

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Vladimir Makarenkov

Segareaalsuse seadmed: piirangud ja võimalused

Bakalaureusetöö (9 EAP)

Juhendaja: Madis Vasser, PhD

TARTU 2025

Segareaalsuse seadmed: piirangud ja võimalused

Lühikokkuvõte:

Käesolevad bakalaureusetöös uuriti Meta Quest 3 seadme alusel segareaalsuse rakenduste arendamise võimalusi ja piiranguid. Võimaluste ja piirangute väljaselgitamiseks kasutati kaamera pildi läbilaske võimalust (passthrough) hõlmavaid Meta näidisprojekte ja uurimise käigus valminud efekte. Töös käsitletud kaamera ligipääs jaguneb kaheks viisiks - Passthrough API ja Passthrough Camera API. Mõlema viisi kohta uuriti näidisprojekte ja loodi autori poolt näidiseid. Analüüsiti võimalusi Passthrough API kasutamiseks ning arendajate poolt piiratud ja ühetüübilist funktsionaalsust. Samuti uuriti võimalusi Passthrough Camera API kasutamiseks ning piiranguid, mis võivad segada kasutajate sukeldumist segareaalsusesse.

Võtmesõnad: Segareaalsus, Unity, Unreal Engine 5, Meta Quest 3

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Mixed reality devices: restrictions and opportunities

Abstract:

This bachelor's thesis investigated the opportunities and restrictions of developing mixed reality applications based on the Meta Quest 3 device. To identify the opportunities and restrictions, Meta sample projects involving the Passthrough camera and effects created during the research were used. The camera access discussed in the thesis is divided into two methods - Passthrough API and Passthrough Camera API. For both methods, sample projects were studied and samples were created by the author. The possibilities for using the Passthrough API and the limited and one-type functionality provided by developers were analyzed. Also the potential uses for the Passthrough Camera API and the limitations that may interfere with users' immersion in mixed reality were studied.

Keywords: Mixed reality, Unity, Unreal Engine 5, Meta Quest 3

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

Sisukord

Sissejuhatus	4
1. Teoreetiline ülevaade	6
1.1 Segareaalsus	6
1.2 Segareaalsuse seadmed	7
1.3 Meta Quest 3	8
1.4 Unreal Engine 5	8
1.5 Unity	9
2. Metoodika	10
2.1 Keskkonna seadistamine	10
2.1.1 Meta Quest 3 seadistamine	10
2.1.2 Unreal Engine 5	11
2.1.3 Unity	12
2.2 Dokumentatsioonid ja näited	12
2.2.1 Meta for Developers dokumentatsioonid (MetaHorizon)	13
2.2.2 Unreal-PassthroughSample	14
2.2.3 Unity-PassthroughCameraAPISamples	16
2.3 Efektide loomine	17
2.3.1 Unreal Engine 5	17
2.3.2 Unity	20
3. Analüüs	23
3.1 Piirangute analüüs	23
3.2 Võimaluste analüüs	24
Kokkuvõte	26
Viidatud kirjandus	27
Lisad	30
Lisa 1. GitHub repositooriumi MixedRealityEffects	30
Litsents	31

Sissejuhatus

Sega- ja virtuaalreaalsuse seadmete populaarsus on üle aja kasvanud ja on leidnud laialdast kasutust erinevates valdkondades [3]. Ettevõteted võtavad kasutusele aina rohkem segarealaalsuse seadmeid, et tõsta oma töötajate efektiivsust ja töötlust, töötada välja uusi lahendusi ning madaldada kulusid [3]. Haridusasutused kasutavad neid selleks, et tagada õpilaste keskendumist teemasse läbi interaktiivsuse ja tänu sellele vähendada raskusi õppimises [3]. Segarealaalsuse kasutamine simulaatoritena või õppimisvahenditena on ka viis, kuidas asutused erinevatest valdkondades, nagu tervishoid, saavad kasutada seadmeid [3]. Sellest tuleneb, et segarealaalsuse seadmed ei ole enam kasutuses ainult meelelahutuseks ja ajaviiteks, vaid nendest on kujunenud tööriistad, mis on mõneski valdkonnas näidanud laialdast rakendust ja efektiivsust.

