koko kehon liikettä voidaan myös seurata esimerkiksi nilkkoihin

kiinnitettävillä seuraajilla (engl. tracker), joista uusimpana innovaationa HTC on julkaissut omilla kameroilla ja prosessoreilla varustetut seuraajat [7].

Virtuaali- ja lisätyn todellisuuden laitteita hyödynnetään pelien lisäksi muun muassa koulutuksessa, tutkimuksissa ja työympäristöissä. Oppiminen tapahtuu huomattavasti tehokkaammin virtuaaliympäristön luoman kokemuksen myötä kuin esimerkiksi vain videoita katselemalla, sillä hyvin luotu virtuaalikokemus on niin mukaansatempaava, että aivot aktivoituvat miltei samalla tavalla kuin saman kokemuksen kokisi oikeassa elämässä. Esimerkiksi tutkimuksessa, jossa kaksi ryhmää pyrkivät muistamaan taijiliikesarjoja, ryhmä, joka näki ne stereoskooppisesti 3D:nä suoriutui niistä 25 % paremmin kuin tavallisia videoita katsonut ryhmä. Virtuaalimaailman kokemukset ovat usein turvallisempia ja kustannustehokkaampia kuin oikeassa maailmassa. [8.] Esimerkiksi lentokoulutuksessa ei tarvitse huolehtia lentokoneen tai lentäjän ehjänä pysymisestä.

Laajennettu todellisuus eli XR (engl. extended reality) on yleisesti käytetty katto-termi, joka sisältää virtuaalitodellisuuden lisäksi lisätyn todellisuuden (AR) ja tehostetun todellisuuden (MR). Virtuaalimaailmassa olevien asioiden kanssa vuorovaikutus tapahtuu joko käsissä pidettävien ohjaimien avulla tai suoraan käsillä, mikäli laseissa on sen mahdollistavat laitteistot ja ohjelmistot. Ohjainten kanssa tapahtuva seuranta puolestaan tapahtuu magnetometrin, kiihtyvyyssmittarin, gyroskoopin ja optisen seurannan, kuten IR-ledien, avulla [9].

2 Merkittävimmät XR-yritykset

Tunnetuimpia XR-alan yrityksiä ovat Meta, Valve, HTC, HP ja Sony [10]. CNET:n lisäksi pelilehdet PCGAMER, PCMAG ja IGN ovat kaikki maininneet Meta Quest 2:n ja Valve Indexin viiden parhaan joukkoon. Sonyn laitteet, muista valmistajista poiketen, on kuitenkin tarkoitettu vain sen omien konsolien, PlayStationien, kanssa käytettäväksi, mikäli ei käytä kolmannen osapuolen tarjoamia keinoja yhdistääkseen niitä tietokoneeseen [11]. Esimerkiksi Sonyn uusimpien PSVR2-lasien käyttöön vaaditaan myös Sonyn uusin konsoli, PlayStation 5. Meta on viime vuosina dominoinut VR-markkinoita: sen vuoden 2021

markkinaosuus oli 80 % [12]. Suomesta löytyy kansainvälisesti noteerattu, mutta jo hintojensakin puolesta lähinnä yrityksille VR- ja XR-laitteita myyvä Varjo [13]. Yritys on erityisen tunnettu silmänresoluutioisista näytöistä, joita suositaan muun muassa simulaattorikäytössä [13].

2.1 Meta Quest 2 -virtuaalilasit

Metan suosituin virtuaaliodellisuustuote on vuonna 2020 julkaistut Meta Quest 2 -virtuaalilasit, jotka vielä ennen Facebookin uudelleenbrändäystä Metaksi, vuoden 2021 lopulla, tunnettiin Oculus Quest 2 -nimellä [14]. Näillä VR-laseilla virtuaaliympäristön kanssa vuorovaikuttaminen tapahtuu joko ohjaimilla tai pelkillä käsillä, sillä laseissa on käsiensurannan mahdollistavat kamerat, laitteisto ja ohjelmisto. Useimmat VR-lasit vaativat erillisen tukiaseman, tai ideaalisesti kaksi, lasien ja ohjaimien liikkeiden optimaaliseen seuraamiseen, mutta Meta Quest 2 -lasit hyödyntävät sisältä ulospäin seurantaa (engl. inside-out tracking), jolloin erillisiä tukiasemia ei tarvita. Laseissa on myös mobiililaitetta vastaava suoritin, ja niille luodaan optimoituja versioita peleistä ja sovelluksista, jolloin tietokoneeseen yhdistäminen kaapelilla ei ole välttämätöntä. [15.] Metan lasit ovat hyvin saavutettavissa, koska ne eivät vaadi tehokasta ja kallista tietokonetta toimiakseen, mutta voidaan myös sellaiseen kytkeä, mikäli parempi grafiikka on mieleen.

Lasit toimivat 6DOF-periaatteella (DoF on engl. degrees of freedom), mikä viittaa kehoon (engl. rigid body) 3D-tilassa. Konsepti ottaa huomioon käyttäjän pään sijainnin, pään liikkeen ja orientaation. Sijainnin muutoksen (engl. translation) XYZ-akseleilla ja rotaation muutoksen XYZ-akseleilla seuraaminen mahdollistaa objektin tarkkailun 360-asteella sekä alhaalta ylöspäin että ylhäältä alaspäin katsottuna. [16.]

Laseissa, jotka näkyvät kuvassa 1, on etupaneelin kulmissa neljä kameraa, jotka toimivat niin ohjaimien kuin käsien seurannassa. Ohjaimissa itsessään on myös kosketuksen tunnistavat etusormen liipaisin ja peukalolla käytettävät sauva ja kolme näppäintä. Niiden sivussa on erikseen vielä tarttumiselle oma

liipaisin, jota käytetään jäljelle jääneillä sormilla, mutta se ei tunnista kosketusta erikseen. [15.]



Kuva 1. Oculus Quest 2 -virtuaalilasit ohjaimineen [17].

Lasien kameroissa on myös "Passthrough"-ominaisuus, ja kun se on kytkettynä päälle, käyttäjä näkee ympäristönsä kameroiden syöttäessä kuvaa laitteen näyttöihin. Ominaisuutta hyödynnetään muun muassa aina, kun lasit kytketään päälle. Laite kysyy käyttäjältä, minkälaisen pelialueen tämä haluaa, ja käyttäjä voi piirtää ohjaimellaan alueen maahan samalla kun näkee huoneen kameroiden kautta, jotta voi välttää seiniin ja muihin esteisiin osumisen [15].

2.2 Valve Index -virtuaalilasit ja -ohjaimet

Vuonna 2019 julkaistut Valve Index -virtuaalilasit ja -ohjaimet, jotka näkyvät kuvassa 2, ovat noin tuhannen euron hintaiset (kun mukaan lasketaan lasien toimiaukseen tarvitsemat ohjaimet ja tukiasemat) ja hyvin suosittu laite. Indexin ohjaimia on kehitetty markkinoiden parhaiksi, niin ergonomian kuin kyvyn seurata jokaista sormea erikseen ansiosta. Siksi näitä ohjaimia suositaan myös Indexin kanssa kilpailevien lasien kanssa käytettäväksi. [10.]



Kuva 2. Valve Index -lasit ja -ohjaimet [18].

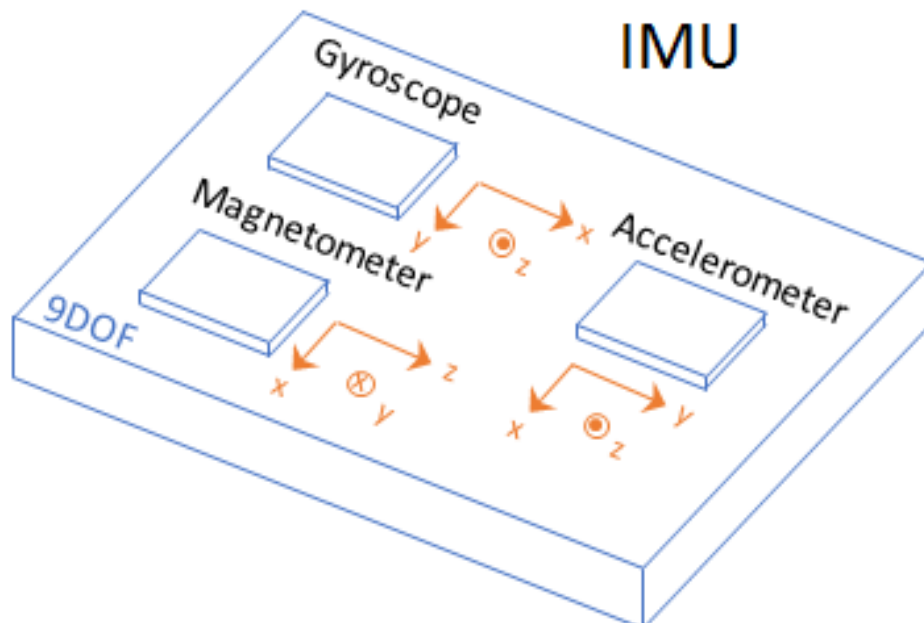
Meta Quest 2:een verrattuna Valve Indexin ohjaimet tuovat vapautta, sillä ohjaimista voi päästää irti niiden käsihihnojen ansiosta, joiden läpi kädet laitetaan [18]. Metan ohjaimien tartuntaliipaisimen sijaan Indexin ohjaimissa on laatta, joka tunnistaa erikseen keskisormen, nimettömän ja pikkurillin kosketuksen, mikä mahdollistaa esimerkiksi monipuolisemman käsimerkkien teon virtuaali-maailmassa [10].

3 Ohjainten seuranta

Useimmissa ohjaimissa hyödynnetään IMU-sensoreita (IMU on engl. Inertial Measurement Unit eli inertiaalinen mittausyksikkö), joiden gyroskoopin, kiihtyvyyssmittarin ja joissakin tapauksissa myös magnetometrillä avulla voidaan seurata ohjaimia myös niiden ollessa lasien näkökentän ulkopuolella, jolloin optiset paikannuskeinot eivät toimi. Optinen seuranta tapahtuu ohjaimissa sijaitsevien IR-ledien (IR on engl. infrared eli infrapuna) ja lasien kameroiden avulla. Mikäli lasissa ei ole sisältä ulospäin tapahtuvan seurannan toteutusta, lasissa itsessään on IR-ledit, jotka puolestaan tukiasema havaitsee.

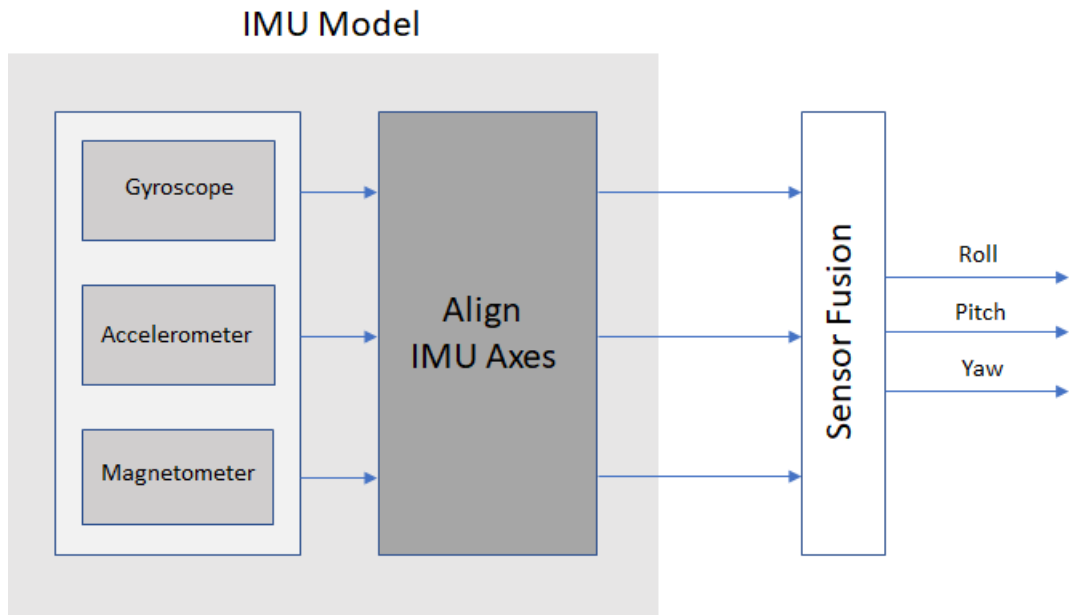
3.1 IMU-sensori

IMU-sensori on siru laitteen, kuten älypuhelimien tai tässä tapauksessa VR-laitteen ohjaimen, piirilevyssä. Se tarjoaa dataa kyseisen laitteen liikkeestä ajan funktiona. IMU-sensoria, joka sisältää kiihtyvyyssanturin ja gyroskoopin lisäksi myös magnetometrin, kutsutaan yhdistettynä 9-akseliseksi IMU:ksi, sillä jokainen antureista seuraa tiettyä tapahtumaa x-, y- ja z-akselilla. Kiihtyvyyssanturi seuraa kiihtyvyyttä kertoen mihin suuntaan laite kulkee ja gyroskooppi ja magnetometri puolestaan missä kulmassa laite on, gyroskooppi kulmanopeudella ja magnetometri kompassin lailla orientaation maan vetovoiman suhteen. [3.] Kukin komponentti on havainnollistettu kuvassa 3.