Käesoleva bakalaureuse töö eesmärk on anda eestikeelne ülevaade hetkeseisundist segarealaalsuse arendamise võimalustest ja piirangutest Meta Quest 3 virtuaalreaalsuse seadme, mis toetab ka segarealaalsust, alusel. Kuna tegemist on kõige hiljutisema Meta virtuaalreaalsuse seadme mudeliga (arvestamata Meta Quest 3S-i, mis on Meta Quest 3 taskukohasem versioon), siis rakenduste arendamine just sellele seadmele on eriti aktuaalne.

Sega- ja virtuaalreaalsuse seadmete ja funktsionaalsusega seotud töid on palju. Eriti suur materjali kogus on leitav inglise keeles. Antud lõputöö on kirjutatud eesti keeles ning selle tulemus kehtib spetsiaalse tarkvara, versiooni ja seadme kohta.

Bakalaureusetöö jaguneb praktiliseks ja teoreetiliseks osaks. Praktilise osa raames piirangute ja võimaluste selgitamiseks uuritakse ja luuakse erinevaid efekte segarealaalsuse seadmele Meta Quest 3. Läbitakse erinevaid samme nagu seadme ja tarkvara seadistamine arendamise jaoks, arendamise ajal tekkivate erinevate probleemide lahendamine, dokumentatsioonide ja juhendite abil efektide loomine, olemasolevate lahenduste uurimine ja nende testimine seadmel. Teoreetiline osa jaguneb omakorda mitmeks osaks, kus alguses antakse taustainformatsiooni segarealaalsusest, seadmete tööst, kasutusvaldkondadest ja eripäradest. Metoodika peatükk selgitab, kuidas praktilise osa raames uuriti segarealaalsuse seadmete võimalusi ja piiranguid. Analüüsi peatükis kirjeldatakse ja analüüsitakse täpsemalt, mida

saavutati praktilise töö raames. Kokkuvõtte peatükis võetakse kokku kõikides peatükkides käsitletud teemasid ja tulemusi.

1. Teoreetiline ülevaade

1.1 Segareaalsus

Laiendatud reaalsus (XR - extended reality) jaguneb virtuaalseks reaalsuseks (VR - virtual reality), liitreaalsuseks (AR - augmented reality) ja segareaalsuseks (MR - mixed reality), mis erinevad üksteisest virtuaalse ja reaalse maailma ühendamisviiside poolest [1, 4]. Virtuaalne reaalsus ühendab kasutajat ainult digitaalse keskkonnaga, milles kasutaja saab realistliku kogemuse virtuaalses keskkonnas koostöös esemetega ning kus virtuaalne reaalsus stimuleerib erinevaid kasutajate meeli sukeldumisel uude keskkonda [5]. Liitreaalsus omakorda toob digitaalsed esemed reaalsesse maailma, kuid elemendid üksteisega kokku ei puutu [1]. Segareaalsus asub virtuaal- ja liitreaalsuse vahel ning ühendab endas mõlema tehnoloogia omadusi, kus virtuaalsed elemendid interakteeruvad tegelikkusega [2]. Näiteks virtuaalse raamatu saab kinnitada kindlasse kohta pärismaailmas, kust seda saab igal ajahetkel leida [4].

Segareaalsuse klassifitseerimise alla kuuluvad kõik liitreaalsuse ja virtuaalreaalsuse vahele jäävad näited [1]. Segareaalsuse eristamiseks on olemas kolm omadust [2]:

- Esimene omadus on immersioon, mis tähendab seda, et kasutaja ümbrust ja keskkonda töödeldakse reaalselt;
- Teiseks on interaktsioon, mis peab toimuma loomulikult kasutaja jaoks;
- Kolmandaks on informatsioon, mis tähendab, et ajas ja ruumis eksisteerivad digitaalsed objektid asuvad kasutaja keskkonnas ning neid on võimalik mõjutada.

Rokhsaritalemi jt [2] järgi on segareaalsusel mitu eelist võrreldes liit- ja virtuaalreaalsuse tehnoloogiatega, nagu võimalus kaardistada keskkonda. Keskkonna kaardistamise tulemusena kasutaja interakteerub nii virtuaalse kui ka reaalse maailmaga, arvestades oma asukohta ja aega ruumis. Siiski on Rokhsaritalemi jt [2] sõnul selle arendamisel ka erinevaid probleeme. Peamiseks toovad nad välja pildi väljastamise, mille jaoks on vajalik korralik resolutsioon ja pildi kontrast. Samuti osutus probleemiks erinevate jälgimistehnoloogiate ja lahenduste loomine, mis tagab korrektse reaalse ja virtuaalse keskkonna vahelise suhtlemise [2].