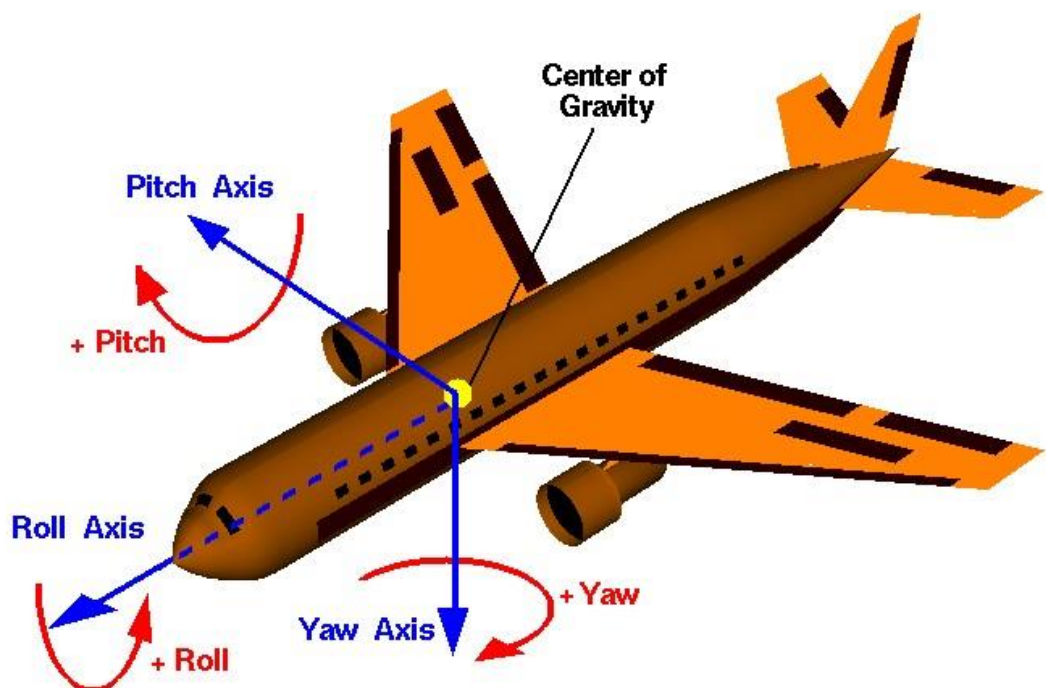
Kuva 3. IMU-sensori havainnollistettu piirilevynä [19].

Antureista saatu data yhdistetään kohdistuen akselit, jolloin lopputulokseksi saadaan lentodynamiikasta tutut kallistuminen (engl. roll), nyökkääminen (engl. pitch) ja kääntyminen (engl. yaw) kuvan 4 lailla [19].



Kuva 4. IMU:n sensorien datamuunnos [19].

Kallistuminen (roll kuvassa 5) on kääntymistä pituusakselin suhteen, kääntyminen (yaw kuvassa 5) kääntymistä pystyakselin suhteen ja nyökkääminen (pitch kuvassa 5) kääntymistä poikkiakselin suhteen [20].

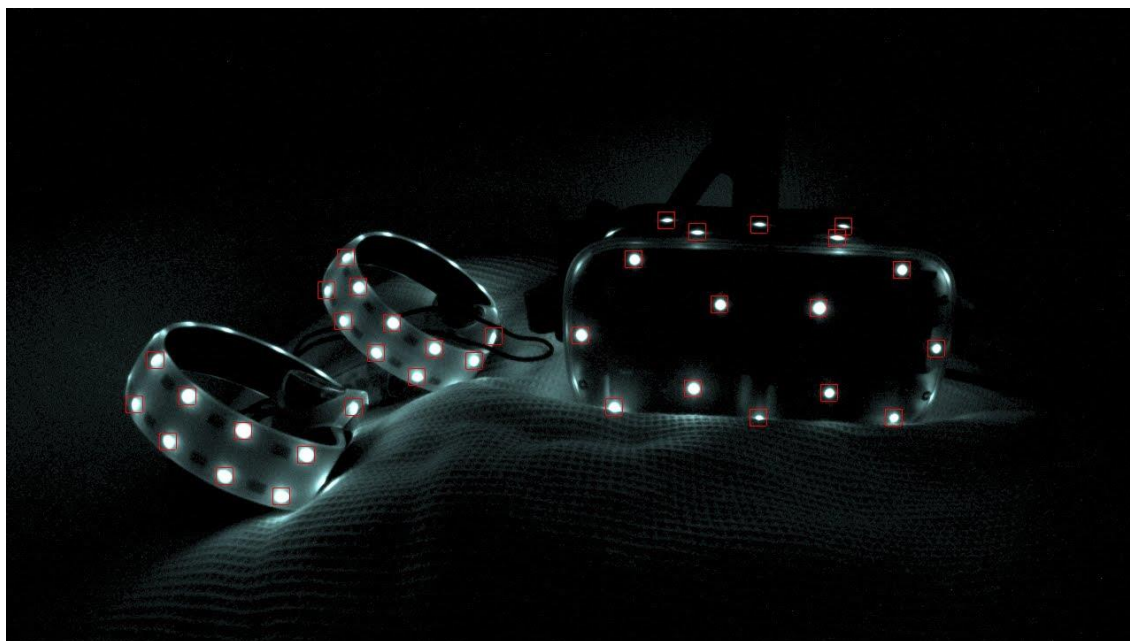


Kuva 5. IMU-rotatiot esitettynä lentokonemallin avulla [20].

Jotta IMU-sensorin tulokset olisivat mahdollisimman realistisia, tarvitaan Ground Truth -dataa, johon tuloksia verrata. Ground Truth on generoitu tietoaaineisto, joka on ideaalitulokset, johon algoritmin tulisi pyrkiä. Datan generoimiseen voidaan hyödyntää automaatiota, esimerkiksi robottikättä, tai se voidaan myös luoda synteettisesti. Näin minimoidaan ihmisvirhe, sillä asiaa, joka IMU:n kohdalla olisi joukko liikkeitä, tulisi toistaa mahdollisimman tarkasti. [21.]

3.2 Optinen seuranta

IR-valojen vilkkuessa tietyllä taajuudella lasien tai tukiaseman kamera erottaa ohjaimien IR-valot mahdollisista muista ympäristössä olevista lähteistä [9]. Kuvasta 6 näkee Oculus Quest -VR-lasien lediasetelman ja ledien muodostamat valoläiskät (engl. blob).



Kuva 6. IR-lediasetelma Meta Quest 2 -VR-laitteistossa [22].

Seurannan kannalta huomioonotettavia asioita ovat esimerkiksi läiskien kirkkaus ja koko. Läiskä tarkoittaa kuvassa ryhmää pikseleitä, joilla on jokin sama piirre, kuten sama harmaasävyyluku (engl. greyscale value). Läiskien tunnistukseen on olemassa algoritmeja ja Pythonilla on OpenCV-kirjastoa hyödyntävä SimpleBlobDetector-algoritmi, esimerkikoodi 1, joka muuntaa alkuperäisen

kuvan binäärikuviksi raja-arvojen avulla. Tämän jälkeen jokaisessa binäärikuvassa yhdistetyt valkoiset pikselit ryhmitellään niin sanotuiksi binääriläiskiksi. Binääriläiskien keskipisteet lasketaan, ja läiskät, jotka ovat lähempänä kuin asetettu minimietäisyys toisiinsa nähden, yhdistetään. Viimeiseksi yhdistetyistä läiskistä lasketaan ja palautetaan niiden keskipiste ja säde. [23.]

```

# Standard imports
import cv2
import numpy as np

# Read image
im = cv2.imread("blob.jpg")

# Setup SimpleBlobDetector parameters.
params = cv2.SimpleBlobDetector_Params()

params.blobColor = 255 # white

# Change thresholds
params.minThreshold = 10
params.maxThreshold = 200

# Filter by Area.
params.filterByArea = True
params.minArea = 10

# Filter by Circularity
params.filterByCircularity = True
params.minCircularity = 0.1

# Filter by Convexity
params.filterByConvexity = True
params.minConvexity = 0.87

# Filter by Inertia
params.filterByInertia = True
params.minInertiaRatio = 0.01

# Create a detector with the parameters
detector = cv2.SimpleBlobDetector_create(params)

# Detect blobs.
keypoints = detector.detect(im)

# Draw detected blobs as red circles.
# cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS ensures
# the size of the circle corresponds to the size of blob

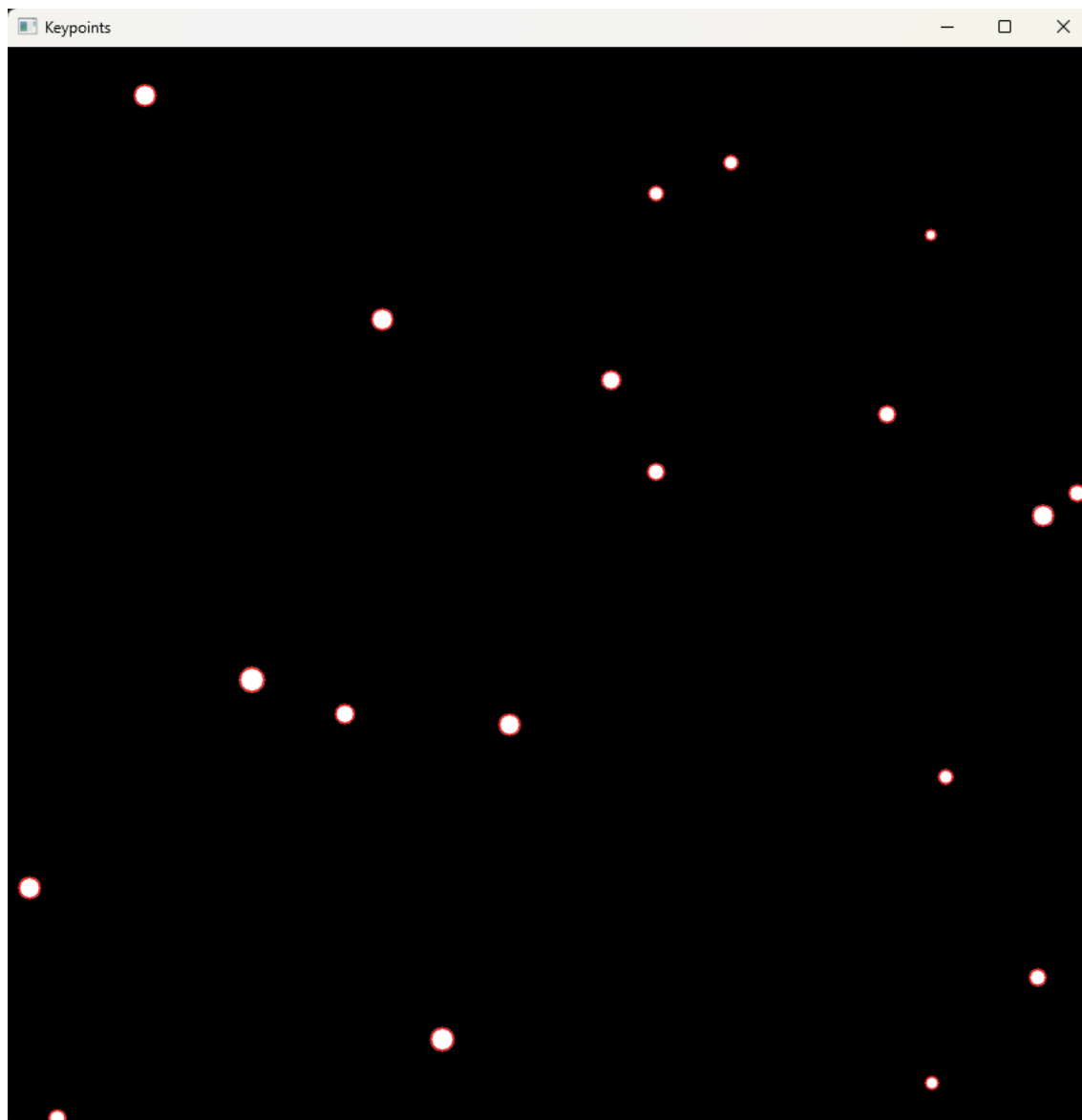
im_with_keypoints = cv2.drawKeypoints(
    im, keypoints, np.array([]), (0, 0, 255),
    cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS
)

# Show blobs
cv2.imshow("Keypoints", im_with_keypoints)
cv2.waitKey(0)