Segareaaalsuse tehnoloogia on tekitanud suurt huvi ning saavutanud laialdaselt kasutust ärivaldkonnas [3]. Tänu sellele suured firmad madaldavad kulusid, parandavad kommunikatsiooni-, tööstus- ja tootlusprotsesse [3, 4]. Näiteks Mercedes Benz kasutab segareaaalsust kaugtoe jaoks [3]. Tehnoloogia kasutamise võimalusi on ka teistes valdkondades samuti töö kvaliteedi, kiiruse ja tõhususe parandamiseks [3]. Turunduse valdkonnas on segareaaalsus pakkunud eriti palju uusi lahendusi, kus nüüd on võimalus proovida, näidata ja arutada tooteid virtuaalselt [3, 4].

Segareaaalsus leiab kasutust ka tarbijaturul [3], kus seda rakendatakse erinevates valdkondades nagu haridus, navigatsioon ja meelelahutus [3, 4]. Võimalus luua midagi virtuaalselt, aga samal hetkel jättes võimalust sellega tegeleda loob mängulise ja meelelahutusliku keskkonna, mida kasutatakse näiteks õpetamises [3, 4], kus õpilased töötavad läbi materjali segareaaalsuse abil, saades seletusi ja näiteid, mis lisavad interaktiivsust.

Eksperdid hindavad segareaaalsuse turgu väga kõrgelt ja ennustavad sellele kasvu [3]. Hetkel mängib suurt rolli mängude valdkond, aga tulevikus on oodata teiste valdkondade poolt ülevõtmist ning edasist arendamist [3, 4]. Tõusvaks trendiks on segareaaalsuse võime tuvastada loomulike tunnuseid, mis toob uusi rakendusvõimalusi uutes keskkondades [3]. Suurte ja keeruliste seadmete asemel on samuti oodatud lihtsamate ja kättesaadavate seadmete arendamine ja integreerimine [3]. Nii tehnoloogia kui ka tarkvara arengu ja kättesaadavuse abil on oodatud uute tarkade prillide kasutuse kasvu USA töötajate seas ning samuti ka segareaaalsuse kasutamist igapäeva elu lahutamatuks osaks [4].

1.2 Segareaaalsuse seadmed

Rokhsaritalemi jt [2] järgi omavad laiendatud reaalsuse seadmed erinevaid andureid ja mehhanisme, mille abil tagatakse reaalse ja virtuaalse keskkonna ühinemist. Nende seadmete üldised omadused hõlmavad positsiooni jälgimist ja jälgimissensorite kasutamist. Positsiooni jälgimise abil saadakse teada kasutaja asukohta ja pöörlemisliigutusi. Jälgimissensorite abil saab jälgida pea, käte, silmade, kõne ja ka ümbruskonna liikumist. Interaktsiooniks kasutatakse kas kontrollereid (VR, AR) või andurid loevad käteliigutusi ja žeste (AR, MR) [4].

Segareaalsuse arendamiseks kasutatakse kahe erineva tehnoloogia tüübiga seadmeid, kus ühe puhul kuvatakse reaalselt maailma otse läbi klaasi ja teise puhul kasutatakse videopõhist läbivaadet läbi ekraani. Ekraanipõhised seadmed jagunevad omakorda samuti erinevatesse kategooriatesse (peaseadmed, käeshoitavad seadmed, monitori- ja projektsioonipõhised ekraanid), mis erinevad mugavuse ja sukeldumise tagamise poole pealt ning mõjutavad läbi selle kasutaja kogemust segareaalsusega [2]. Segareaalsus suuresti tugineb pilvearvutustel ning tehisintellektil, mille tulemus edastatakse seadmete kaudu, näiteks läbi peaseadme. Niisuguste peaseadmete hind võib varieeruda vahemikus 300 - 3000 dollarit. Ka teised seadmed on varustatud erinevate komponentidega, kus komponentide arv määrab seda, kui võrd täiusliku kogemuse kasutaja saab [3].