```

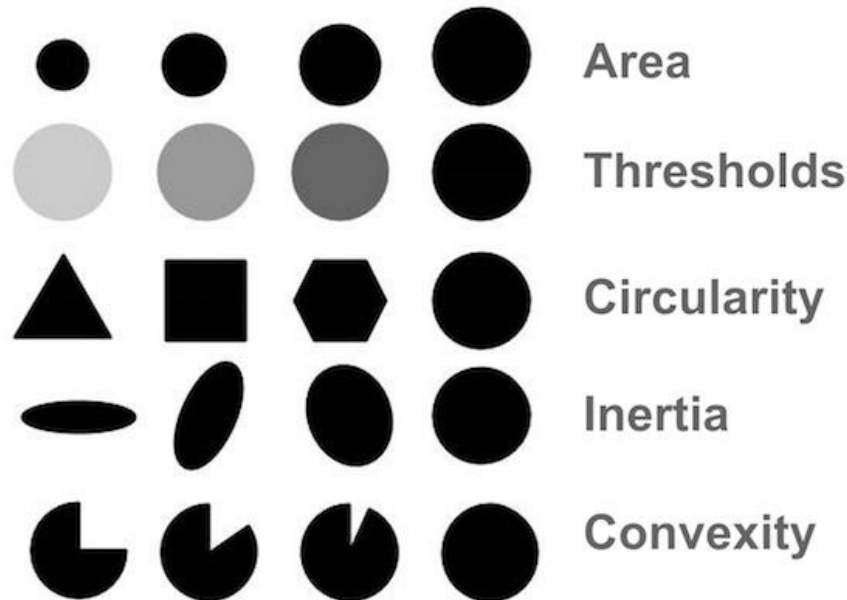
Esimerkkikoodi 1. Pythonin OpenCV-kirjastoa hyödyntävä koodi lukee samassa kansiossa olevan "blob.jpg"-kuvan ja piirtää punaiset ympyrät parametrien lo-massa havaittujen läiskien ympärille [23].

Koodi on lainattu suoraan lähteestä, mutta blobColor-parametrille lisättiin arvo 255, sillä kuvassa 7 etsityt läiskät ovat valkoisia, kuten VR-laitteissa olevat läiskät kuvassa 6. Myös minArea-parametria on alennettu kymmeneen pikseliin.



Kuva 7. Esimerkkikoodilla 1 havaitut valkoiset ympyrät Howard Bushousen ResearchGate-verkkosivustolle lataamassa "IR channel blob mask" -kuvassa [24].

Esimerkkikoodissa 1 läiskiä voidaan suodattaa kuvan 8 tavoin alueella (engl. area), väriarvolla (engl. thresholds), pyöreydellä (engl. circularity), kuperuudella (engl. convexity) ja inertiasuhteella (engl. inertia rate) [23].



Kuva 8. SimpleBlobDetector-algoritmin suodatusvaihtoehdot [23].

Väriarvon kannalta, mitä lähempänä parametri blobColor on nollaa, sitä tummempaa läiskää haetaan, ja mitä lähempänä 255, sitä valkoisempaa. Koon puolesta parametri minArea suodattaa pois asetetun arvon alhaisemmat pikselimääräiset läiskät. [23.] Pyöreys lasketaan kaavan 1 mukaisesti.

$$\frac{4\pi * Pinta-ala}{(kehä)^2} \quad (1)$$

Ympyrä on pyöreydeltään 1, kun esimerkiksi neliön pyöreys on arvoltaan 0,785. Kuperuus määritetään läiskän pinta-ala jaettuna läiskän kupera osa ja kuperuuden maksimiarvo on 1. Inertia tarkoittaa tässä tapauksessa sitä, kuinka venynyt (engl. elongated) läiskä on [23].

Metan uusimpien VR-lasien Meta Quest Pron kanssa lokakuussa 2022 julkaistuissa ohjaimissa IR-valoläiskien lisäksi ohjaimet on varustettu kolmella kameralla ja niiden sisällä olevilla prosessoreilla. Tämä mahdollistaa ohjaimien oman

sisältä ulospäin seurannan, minkä takia ohjainten sijainti päivittyy tarkasti myös silloin, kun ohjaimet ovat poissa VR-lasien kameroiden näkökentästä. Quest Pro ohjaimia myydään myös erikseen ja ne toimivat Meta Quest Pro:n lisäksi Meta Quest 2 -lasien kanssa. [25.]

Ohjainten kamerat, kuten sisältä ulospäin seurantaan kykenevät lasitkin, hyödyntävät SLAM-teknologiaa (engl. lyhenne simultaneous localization and mapping) eli samanaikaisesti tapahtuvaa lokalisaatiota ja kartoittamista. Ympäristön kartoittaminen ja laitteen sijainnin sijoittaminen karttaan samanhetkisesti auttaa esimerkiksi esteiden havaitsemisessa. [26.]

4 Käsienseuranta

Käsienseurannan (engl. hand tracking) avulla käyttäjä voi olla vuorovaikutuksessa ympäristönsä kanssa virtuaalimaailmassa ilman ohjaimia. Tähän vaaditaan laseilta sisältä ulospäin seuranta. Käsienseurantaohjelmisto käsittelee kameran keräämän datan käsien sijainnista, orientaatiosta ja nopeudesta ja luo käsistä reaaliaikaiset virtuaaliset versiot, jotka voidaan renderöidä tuettuihin ohjelmistoihin. Virtuaalisten käsien kuuluisi päivittyä virtuaalimaailmassa mahdollisimman tarkasti oikeisiin käsiin nähden, jottei elämykseen uppoutuminen eli immersio rikkoutuisi. [27.] Ohjainten näppäinten tarjoama vuorovaikutus tai ohjaimilla asioihin osoittelu korvataan käsienseurannassa käsielein. Esimerkiksi käsienseurantayrityksen Ultraleapin luomalla käsienseurannalla etusormella voidaan osoittaa asioihin ja valikoissa osoitetun asian voi valita, kun laittaa etusormen ja peukalon yhteen nipistysliikkeen lailla. Valikon saa avattua Ultraleapilla, kun kääntää kädestä kämmenpuolen näkyviin. [28.]

Laitteistopuolella edellytyksinä toimivalle käsienseurannalle Ultraleap listaa tarakan syvyysnäön, laajan näkökentän, VR-lasien näyttöä vastaavan ruudunpäivitysnopeuden ja lisävalaistuksen luomisen. Näkökentän tulisi olla ideaalisesti laajempi tai syvempi kuin laseissa itsessään, jotta käsiä seurataan jo, kun ne tulevat lasien näyttämään näkökenttään. Nopea ruudunpäivitysnopeus puolestaan vähentää viivettä, mikä haittaisi immersiota. Lasien olisi lisäksi hyvä

tuottaa lisävalaistusta käsiin, sillä näkyvän valon tuottama valaistus ei välttämättä ole riittävän kirkas tuottamaan tarvittavan tarkkaa dataa. [27.]

4.1 Ultraleapin käsienseuranta

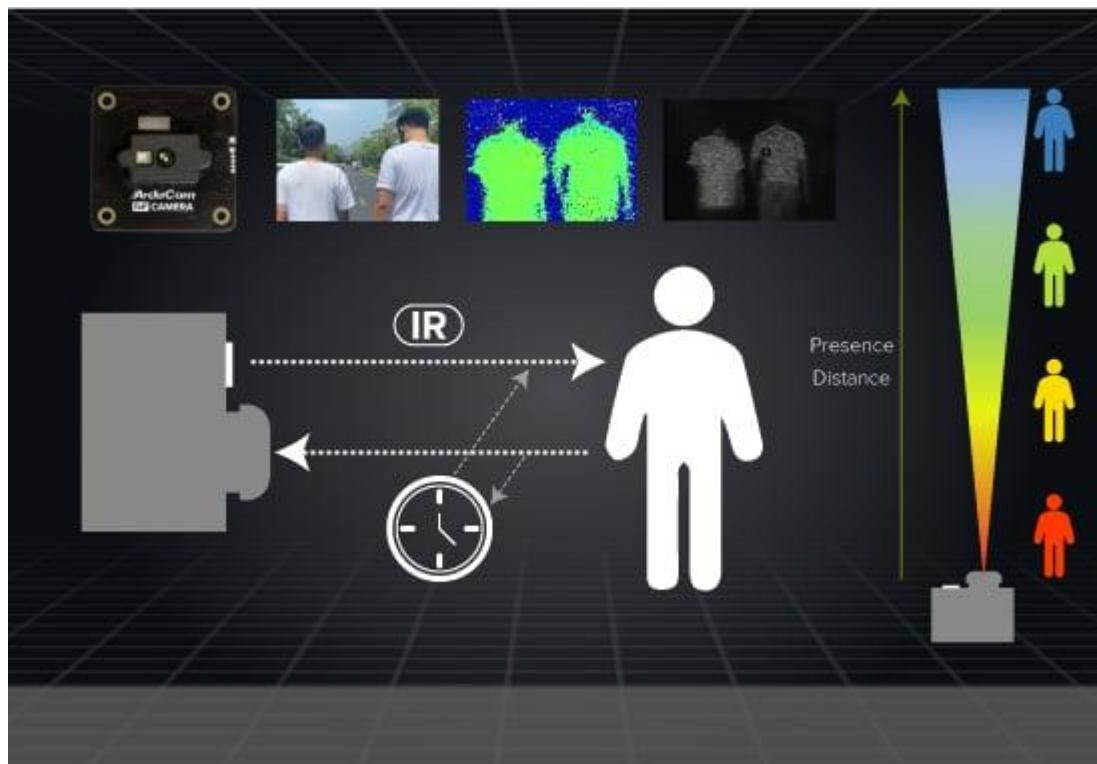
Laitteistopuolella Ultraleap suorittaa käsien valaistuksen IR-LEDien ja kahdella kameralla, jotka on synkronoitu LEDien kanssa, jotka vilkkuvat 120 kertaa sekunnissa [27]. Infrapunavalon ollessa aallonpituudeltaan suurempaa kuin näkyvä valo, ihmissilmä ei sitä havaitse, mutta kamerat havaitsevat [2]. Ultraleapin kameroiden ottama kuvanlaatu vastaa kuvaa 9.



Kuva 9. Näkymä Ultraleapin moduulin kahdesta kamerasta [27].

Ohjelmistopuolella, kun laitteiston keräämää dataa on karsittu, vuoron saa Ultraleapin oma seurantamoottori. Moottori hyödyntää luuston seurantamallia, joka luo kyynärvarsista, ranteista ja käsistä rakenteet mallintaen niiden nivelet ja luut. Kun seurantamoottori toimii, itse VR-sovelluksen on vielä tunnistettava, mitä käyttäjä yrittää käsillään missäkin tilanteessa tehdä. Täytyy esimerkiksi voida erottaa aie tarttua jostakin siitä, että käsi on nyrkissä ja aikoo lyödä jotakin. Pelimoottoreita esimerkiksi ei ole suunniteltu ihmiskäsille, joten tarttumisen sijasta tuleekin helposti tönäistyä kappaletta sen sijaan, että pelimoottori rekisteröisi tapahtuman tarttumisaikeena. [27.]

Muun muassa suomalainen XR-yritys Varjo käyttää Ultraleapin käsienseurantaa laitteissaan [28]. Varjon XR-3-lasit hyödyntävät käsienseurannassa muun muassa syvyyskameraa ja ToF-kameraa (engl. time-of-flight) [29]. ToF-teknologia toimii siten, että kamera erittää valoa ja mittaa, kuinka kauan menee, että esimerkiksi käsistä kimpoava valo palaa takaisin kameraan, kuten kuvassa 10 selitetään [30].