1.3 Meta Quest 3

Meta Quest tootesarjast omab Meta Quest 3 märkimisväärset riistvaralist eeldust. Kui võrrelda seda Meta Quest 2 ja Meta Quest Pro seadmetega, siis Quest 3 omab võimsamat protsessorit, kõrgemat resolutsiooni ja sügavuse andurit, mille abil saab ümbruskonda kaardistada [8]. Erinevalt Quest Pro-st Quest 3 ei võimalda silmade ja näo jälgimist [8], mida kasutatakse enamasti näo liigutuste edastamiseks virtuaalsele tegelasele.

Segareaalsuse rakendamise poolest omab suuremat tähtsust ümbruskonna kuvamine ja võimalused sellega interakteeruda. Selle koha pealt on Meta Quest 3-l samuti suur eelis, tänu kaamera läbilaskevõimele [9]. Omades suuremat pikslite arvu, paremat resolutsiooni ja dünaamilist valguse korrigeerimist väljastab Quest 3 palju paremat ja detailsemat pilti võrreldes oma eelkäijatega [9]. Lisaks sellele eelmainitud sügavuse andur suudab kaardistada automaatselt ruumi, sealhulgas ka mööblit [9]. Võrreldes Quest 2 ja Quest Pro versioone, kus kaardistamine toimub käsitsi [9], siis Quest 3 loob palju loomulikuma ja mugavama kogemuse kasutaja jaoks. Kõik need omadused teevad Quest 3 parimaks seadmeks Meta Quest tootesarjast segareaalsuse rakenduste arendamiseks.

1.4 Unreal Engine 5

Unreal Engine 5 on Epic Games poolt C++ keeles arendatud mängumootor, mis toetab mitut platvormi ja on üks populaarsematest mängumootoritest [10]. Unreal Engine 5 pakub

arendajatele võimalust kasutada arenduses C++ keelt või selle asemel samat loogikat implementeerida kasutades visuaalset skriptimise võimalust läbi Blueprint süsteemi [10].

1.5 Unity

Unity on Unity Technology poolt arendatud mängumootor, mis toetab mitut erinevat platvormi [11]. Mängumootor kasutab skriptimiseks C# programmeerimise keelt ning seda peetakse lihtsaks arusaamises [11].

2. Metoodika

Meta Quest 3 segarealsuse võimaluste ja piirangute väljaselgitamiseks otsustati autori ja juhendaja poolt uurida olemasolevaid lahendusi ja näiteid. Olukorra paremaks mõistmiseks lisaks uurimisele katsetatakse luua näidete põhjal erinevaid segarealsuse efekte. Sellise lähenemise käigus saavutatakse täiusliku arusaamist sellest, kuidas Meta Quest 3 rakenduste arendamise protsess näeb välja, missugused probleemid esinevad ning kuidas võimalused ja piirangud erinevad kasutatavast arenduskeskkonnast.

Segarealsuse olemusest tuleneb omadus, mille järgi peab virtuaalne maailm mõjutama reaalselt maailma realses ajas [6]. Selle kirjelduse alla sobivad väga hästi ümbruskonda muutvad efektid, mis kuvavad kasutajale tuttavat keskkonda päriselust, aga töödeldud või muudetud kujul ning seda kõike samal ajal. See tähendab, et lisaks virtuaalse sisu loomisele on tähtis ka seadme ligipääs ja võimalused reaalse keskkonna kasutamises.

2.1 Keskkonna seadistamine

Meta Quest 3 seadme jaoks rakenduste tegemine vajab ettevalmistust nii seadme kui ka tarkvara poole pealt. Selleks, et tagada sujuvat tööd, peavad kasutatavad keskkonnad olema õigesti seadistatud ning vastama segarealsuse seadmele. Autor jätab välja keskkondadega seotud arvuti spetsifikatsioonide nõuded.

2.1.1 Meta Quest 3 seadistamine

Järgnev lõik põhineb Meta dokumentatsioonile [7] ja autori kommentaaridele. Meta Quest 3 seadme komplekti kuulub peaseade, kaks vastavat seadmele kontrolleriit ning USB-C kaabel laadimiseks ja arvutiga ühendamiseks. Kui peaseade on vajalik virtuaalse ja segarealse keskkonna sisenemiseks, siis kontrolleriid on vajalikud, et navigeeruda nii Meta Quest 3 süsteemi tarkvaras kui ka selleks, et rakenduste ajal nad täidaksid nendele määratud funktsiooni. Peale seadme esmast seadistamist on vajalik rakendada ka seadistusi arendamise jaoks. Nende hulka kuuluvad: arendamise režiim, mis lubab ühendada seadet arvutiga ja pääseda ligi arenduskeskkonnadele, Meta Quest 3 ühendamine Meta Quest Developer

Hub-iga, mis võimaldab kontrollida seadet arvutist ning valminud rakendusi installeerida ja samuti tundmatute allikate rakenduste lubamine, et mitte ametlikku tarkvara käivitada.

2.1.2 Unreal Engine 5

Enne Unreal Engine 5 kasutamist on vaja seda alla laadida. Installeerimiseks on vajalik, et arvuti vastaks miinimum tarkvara ja riistvara nõuetele [12]. Meta dokumentatsioon nõuab samuti Meta arendaja kontot (Meta Developer Account) ja selle verifitseerimist [13].

Unreal Engine 5 kasutamisel on Meta poolt antud ka mitu soovitus: kasutada Meta ametlikku GitHub repositooriumi ja laadida alla eelseadistatud projekti või installeerida Unreal Engine Epic Games-i keskkonnast [14].

Unreal Engine 5 seadistamise juhendis on toodud välja, et Meta poolt seadistatud projekt omab juba vajalikke seadistatud pluginaid ning lisaparandusi [14]. GitHubi projektis on võimalik valida Unreal Engine versioon koos Oculus Platform SDK versiooniga [14]. Oculus Platform SDK võimaldab lisada erinevaid Meta Horizon OS funktsioone projekti [15]. Selleks, et saada ligipääsu Meta GitHub repositooriumile on vajalik eelmainitud Meta Developer Account kinnitamine ja Epic Games konto ühendamine GitHubiga [14].

Unreal Engine 5 allalaadimisel Epic Games keskkonnast on vajalik installeerida Epic Games Launcher, luua kasutaja ja installeerida vajalik Unreal Engine versioon [14]. Meta tarkvara arendamiseks on vajalik installeerida lisaks Meta XR plugin [14].

Meta XR plugin on vajalik Meta Horizon tarkvara arendamiseks Unreal Engine jaoks ning selle allalaadimine peab toimuma läbi Meta dokumentatsiooni, kus saab valida versioon ja paigaldada see käsitsi mängumootori vastavasse kausta [16].

Autor kasutas Unreal Engine allalaadimist Epic Games Launcherist, laadides alla versioon Unreal Engine 5.3.2. Allalaaditud Meta XR plugin versiooniks valiti 65.0. Lisaks laaditi alla Unreal Engine 5 Platform SDK ja Meta XR Simulator ning paigaldati samal viisil nagu Meta XR. Nende pluginate allalaadimine pole kohustuslik, aga annab ligipääsu abistavale ja lisafunktsionaalsusele.

Esmase kasutamise korral võib tekkida vajadus installeerida ja seadistada Android SDK, Android NDK ja JDK. Autor laadis alla Java SE Development Kit 17.0.10. Android Studio Flamingo sees installeeriti SDK Platform seadete all Android API 34 ja Android 12L (Sv2), SDK Tools tööriistade Android SDK Build-Tools 35-rc2 seadete all 34.0.0, 33.0.1 ja NDK seadete all 25.1.8937393. Android SDK Command-line Tools versiooniks valiti 11.0, CMake versiooniks 3.10.2 ning lisaks laaditi alla Android Emulator, Android Emulator hypervisor driver ja Android SDK Platform-Tools. Peale seda kustutati JAVA_HOME muutuja. Järgmisena on vajalik määrata vastavalt SDK, NDK ja JDK Unreal Engine seadetes. Autor proovis seadistada arenduskeskkond täiesti algusest, et saada paremat arusaama Unreal Engine ja pluginate seadete tööst.

2.1.3 Unity

Unity seadistamiseks on vajalik registreerida Unity ID, installeerida Unity Editor moodulitega Android Build Support, OpenJDK, Android SDK & NDK Tools ning lisaks laadida alla Meta Quest Link, mis võimaldab ühendada Meta Quest seadet arvutiga [17].