Kuva 10. Arduinin ToF-moduulin toiminta [30].

4.2 Metan käsienseuranta

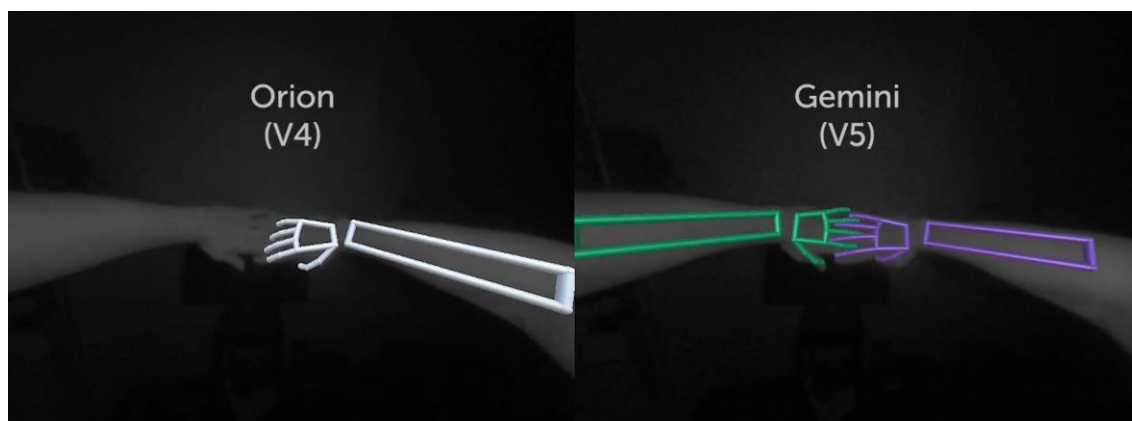
Meta tarjoaa käsienseurannastaan melko niukasti tietoa Ultraleapiin verrattuna. Seuranta tapahtuu reaaliaikaisesti kameroiden ottamien kuvien avulla arvioimalla ohjelmistopuolella pisteet eri puolille käsiä, kuten rystysiin ja sormenpäihin. Metan Quest 2:een tallennetaan äskettäin generoitu data käsien asennosta ja arvio käsien koosta. Arvioiden myötä käden tilalle asetetaan geneerinen käsimalli virtuaalimaailmassa ja jo prosessoidut kuvat Metan mukaan poistetaan, eikä niitä säilytetä Metan palvelimilla. [31.]

5 Seurantaa haittaavat tekijät

5.1 Käsienseuranta

Mikäli valaistus on liian kirkas tai liian himmeä, käsien havaitseminen vaikeutuu. Metan lasien käyttäjiä huomautetaan valaistuksen puutteesta varsinkin iltaisin. Hehkulamput, halogeenilamput ja luonnonvalo valaisevat myös ympäristöä infrapunalla, joten IR-ledien ja kameroiden on operoitava ideaalisella aallonpituudella [28]. Ultraleapin kameramoduulin IR-suodatin, pieni neliö kameroiden päällä, suodattaa pois kaikki aallonpituudet, jotka eivät ole 825–875 nanometrin sisällä. Ohjelmistopuolella, mikäli se havaitsee, että huoneen valaistus on hyvin himmeä, se kääntää ledejä himmentymään. Mikäli kädet liikkuvat hitaasti, ohjelmisto saattaa hidastaa kameran päivitysnopeutta, jotta seurantatarkkuus paranee. [32.] Mikäli kädet liikkuvat liian nopeasti, käsiä on hankala seurata [27].

Laitteiston ja ohjelmiston tehokkuus täytyy myös ottaa huomioon. Mikäli esimerkiksi kamerat ovat liian huonoresoluutioisia tai näkökenttä (engl. FOV, field of view) ei ole tarpeeksi laaja, vaikeuttaa se seurantaa myös. Käsienseuranta vaatii ohjelmistopuolelta optimointia ja edistyneitä algoritmeja. Kameroiden täytyy myös olla kalibroitu, ja eräs ongelmatilanteista ilmenee myös, jos käyttäjän kädet ovat liian lähellä toisiaan, kiinni toisissaan, kuten kuvassa 11 ilmenee, tai lähellä käyttäjän vartaloa. [27.]



Kuva 11. Ultraleapin kaksi käsienseuranta-algoritmia [28].

5.2 Ohjaimet

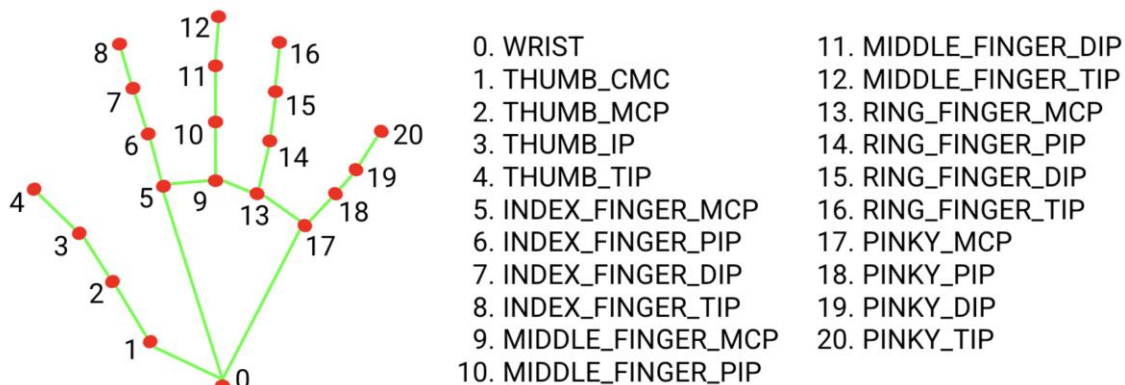
Valoisuus on iso tekijä IR-valoläiskillä seurattavien ohjaimien seurannassa, ja eri valolähteiden ja pintojen luomat heijastukset saattavat vaikeuttaa valoläiskien seurantaa. Esimerkiksi rosoisuus tai säännölliset toistuvat muodot taustalla voivat saada käsienseurannan näkemään nekin IR-ledien muodostamina valoläiskinä, jos niistä heijastuu valoa. Valoläiskien on myös oltava sijoitettu niin, että käyttäjän sormet eivät käytössä mene ledien päälle, ja materiaali valittava niin, että valoläiskät muodostuvat tasaisesti ottaen huomioon valon taittuminen päällysmateriaalista. [33.]

Ohjaimien täytyy olla VR-lasien näkökentässä, jotta IR-valoläiskien seuranta onnistuu ja myös ohjainten pitkä etäisyys laseista vaikeuttaa seurantaa. Ohjaimien lähekkäisyys toisiinsa nähden saattaa myös luoda seurantajärjestelmälle haasteita. Laitteiston ja ohjelmiston optimointi ja ohjaimien hyvä kalibrointi ovat myös ratkaisevia tekijöitä ohjainseurannassa. [33.] Riittävä virransaanti on myös olennaista.

6 Käsienseurannan toteutuksia

Ennen kuin työssä tutustuttiin Unityn puolella oleviin paketteihin, etsittiin yksinkertainen koodiesimerkki, joka havainnollistaa, miltä pisteiden piirto käsiin näyttää. Python-kirjastot MediaPipe ja OpenCV tulivat ensimmäisessä Google-tuloksessa vastaan. Python-koodin suorittamista varten tarvitaan koodieditori, kuten Visual Studio Code, ja aiemmin mainittujen Python-kirjastojen asennus ja tuonti [34].

MediaPipe voidaan jakaa kämmenen tunnistusmalliin (engl. palm detection model) ja hand landmarker -malliin. Kämmenentunnistus erottaa syötekuvasta käden ja luo siitä rajatun kuvan, johon hand landmarker -malli puolestaan luo kuvassa 12 näkyvät 21 pistettä. [35.]



Kuva 12. MediaPipen Hand Landmarkerin tunnistamat 21 pistettä käsienseurannassa [34].

Käsimalia, joka perustuu koneoppimiseen, on MediaPipen mukaan koulutettu noin 30 000:lla oikean maailman kuvalla ja myös keinotekoisesti piirretyillä käsimalleilla eri taustoja vasten. Koska kämmenen tunnistusmallin suorittaminen vie aikaa, kun ollaan video- tai suoratoistotilassa, kirjaston kokonaisuutena toimiva MediaPipen "Hand Landmarker" -tehtävä (engl. Hand Landmarker task) käyttää hand landmarker -mallin määrittelemää rajaavaa laatikkoa yhden kuvan alueena, jotta käsiä voidaan paikantaa seuraavassa kuvassa. [34.] Tätä MediaPipen kirjastoa hyödyntävä esimerkkikoodi 2 piirtää kyseiset 21 pistettä kamerasyötteessä havaittuihin käsiin, esimerkiksi kuvassa 13.

```

import cv2
import mediapipe as mp

cap = cv2.VideoCapture(0) # videosyöte
mpHands = mp.solutions.hands
hands = mpHands.Hands()
mpDraw = mp.solutions.drawing_utils

while True:
    success, image = cap.read() # kuva päivittyy kun käsiä havaittu
    imageRGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # mediapipe
    toimii RGB-väreillä eikä BGR-väreillä
    results = hands.process(imageRGB)

    # jos käsi havaittu
    if results.multi_hand_landmarks:
        for handLms in results.multi_hand_landmarks: # käsi ker-
rallaan
            for id, lm in enumerate(
                handLms.landmark
            ): # x ja y koordinaatit joka pisteestä listassa
                h, w, c = image.shape # h = korkeus, w = leveys, c =
kanava
                cx, cy = int(lm.x * w), int(lm.y * h) # keskipisteet

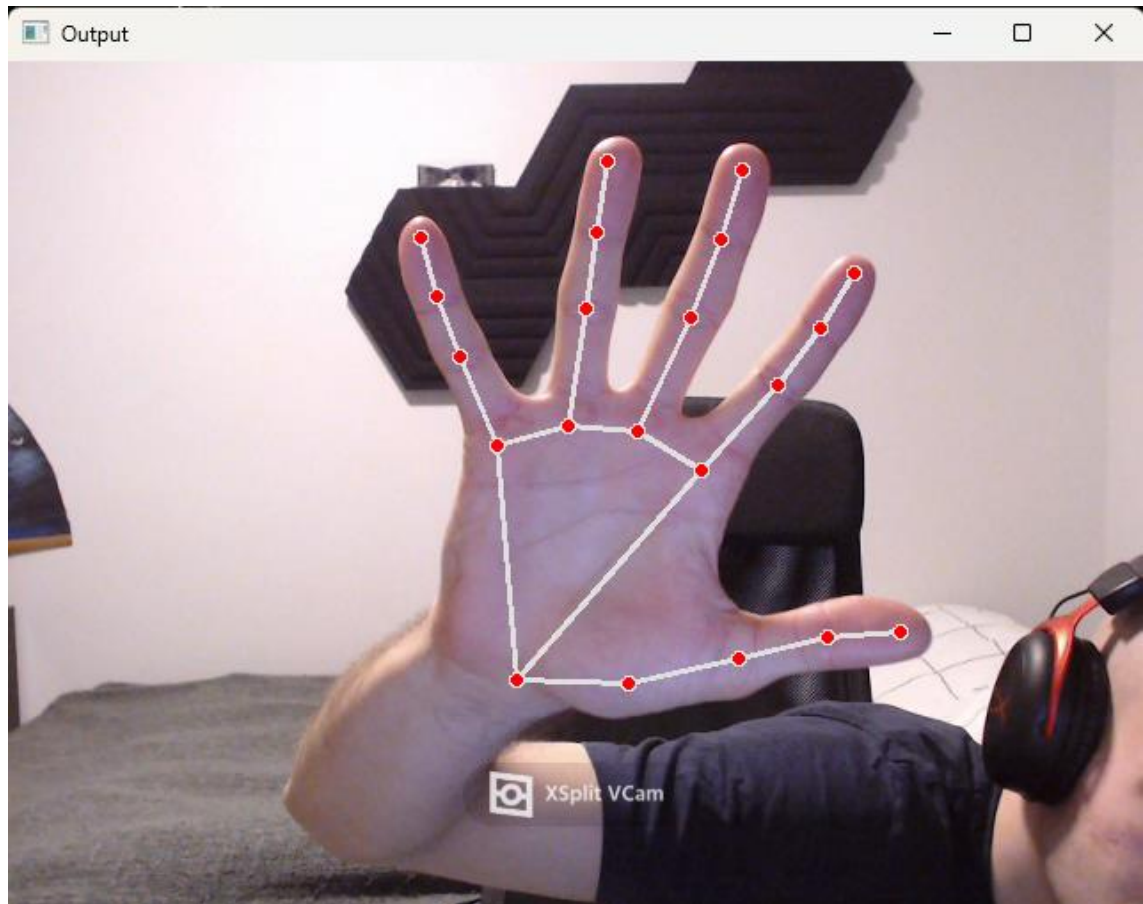
                # if id == 20: #ympyröi tietyn id:een eli pisteen kä-
dessä
                #     cv2.circle(image, (cx, cy), 25, (255, 0, 255),
cv2.FILLED)

                mpDraw.draw_landmarks(image, handLms, mpHands.HAND_CONNEC-
TIONS)

                cv2.imshow("Output", image)
                cv2.waitKey(1)

```

Esimerkkikoodi 2. Python-ohjelma, joka luo yhteyden web-kameraan ja päivittää sen kuvaa havaittujen käsien mukaan [34]. Koodi, jota on insinööriyössä hieman kommentoitu, on lainattu suoraan lähteestä.