Erinevalt Unreal Engine 5 seadistamisest saab Meta XR pluginad lisada Unity Asset Store-ist ning laadida alla Unity projektis [17]. Meta XR seadistamiseks on vaja valida arendamise platvormiks Android, seadistada projekt XR tööriistade kasutamiseks ning installeerida kõik Meta dokumentatsiooni poolt vajalikud Meta XR pluginad, peale seda projekti seadistamise tööriist teeb kõik vajalikud seadistused ise [17].

Autor järgis eelmainitud samme ja seadistas tühja Unity projekti versioonil 6000.0.38f1. Peale seda oli võimalik pääseda ligi Meta XR tööriistadele ning kasutada nende sisseehitatud funktsioone, et lisada projekti peaseade objekti.

2.2 Dokumentatsioonid ja näited

Meta Quest 3 võimaluste ja piirangute uurimiseks otsustati autori ja juhendaja poolt uurida olemasolevaid näiteid ja dokumentatsioone, mis kirjeldavad sisseehitatud funktsioonide kasutust ja võimalusi.

2.2.1 Meta for Developers dokumentatsioonid (MetaHorizon)

Meta poolt on avaldatud dokumentatsioonid, mis kirjeldavad tööriistade sisseehitatud funktsioonide kasutusvõimalusi. Segareaalsuse võimaluste uurimise raames kasutab autor sellele teemale pühendatud dokumentatsioone.

Passthrough API laseb kuvada reaalsuse pilti Meta Quest 3 seadmele ja omab mitu erinevat funktsiooni, mis võimaldavad arendada paremat segareaalsuse kogemust [18]. Passthrough ligipääs on antud läbi Meta XR plugina ning selle kuvamine oleneb arendatava tarkvara vajadustest [19]. Üks viis, kuidas Passthrough kiht võib olla kuvatud on läbi keskkonna konstrueerimise funktsiooni, mis jätab seda pidevalt tööle [19]. Teine viis võimaldab lisada erinevatele geomeetrilistele objektidele passthrough kihi komponent eraldi, mis töötab nii kaua, kui komponent on olemas [19]. Lisaks sellele on võimalik pääseda ligi erinevatele kihi seadistustele, tänu millele on võimalik muuta kihi poolt kuvatavat reaalsuse pilti [19]. Seadistuste kasutamine annab võimalust luua erinevaid efekte, muutes värve, kontrastsust, heledust ja värvide kaardistamist [19]. Autori poolt sai Passthrough API funktsionaalsus katsetatud ning selle abil tehtud erinevad efektid, mida kirjeldatakse peatükis 2.3.1.

Depth API võimaldab reaalses ajas virtuaalsete objektide paremat integreerimist segareaalsusesse [20], kasutades Meta Quest 3 sügavuse andurit. Ilma Depth API kasutamist esinevad virtuaalsed objektid Passthrough reaalse maailma pildi pealmisel kihil ning ei anna täiuslikku segareaalsuse sukeldumise tunnet. Lisaks sellele kompenseerib sügavuse määramine keskkonna kaardistamise käigus kaardistamata jäetud elementide kaasamist segareaalsusesse [20].

Scene tööriist võimaldab kasutajatel liikuda ruumis ja tänu Scene Setup funktsioonile kaardistada ümber olevaid objekte ning salvestada Scene Modelina [21]. Tööriista funktsionaalsust ei käsitleta selles töös, aga selle mainimine on osa segareaalsuse võimaluste käsitlemisest.

2.2.2 Unreal-PassthroughSample

Dokumentatsioonides [30] kirjeldatud võimaluste demonstreerimiseks on Meta arendajate poolt pandud kokku GitHubi näidisprojektide repositooriumid [28]. Sobivate näidiste valimisel lähtuti efektide uurimise ja loomise eesmärgist. Eelmainitud funktsioonide seast sobib eesmärgi täitmiseks kõige paremini Passthrough API funktsionaalsust hõlmav näidisprojekt.