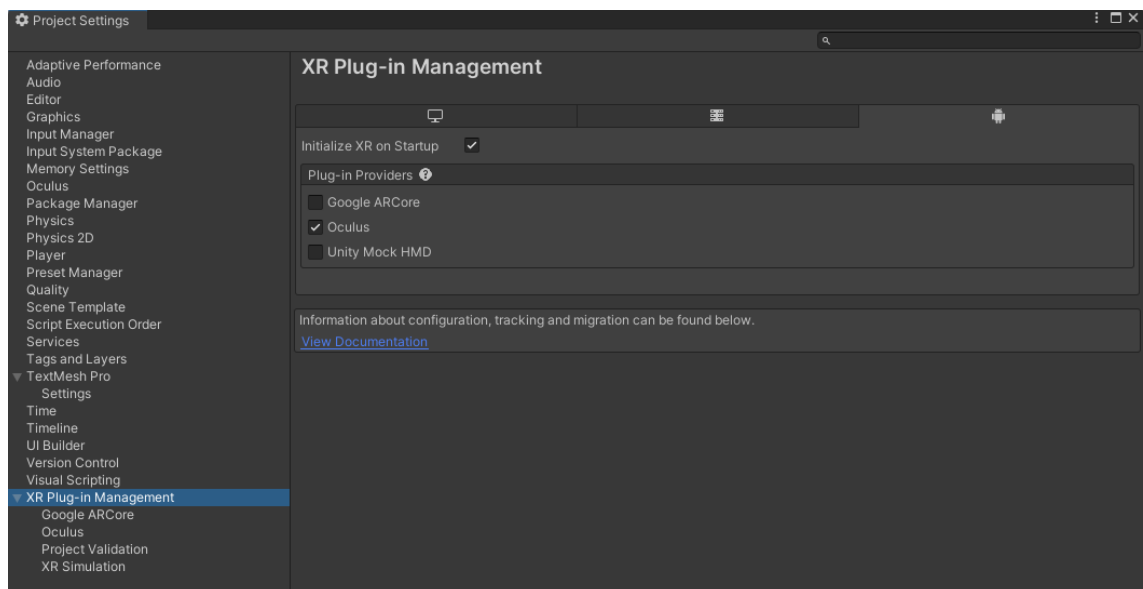
Kuva 13. Kaikki 21 pistettä piirtyvät käteen esimerkkikoodilla 2. Kuva jäätyy, kun käsiä ei havaita, ja päivittyy reaaliajassa, kun edes yksi käsi on kameran nähtävillä.

6.1 Oculus Integration -ohjelmistokehityspaketti

Jotta Metan virtuaalilasit voidaan yhdistää Unity-pelimootoriin, täytyy käyttäjän ensiksi kytkeä kehittäjätila päälle omalta Meta-tililtään Meta Quest -puhelinsovelluksesta. Kun esimerkiksi Quest 2 -lasit on yhdistetty koneeseen Oculus Link- tai vastaavalla kaapelilla, lasien valikosta kytkettäessä Oculus Link päälle näkyy pelitilassa Unityn pelinäkömyös laseissa. Tämä tekee testausiteraatioista huomattavasti vaivattomampaa verrattuna sovelluksen rakentamiseen aina erikseen jokaisen ominaisuuden jälkeen.

6.1.1 Käyttöönotto

Unity-projektin voi luoda standardiputkella (engl. standard pipeline), minkä jälkeen, kun projektin avaa, Package Managerista täytyy tuoda projektiin Oculus Integration -paketti, kun sen on ensin noutanut Unity Asset Store -kauppapaikasta. Aukeavien ikkunoiden ohjeita seuraten ja Unityn uudelleenkäynnistyksen jälkeen mennään projektiasetuksiin ja ladataan XR Plug-in Management (kuva 14), minkä jälkeen laitetaan päälle sekä PC- että Android-välilehdistä Oculus. [36.] Unity-ikkunan oikeassa alakulmassa Oculus-logoa klikkaamalla voidaan laittaa suoraan kaikki paketin suosittelemat asetukset päälle.

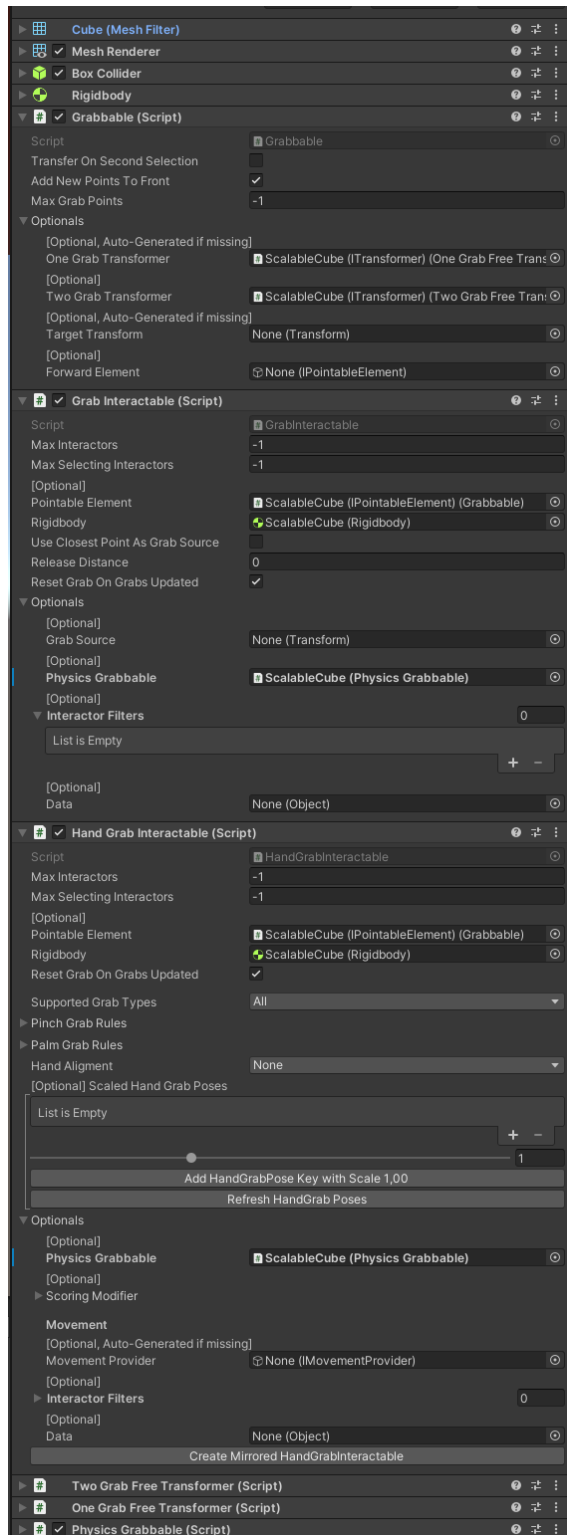


Kuva 14. XR Plug-in Management projektiasetuksissa.

Näkymästä (engl. scene) voidaan poistaa pääkamera, minkä jälkeen raahataan OculusInteractionSampleRig-valmispaketti (engl. prefab) näkymään [36]. Sen alta löytyvät OVRCameraRig- ja InputOVR-objekti. OVRCameraRig-objektin alta löytyvät ohjainmallit ja hieman käsienseurantalogiikkaa, mutta InputOVR:n alta löytyy suurin osa ohjaimilla ja käsillä tapahtuvasta vuorovaikutuslogiikasta.

6.1.2 Tarttuminen

Lisättyyn valmispakettiin lisätään HandGrabInteractor-valmispaketti, kummankin käden HandInteractors-objektin alle. Peliobjekti, johon halutaan tarttua, tarvitsee Collider-, RigidBody- ja Grabbable-komponentin. On myös aiheellista lisätä HandGrabInteractable-komponentti käsiä varten ja GrabInteractable-komponentti ohjaimia varten. [36.] Mikäli haluaa objektin fysiikkojen toimivan paremmin, kannattaa myös objektiin lisätä PhysicsGrabbable-komponentti. Heittofysiikoita varten täytyy näkymässä raahata InputOVR-objektin vasemman ja oikean käden alle HandVelocityCalculator-valmispaketti. Kuva 15 havainnollistaa, miltä tartuttavan objektin komponentit näyttävät.



Kuva 15. Kahdella kädellä skaalattavan objektin komponentit.

OVRCameraRig-objektista löytyy valmiina jo molempien käsien HandInteractors-objektin alta HandPokeInterActor, joka komponentteineen toimii vuorovaikuttajana tökkäyksessä. Tökättävällä objektilla puolestaan,

kehityspaketista löytyvää BigRedButton-objektia mallina käyttäen, tulee olla PokeInteractable-komponentti ja InteractableUnityEventWrapper-komponentti. InteractableUnityEventWrapper-komponentin alta "When Select ()" alle voi lisätä oman funktion, joka suoritetaan, kun nappia painaa. Luodaan hyvin yksinkertainen skripti, esimerkikoodi 3, joka on kytketty samaiseen nappiin, peliohjainten uudelleensynnyttämiseksi.

```
[SerializeField]
private GameObject _prefab;

[SerializeField]
private GameObject _objectInScene;

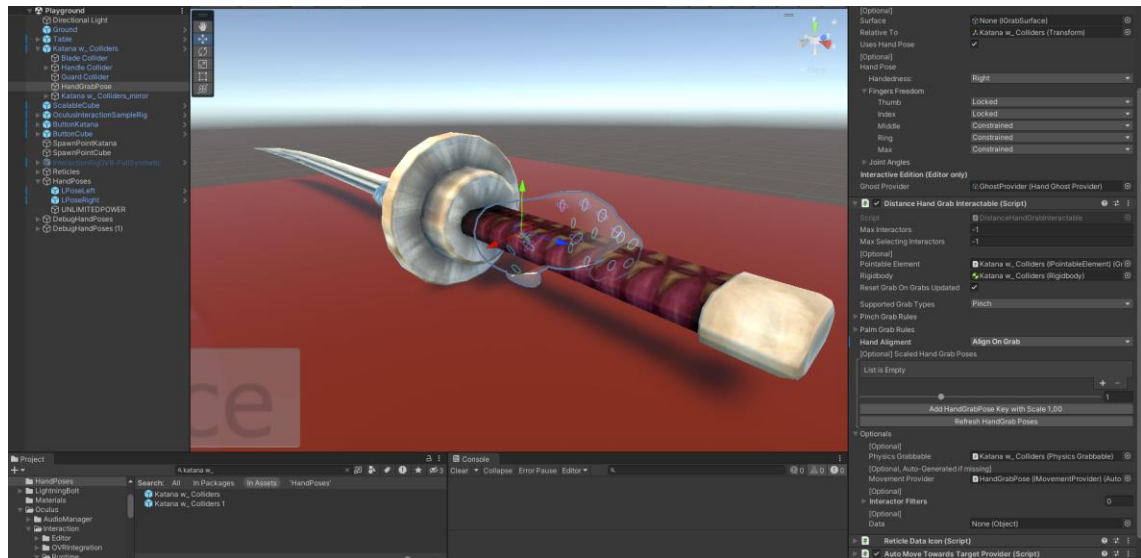
[SerializeField]
private GameObject _spawnPoint;

// Destroy the current object in scene and place a new one at the
spawnpoint
public void Spawn()
{
    GameObject clone;
    clone = Instantiate(_prefab, _spawnPoint.transform.position,
Quaternion.identity);
    Destroy(_objectInScene);

    _objectInScene = clone;
}
```

Esimerkkikoodi 3. Luotu funktio, joka löytyy näkymässä napista. Sen tehtävä on tuhota nykyinen objekti ja luoda uusi asetettuun pisteeseen, kun nappia painetaan. Tämä helpottaa esimerkiksi fysiikkatestailua, kun objektia heitetään.

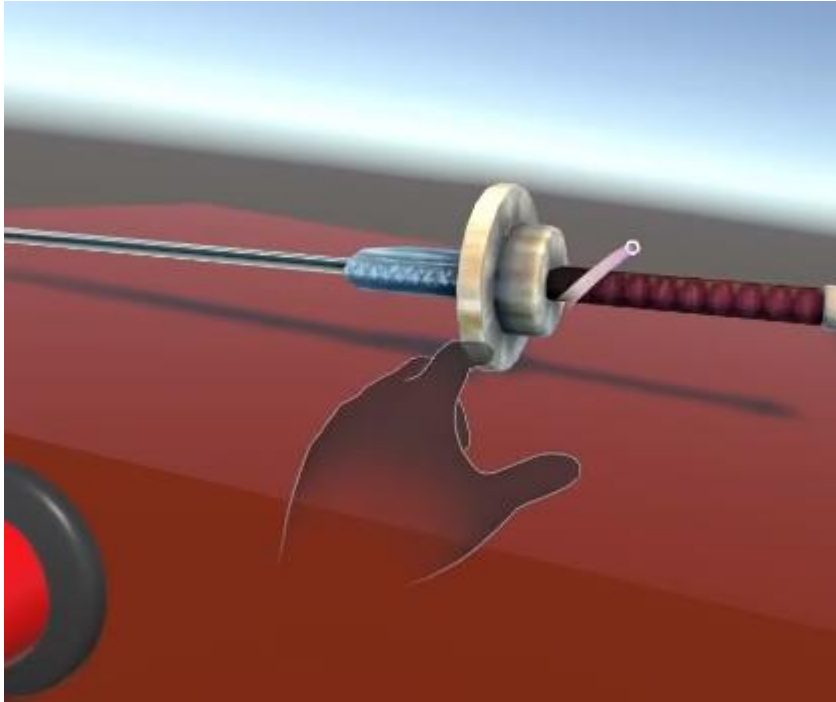
Mikäli haluaa implementoida tarttumista myös etäältä, objektille lisätään DistanceHandGrabInteractable- ja DistanceGrabInteractable-komponentti. Tartuttaviin objekteihin voi myös luoda tartuntapisteen ja asetella käsimallilla, miltä tarttuminen näyttää HandPose-komponentin avulla. [36.] Kun käsi on tarpeeksi lähellä objektia ja tarttuminen tapahtuu, objekti hakeutuu käteen ja luotu käden asento näytetään. Sormia voidaan myös erikseen lukita, jotta ei näy, jos käyttäjä liikuttaa niitä. [37.] Kuvassa 16 näkyy oikean käsimallin asennon säätöä ja objektille lisätyt etätartunnan mahdollistavat komponentit.



Kuva 16. Peukalossa ja pikkurillissä on neljä ympyrää, joista muuttaa sormen asentoa, ja muissa kolme. Miekka on osa ilmaista pakettia Unity-kauppapaidista [38].

Käsien puolella täytyy etätartunnassa `HandInteractorsLeft`- ja `HandInteractorsRight`-objektin alle lisätä `DistanceHandGrabInteractor`-valmis-paketti. Objektin liikkumisesta vastaavaksi komponentiksi voi valita `AutoMoveTowardsTargetProvider`-komponentin. Kuvassa 16 näkyy `DistanceHandGrabInteractable`-komponentti referensseineen. Objektin lisätään myös `ReticleDataIcon`-komponentti viivan piirtämistä varten. [40.]

Tämän jälkeen luodaan tyhjä peliobjekti ja lisätään sen alle `ReticleIcon`- ja `ReticleLine`-objekti. Molemmat tarvitsevat ensimmäiseen komponenttiinsa `Distance Interactorin`, joten lisätään käsikohtainen `DistanceHandGrabInteractor` niihin. `ReticleIcon`-objekti tarvitsee "Center Eye" -kohtaan näkymässä olevalta `OVRCameraRig`-objektilta `CenterEyeAnchor`-objektin. Toista kättä varten objektit voidaan kloonata ja muuttaa tarvittavat asiat toiselle kädelle. [40.] Lopputulokseksi on kuvassa 17 näkyvä kädestä objektiin piirtyvä viiva ja ympyrä osoitustussa kohdassa.

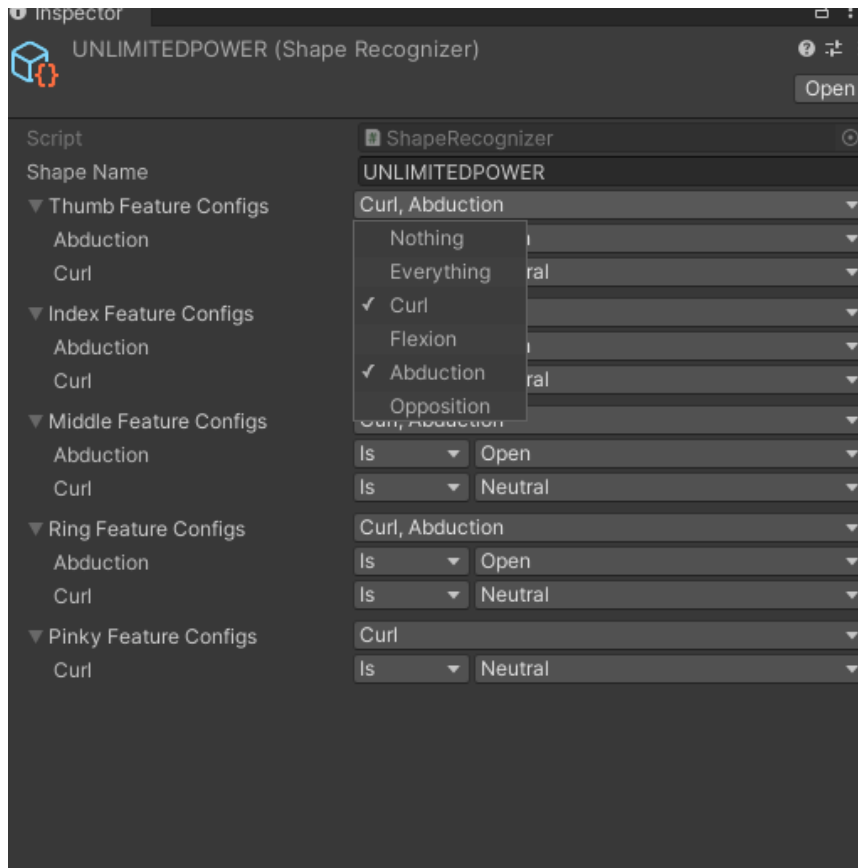


Kuva 17. Piirtyvä viiva ilmaisee, että asia, johon osoitetaan, on etätartuttava.

6.1.3 Käsieleiden hyödyntäminen

Metan paketista löytyy valmispaketteina muun muassa seuraavat käsiasennot: peukalo ylös, peukalo alas, kivi, sakset, paperi, pysäytys, L ja kehysmerkki eli kahden käden L-merkki, jossa toisen käden etusormi on toisen peukaloa vasten. Käyttäjä voi myös luoda käsiasentoja itse. Niitä varten luodaan muoto omaan kansioon navigoimalla Create, Oculus, Interaction, SDK, Pose Detection ja viimein Shape. [41.]

Jokaiselle sormelle voi erikseen säätää, kuinka koukussa sormi on kahteen ensimmäiseen niveleeseen nähden (curl), kuinka koukussa rystyseen nähden (flexion), kuinka erillään se on viereisestä sormesta (abduction) ja onko se kiinni toisessa sormessa (opposition). Jokaisen edellä mainitun voi asettaa yksinkertaistettuna kyllä- tai ei-arvoksi tai jotain siltä väliltä, esimerkiksi "Curl"-asetuksessa "Neutral", jolloin sormen kaksi ensimmäistä niveltä ovat samaisessa asennossa, kun esimerkiksi nappaisi kahvikupista kiinni. [42.] Kuvassa 18 luodaan käsimuotoa, joka vastaisi jalkapalloon tarttumista yhdellä kädellä.

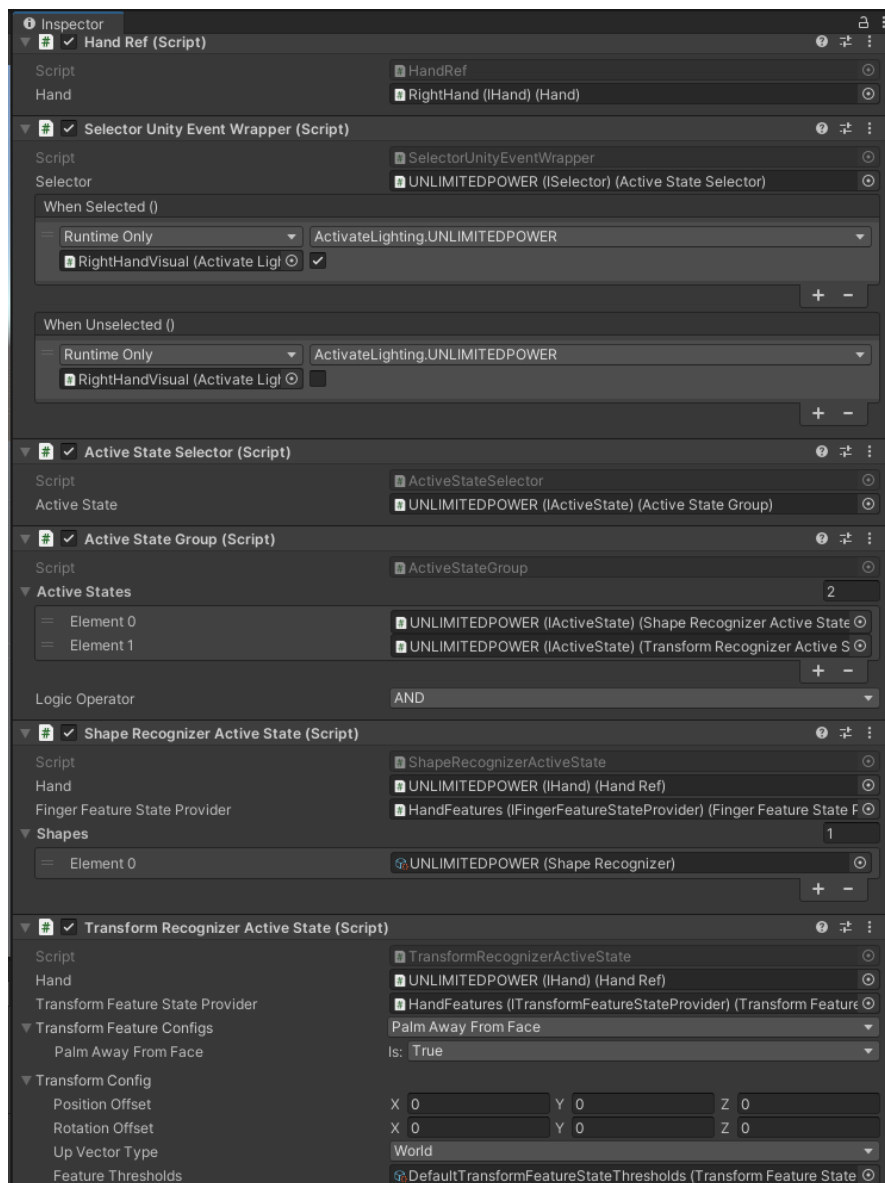


Kuva 18. Käsimuodon luominen.

Kun haluttu muoto on luotu, luodaan näkymään tyhjä peliobjekti, johon lisätään HandRef-, SelectorUnityEventWrapper-, ActiveStateSelector-, ActiveStateGroup-, ShapeRecognizerActiveState- ja TransformRecognizerActiveState-komponentti. TransformRecognizerActiveState-komponentille "Feature Thresholds" -kohtaan voidaan lisätä koodattava objekti (engl. scriptable object) DefaultTransformFeatureStateThreshold tai luoda uusi ja asettaa omia raja-arvoja. Samasta komponentista "Up Vector Type" -asetus vaihdetaan "World"-arvoon ja "Transform Feature Configs" -kohdassa voidaan laittaa lisäehdoiksi esimerkiksi, että kämmenen täytyy olla käyttäjää päin tai poispäin tai sormet tai ranne osoittavat ylös tai alas. [41.]

ShapeRecognizerActiveState-komponentti tarvitsee "Shapes"-kohtaan käsimuodon. Käsimuoto viedään myös SelectorUnityEventWrapper-komponentin "Selector"-kohtaan. ActiveStateGroupin "Active States" -kohtaan viedään

ShapeRecognizerActiveState- ja TransformRecognizerActiveState-komponentti. Molemmat komponentit tarvitsevat InputOVR:n käsimuodon kanssa käytettävän käden alta löytyvän HandFeature-objektin. ActiveStateSelector-komponentin ”Active State” -kohtaan viedään ActiveStateGroup-komponentti, ja lopuksi HandRef-komponentti tarvitsee referenssiksi käsiobjektin eli RightHand- tai LeftHand-objektin. [41.] Kuva 19 havainnollistaa, miltä komponentit näyttävät näiden toimenpiteiden jälkeen.



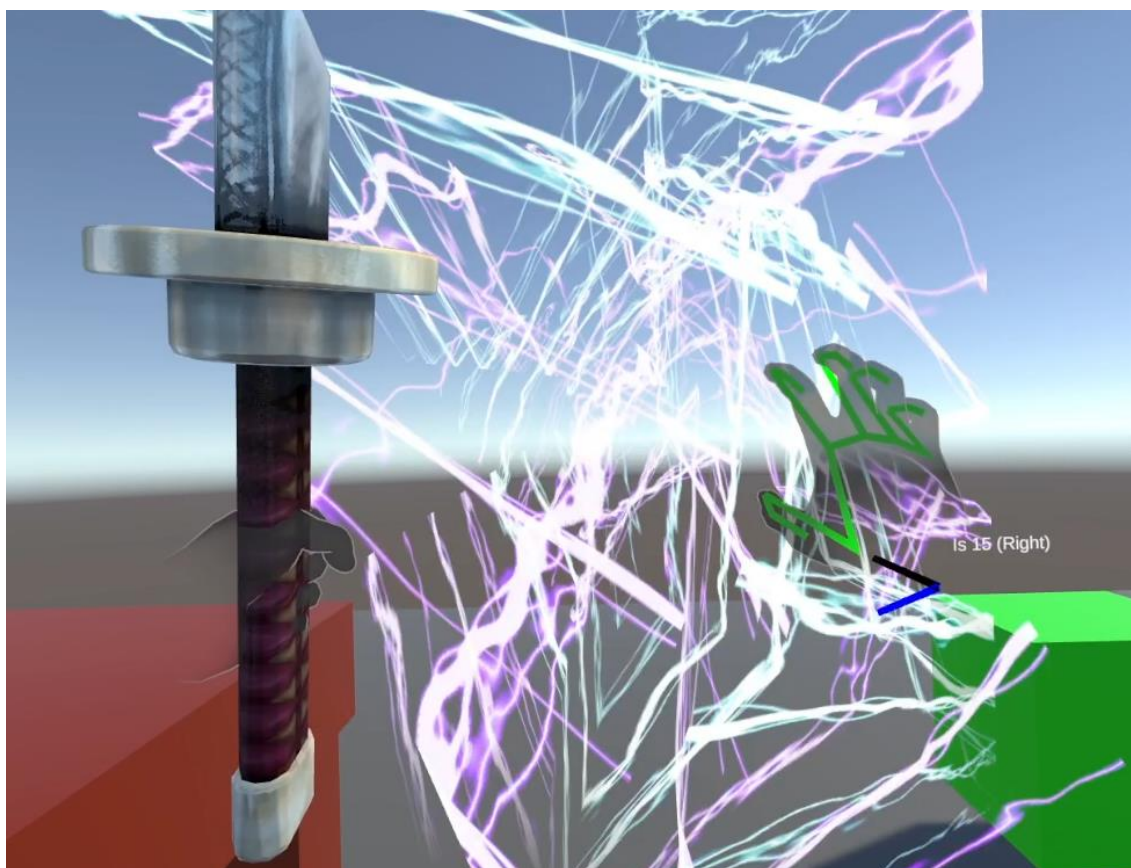
Kuva 19. Käsimuoto-objektin komponentit ja niiden referenssit.

Käsiasentojen virheenkorjausta varten luodaan tyhjä peliobjekti, jolle lisätään HandShapeSkeletalDebugVisual- ja

TransformFeatureVectorDebugParentVisual-komponentti. Ensimmäisen komponentin "Shape Recognizer Active State" -kohtaan ja toisen komponentin "Transform Recognizer Active State" -kohtaan viedään haluttu käsisiento-objekti. Tyhjälle objektille viedään lapsiksi FingerFeatureSkeletalDebugVisual- ja TransformRecognizerDebugVisual-objekti, jotka laitetaan tyhjän objektin komponenttien referensseiksi. FingerFeatureSkeletalDebugVisual-objekti tarvitsee referenssiksi käytössä olevan käden HandFeatures-objektin.

TransformRecognizerDebugVisual-objekti puolestaan tarvitsee RightHand- tai LeftHand-objektin, riippuen siitä, kumpi käsi on käytössä, lisäksi myös samaisen käden HandFeatures-objektin ja käsisiento-objektin. [41.]

Kun käsi on halutussa asennossa, viivat näkyvät vihreinä, kuten kuvassa 20. Muussa tapauksessa viivat ovat punaisia. Kuvassa näkyvät salamet ovat peräisin Unity Asset Store -kauppapaikasta ilmaisena saatavasta paketista ja tulevat näkyviin, kun sormet ovat oikeassa asennossa [43].



Kuva 20. Virheenkorjausviivojen piirtyminen käsimalliin.

Ennen kuin sovellus voidaan rakentaa, asetetaan "Build Settings" -valikosta aktiivinen näkymä painamalla "Add Open Scenes" -nappia, minkä jälkeen vaihdetaan alusta Androidille ja tekstuurin pakkaus suositellaan asetettavaksi ASTC-muotoon.

6.2 Unity XR Interaction Toolkit -paketti

Unityn oma XR-paketti tukee useimpien XR-valmistajien ohjaimia ja Unity on erikseen maininnut dokumenteissaan Meta Questin, OpenXR ja Windows Mixed Realityn [44]. OpenXR on ohjelmointirajapinta (API), joka tarjoaa moottoreille, kuten Unitylle ja Unrealille, mahdollisuuden kirjoittaa siirrettävissä olevaa koodia, joka toimii useammalla laitteella [45].

6.2.1 Käyttöönotto

Oculus Integration -ohjelmistokehityspaketin tavoin standardiputkella luotu 3D-projekti on riittävä ja projektiasetuksista asennetaan XR Plugin Management. Sieltä kytketään päälle OpenXR sekä PC- että Android-välilehdistä. Seuraavaksi lisätään Package Managerista vasemmasta yläkulmasta pluspainikkeesta XR Interaction Toolkitin viimeisin versio painamalla pluspainiketta ja valitsemalla "Add package by name", minkä jälkeen kirjoitetaan kenttään: `com.unity.xr.interaction.toolkit`. Kun paketti on ladattu, ladataan paketin "Samples"-välilehden alta "Starter Assets" ja "Hands Interaction Demo". [46.]

Koska halutaan implementoida käsienseurantaa, samalla tavalla tulee ladata myös XR Hands -paketti kirjoittamalla kenttään: `com.unity.xr.hands`. Tästä paketista halutaan myös välilehden "Samples" alta ladata HandVisualizer-paketti. Tämän jälkeen mennään takaisin projektiasetuksiin ja kytketään päälle OpenXR:n alta "Hand Tracking Subsystem" ja "Meta Hand Tracking Aim" PC-välilehden lisäksi myös Android-välilehdestä. Androidissa "Meta Quest Support" -valinta täytyy myös laittaa päälle, ja molemmissa välilehdissä lisätään halutut ohjainprofiilit "Interaction Profiles" alta. [46.]

Hakemistosta raahataan "Complete XR Origin Hands Set Up" -valmispaketti näkymään. Näkymästä voidaan poistaa pääkamera ja lisätä alusta (engl. plane). Valmispaketin "HandVisualizer"-lapselta voidaan kytkeä "Debug Draw Joints", mikäli halutaan nähdä käsissä erikseen jokainen seurattava piste, ja vauhtiviivat (eli mihin suuntaan käsi liikkuu) saa kytkettyä päälle asettamalla "Velocity Type" -asetuksen esimerkiksi arvolle "Linear". [46.]

Näkymä, johon on jo luotu pöytä, laatikko ja miekka, käyttää kehityspaketin esimerkeissä tarjottua "PokeButton" -valmispakettia. Napin painaminen onnistuu jo, sillä XR-valmispaketissa on vasemman ja oikean käden lapsina "Poke Interactor" eli vuorovaikuttaja, ja napissa, joka toimii vuorovaikutteisena, puolestaan XRSimpleInteractable- ja XRPokeFilter-komponentti. Vuorovaikutustapahtumien luominen on yksinkertaista hyödyntämällä vuorovaikuttajan ja vuorovaikutteisen välille Unityn tarjoamia tapahtumia (engl. events). Esimerkiksi PokeButton-valmispaketin XRSimpleInteractable-komponentin alta löytyy "Interactable Events" ja XR-valmispaketin molempien käsien PokeInteractorin alta löytyy "Interactor Events". [47.]

6.2.2 Tarttuminen

Tarttumista varten peliobjektiin lisätään XRGrabInteractable-komponentti, ja mikäli sen halutaan toimivan molemmilla käsillä yhtäaikaisesti, täytyy valikosta "Select mode" valita "Multiple"-asetus. Käsien puolella tarttumisen havaitsee "Direct Interactable". Objektia voidaan myös skaalata muuttamalla sen kokoa kahdella kädellä, jos lisätään XRGeneralGrabTransformer-komponentti ja siitä laitetaan päälle "Allow Two Handed Scaling" -asetus. Käsillä on myös mahdollista tarttua objekteihin säteen avulla, joka lähtee käsimalleista. Kun kytkee "Force Grab" -asetuksen päälle, objekti hakeutuu käteen, kun sitä osoitetaan säteellä ja tehdään tarttumisele.

Objektista menettää tällä hetkellä valitettavan usein tartunnan. Paketti on varsin tuore, sillä Valem Tutorials YouTube-kanavan tekijä sanoo videossaan, joka on julkaistu maaliskuussa 2023, että hän on odottanut Unityn XR toolkitin

käsienseurantatukea ja viimein pääsee tekemään aiheesta videon [46]. Muiden ominaisuuksien tekemistä voisi jatkaa, mutta tarttuminen on sen verran olennainen ominaisuus, että mikäli käyttää Metan laitteita, käyttäjän on helpompi lähteä liikkeelle vuosia hiotulla Oculus Integration SDK -paketilla.

7 Yhteenveto

Insinööriyössä tutustuttiin XR-alan innovaatioihin ja suosituimpien yritysten laitteisiin. Työssä keskityttiin kuitenkin pääasiassa siihen, kuinka käsien ja ohjaimien seuranta tapahtuu virtuaalitodellisuudessa ja mitkä asiat vaikuttavat seurantaan. Tavoitteena oli myös tutkia, kuinka helposti valmispaketteja hyödyntämällä pääsee alkuun sovelluskehityksessä Unity-pelimoottorilla.

Ilmeni, että Meta Quest 2- ja Valve Index -lasit ovat nykyäänkin suosituimpien laitteiden joukossa. Virtuaalilaitteiden ohjaimia seurataan IMU-sensorin ja IR-LEDien avulla joko sisältä ulospäin tapahtuvan seurannan avulla tai huoneeseen ripustettavien tukiasemien avulla. Metan uusimmissa ohjaimissa ja HTC:n koko kehon seurannan mahdollistavissa laitteissa on kuitenkin omat kameransa ja prosessorinsa, jotka lisäävät seurantatarkkuutta.

Käsienseurantaan vaaditaan sisältä ulospäin seurantaan kykenevät lasit, joiden kamerat, prosessointiteho ja toimiva ohjelmisto voivat seurata käsien ja sormien liikettä. Seurantatehokkuutta rajaavat valoisuus, laitteiston tehokkuus, ohjelmiston tehokkuus, laitteiden kalibrointi ja käsien sijainti. Käsien ja ohjaimien seurantaan vaikuttavien seikkojen tutkimista voisi jatkaa omilla virtuaalilaseilla esimerkiksi eri valo-olosuhteissa.

Kun insinööriyössä vertailtiin Oculus Integration -ohjelmistokehityspakettia ja Unity XR Interaction Toolkit -pakettia, huomattiin, että Oculus Integration -ohjelmistokehityspaketti on aloittelijaystävällinen ja siihen pääsee käsiksi ilman koodaustausta varsin nopeasti. Tuloksena molemmilla paketeilla saatiin pohja, jossa käsienseuranta toimii, objektien kanssa voidaan olla vuorovaikutuksessa ja käsieleitä osataan luoda Oculus Integration -paketilla. Unity XR Interaction

Toolkit -paketin käsienseurannan tarttumisominaisuudessa on vielä parantamisen varaa, sillä kappaleesta menettää otteen valitettavan usein, minkä vuoksi käsieleitäkään ei työssä lähdetty paketilla toteuttamaan. Unity XR Interaction Toolkit -paketilla voisi tulevaisuudessa jatkaa kehitystä, kun paketti päivittyy. Oculus Integration -paketilla aikaansaadulla pohjalla voisi lähteä jo kehittämään esimerkiksi fantasiapeliä.

Lähteet

- 1 Sagarm, Khillar. 2019. Difference Between Augmented Reality and Mixed Reality. Verkkoaineisto. Difference Between. <<http://www.differencebetween.net/technology/difference-between-augmented-reality-and-mixed-reality/>>. Luettu 7.9.2022.
- 2 Or, Barak. 2021. What is IMU? Verkkoaineisto. Towards Data Science. <<https://towardsdatascience.com/what-is-imu-9565e55b44c>>. Julkaistu 31.7.2021. Luettu 2.1.2023.
- 3 Electromagnetic Waves. Verkkoaineisto. CRISP. <<https://crisp.nus.edu.sg/~research/tutorial/em.htm#spectrum>>. Luettu 7.9.2022.
- 4 What is virtual reality all about? 2021. Verkkoaineisto. Oculus. <<https://www.oculus.com/blog/what-is-virtual-reality-all-about/>>. Luettu 12.9.2022.
- 5 The Teslasuit literally shocked me | Beta Test. 2018. Verkkoaineisto. CNET. YouTube. <<https://www.youtube.com/watch?v=KC83goKZXK8>>. Katsottu 13.9.2022.
- 6 Stewart, Jude. 2022. VR Still Stinks Because It Doesn't Smell. Verkkoaineisto. Wired. <<https://www.wired.com/story/vr-still-stinks-because-it-doesnt-smell/>>. Julkaistu 31.7.2022. Luettu 13.9.2022.
- 7 Lang, Ben. 2023. HTC Announces Inside-out Tracker for VR Accessories & Body Tracking. Verkkoaineisto. Road to VR. <<https://www.roadtovr.com/htc-vive-tracker-standalone-inside-out-camera-tracking/>>. Julkaistu 20.3.2023. Luettu 1.5.2023.
- 8 Bailenson, Jeremy. 2018. Experience On Demand: What Virtual Reality Is, How It Works, and What It Can Do. New York: W. W. Norton & Company.
- 9 Overview of Positional Tracking Technologies for Virtual Reality. 2014. Verkkoaineisto. Road to VR. <<https://www.roadtovr.com/overview-of-positional-tracking-technologies-virtual-reality/>>. Julkaistu 2.6.2014. Luettu 23.9.2022.
- 10 Stein, Scott. 2023. Verkkoaineisto. CNET. <<https://www.cnet.com/tech/gaming/best-vr-headsets/>>. Julkaistu 18.3.2023. Luettu 18.4.2023.

- 11 Circelli, Ruben. 2022. How to Connect PSVR to a PC. Verkkoaineisto. Lifewire. <<https://www.lifewire.com/how-to-connect-psvr-to-a-pc-5197483>>. Julkaistu 14.6.2022. Luettu 7.3.2023.
- 12 Seitz, Patrick. 2022. Facebook Parent Stands Tall In Small Market For Virtual Reality. Verkkoaineisto. Investor's Business Daily. <<https://www.investors.com/news/technology/virtual-reality-meta-dominates-a-small-market/>>. Julkaistu 22.4.2022. Luettu 4.12.2022.
- 13 Varjo. Verkkoaineisto. <<https://varjo.com/>>. Luettu 18.4.2023.
- 14 Lang, Ben. 2022. One of the Last Bastions of the Oculus Brand is No More. Verkkoaineisto. Road to VR. <<https://www.roadtovr.com/meta-quest-rebranding-oculus-com-website/>>. Julkaistu 25.4.2022. Luettu 5.2.2023.
- 15 Buy Meta Quest 2. Get Two hit games. Verkkoaineisto. Meta. <<https://www.meta.com/fi/en/quest/products/quest-2/>>. Luettu 25.2.2023.
- 16 What is 6DOF? Verkkoaineisto. VRdirect. <<https://www.vrdirect.com/wiki/what-is-6dof/>>. Päivitetty 19.4.2023. Luettu 29.4.2023.
- 17 Stein, Scott. 2022. The Quest 2 Is More Expensive, But It's Still the Best VR Headset For Now. Verkkoaineisto. CNET. <<https://www.cnet.com/tech/gaming/facebook-oculus-quest-2-vr-review-one-of-my-favorite-game-consoles/>>. Julkaistu 12.8.2022. Luettu 25.2.2023.
- 18 Stein, Scott. 2019. Valve Index's new VR controllers feel like the future of gaming. Verkkoaineisto. CNET. <<https://www.cnet.com/tech/gaming/valve-indexs-new-vr-controllers-feel-like-the-future-of-gaming/>>. Julkaistu 30.5.2019. Luettu 25.2.2023.
- 19 Model IMU, GPS, and INS/GPS. Verkkoaineisto. MathWorks. <<https://www.mathworks.com/help/nav/ug/model-imu-gps-and-ins-gps.html>>. Luettu 5.2.2023.
- 20 Hall, Nancy. 2022. Aircraft Rotations. Verkkoaineisto. Glenn Research Center. <<https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/aircraft-rotations/>>. Luettu 5.2.2023.
- 21 Kozyrkov, Cassie. 2020. What is "Ground Truth" in AI? (A warning.). Verkkoaineisto. Towards Data Science. <<https://towardsdatascience.com/in-ai-the-objective-is-subjective-4614795d179b>>. Julkaistu 28.2.2020. Luettu 19.4.2023

- 22 How do common virtual reality tracking systems work? Verkkoaineisto. Mechatech. <<https://www.mechatech.co.uk/journal/how-do-common-virtual-reality-tracking-systems-work>>. Luettu 5.2.2023.
- 23 Blob Detection Using OpenCV (Python, C++). Verkkoaineisto. LearnOpenCV. <<https://learnopencv.com/blob-detection-using-opencv-python-c/>>. Luettu 19.4.2023.
- 24 Bushouse, Howard. Figure 5. Verkkoaineisto. ResearchGate. <https://www.researchgate.net/figure/The-IR-channel-blob-mask-Pixels-impacted-by-blobs-are-marked-with-a-value-of-512-in-the_fig7_252450570>. Katsottu 19.4.2023.
- 25 The Quest 2's Biggest Improvement Yet! 2023. Verkkoaineisto. ThrillSeeker. YouTube. <<https://www.youtube.com/watch?v=5vEwU8kKh4I>>. Katsottu 20.4.2023.
- 26 What Is SLAM? Verkkoaineisto. MathWorks. <<https://se.mathworks.com/discovery/slam.html>>. Luettu 21.4.2023.
- 27 What is Hand Tracking in VR? 2021. Verkkoaineisto. Ultraleap. <<https://www.ultraleap.com/company/news/blog/hand-tracking-in-vr/>>. Julkaistu 7.5.2021. Luettu 7.3.2023.
- 28 How Hand Tracking Works. 2020. Verkkoaineisto. Ultraleap. <<https://www.ultraleap.com/company/news/blog/how-hand-tracking-works/>>. Julkaistu 3.9.2020. Luettu 7.3.2023.
- 29 Introducing Varjo XR-3, the only true mixed reality headset. Verkkoaineisto. Varjo. <<https://varjo.com/products/xr-3/>>. Luettu 26.4.2023.
- 30 How ToF Works. Verkkoaineisto. ArduCam. <<https://www.arducam.com/time-of-flight-camera-raspberry-pi/>>. Luettu 16.4.2023.
- 31 Hand tracking privacy notice. Verkkoaineisto. Meta. <<https://www.meta.com/en-gb/help/quest/articles/accounts/privacy-information-and-settings/hand-tracking-privacy-notice/>>. Luettu 7.3.2023.
- 32 Why The Best Hand Tracking Camera Is Infrared. 2021. Verkkoaineisto. Ultraleap. <<https://www.ultraleap.com/company/news/blog/best-hand-tracking-camera/>>. Julkaistu 13.8.2021. Luettu 25.4.2023.
- 33 Ranganathan, Ananth. 2022. The Oculus Insight positional tracking system. Verkkoaineisto. AI Accelerator Institute. <<https://www.aiacceleratorinstitute.com/the-oculus-insight-positional-tracking-system-2/>>. Julkaistu 27.6.2022. Luettu 1.5.2023.

- 34 Kiruri, Simon. 2022. Creating a Hand Tracking Module using Python, OpenCV, and MediaPipe. Verkkoaineisto. Section. <<https://www.section.io/engineering-education/creating-a-hand-tracking-module/>>. Julkaistu 11.2.2022. Luettu 16.4.2023.
- 35 Hand landmarks detection guide. Verkkoaineisto. MediaPipe. <https://developers.google.com/mediapipe/solutions/vision/hand_landmarker>. Luettu 18.4.2023.
- 36 How To Interact in VR – Oculus Interaction SDK – PART 1. 2022. Verkkoaineisto. Valem Tutorials. YouTube. <<https://www.youtube.com/watch?v=1FRqniErAfs>>. Katsottu 22.4.2023.
- 37 Create a Hand Grab Pose (PC). Verkkoaineisto. Meta. <<https://developer.oculus.com/documentation/unity/unity-isdk-creating-handgrab-poses/>>. Luettu 29.4.2023.
- 38 Seven Swords – Seven Stylized Swords. Verkkoaineisto. Unity Asset Store. <<https://assetstore.unity.com/packages/3d/props/weapons/seven-swords-seven-stylized-swords-14304#description>>. Luettu 1.5.2023.
- 39 How To Grab And Throw Objects In VR – Interaction SDK #4. 2022. Verkkoaineisto. Dilmer Valecillos. YouTube. <<https://www.youtube.com/watch?v=Ril-5dWBOSU>>. Katsottu 29.4.2023.
- 40 Setup Distance Grabbing In VR – Interaction SDK #5. 2022. Verkkoaineisto. Dilmer Valecillos. YouTube. <<https://www.youtube.com/watch?v=r7ikUOgjZIE>>. Katsottu 29.4.2023.
- 41 Oculus Interaction SDK Hand Pose Detection With Custom Hand Poses. 2022. Verkkoaineisto. Dilmer Valecillos. YouTube. <<https://www.youtube.com/watch?v=Duci6GkSHJ4>>. Katsottu 29.4.2023.
- 42 Hand Pose Detection. Verkkoaineisto. Meta. <<https://developer.oculus.com/documentation/unity/unity-isdk-hand-pose-detection/>>. Luettu 30.4.2023.
- 43 Lightning Bolt Effect for Unity. Verkkoaineisto. Unity Asset Store. <<https://assetstore.unity.com/packages/tools/particles-effects/lightning-bolt-effect-for-unity-59471>>. Luettu 1.5.2023.
- 44 XR Interaction Toolkit. 2018. Verkkoaineisto. Unity. <<https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.3/manual/index.html>>. Päivitetty 10.2.2023. Luettu 24.4.2023.

- 45 OpenXR. Verkkoaineisto. Microsoft. <<https://learn.microsoft.com/en-us/windows/mixed-reality/develop/native/openxr#what-is-openxr>>. Luettu 24.4.2023.
- 46 Hand Tracking with Unity XR Interaction Toolkit. 2023. Verkkoaineisto. Valem Tutorials. YouTube. <https://www.youtube.com/watch?v=mJ3fygb9Aw0&t=1s&ab_channel=ValemTutorials>. Katsottu 23.4.2023.
- 47 Poke Interaction with Unity XR Toolkit. Verkkoaineisto. 2023. Valem Tutorials. YouTube. <https://www.youtube.com/watch?v=yUm-rtwflmeg&ab_channel=ValemTutorials>. Katsottu 24.4.2023.