Unreal-PassthroughSample on Unreal Engine 5 mängumootoril tehtud GitHubi näidisprojekt, mis on loodud Passthrough API funktsionaalsuse näitamiseks [21]. GitHubi repositooriumis on projekt kättesaadav erinevates Unreal Engine 5 ja Meta XR versioonides. Funktsionaalsused on jaotatud eraldi tasemeteks [21]. Järgmiselt teeb autor ülevaadet kättesaadavatest funktsioonidest ja võimalustest.

PTBackground ehk tagaplaani tase näitab võimalust kasutada reaalsuse pilti tagaplaani kihina ning kuvada pealiskihina virtuaalseid objekte [21]. Tase kasutab stereo layer shape võimalust kuvada passthrough kihti pealis- või aluskihina. Aluskihina käitub passthrough taustana ning võimaldab näha ja kasutada virtuaalseid objekte. Pealiskihina asub passthrough kiht virtuaalobjektide peal. See funktsioon annab algset võimalust implementeerida segareaalsust ühendades virtuaalseid objekte reaalse maailmaga.

PTLighting tase käsitleb võimalust pimendada fookusest väljas olevat ala ning helendada vastupidi fookuses ala [21]. Sellise tulemuse saavutamiseks on olemas kaks passthrough kihti, millest üks on pimendatud ja teine helendatud ning määratud geomeetrilisele objektile. Geomeetrilise objekti koha ja pinna peal on Passthrough kihi heledus kõrgem, mis loob valgustatud ala efekti. Funktsioon annab ettekujutust sellest, kuidas saab kasutada passthrough kihte omavahel ja samuti ühte võimalustest passthrough kihi parameetrite muutmisest.

PTOpacity näitab, kuidas on võimalik passthrough kihi läbipaistvust muuta [21]. Taseme sees on võimalik kontrolleri abil määrata passthrough kihi läbipaistvust skaalal nullist saja protsendini. Niisugune funktsionaalsus võib olla kasulik vajadusel luua ülekäiku virtuaalse ja reaalse maailma vahel.

PTStyles omab tööriista, mille abil saab erinevalt passthrough kihti stiliseerida [21]. Tasemes on antud ette kaks värvikaart. Nende baasil saab määrata edasi pildi kontrastsust, heledust ja posteriseerimist ning piirjoonte värve. See on üks peamisi funktsionaalsusi, mille abil saab reaalsuse pilti modifitseerida ja efekte luua.

PTColorMap tase näitab, kuidas on võimalik sarnaselt PTStyles tasemele määrata värvide efekte kasutades RGBA parameetreid värvikaardil [21]. Tegemist on alternatiivse lähenemisega, kuidas võib muuta reaalsuse pildi värve. Võrreldes eelmise lähenemisega on antud tasemel võimalik teha täpsemaid värviseadistusi.

PTColorLut tase näitab, kuidas on võimalik kasutada otsingutabeleid (inglise keeles *lookup table*), et modifitseerida passthrough pilti [21]. Tasemel on toodud välja kahte tüüpi otsingutabelit. Otsingutabeleid saavad arendajad luua ise ja määrata passthrough kihi allikaks. Lisaks sellele on võimalik määrata kaalu ehk kui palju otsingutabelid muudavad pilti.

PTColorScaleAndOffset tasemel on võimalik modifitseerida värve kasutades skaalat ja kõrvalekallet [21]. Lisaks teistele tasemetele antud tase demonstreerib samuti, kuidas on võimalik veel pilti modifitseerida kasutades eelmainitud parameetreid.

PTLayer_Placement tase näitab, kuidas on võimalik passthrough kihti kasutada alus- või pealiskihina ning kuidas sellega koos töötab sügavuse abifunktsioon [21]. Selline funktsionaalsus on kasulik virtuaalreaalsuse integreerimiseks reaalse pildiga. Kuvades passthrough kihti aluskihina esinevad virtuaalsed objektid päriselu pildi peal. Vastupidiselt, kui passthrough kiht esineb pealiskihina, siis virtuaalsed objektid kaetakse.

PTMaskedBrush ja PTMasking on kaks taset, mis näitavad, kuidas on võimalik kuvada passthrough kihti teatud ekraani asukohal [21]. Näiteks PTMaskedBrush näitab passthrough kihti kohal, kus kontrolloriga joonistatakse ja PTMasking näitab passthrough kihti kohal, kus asub kontroller [21]. Tasemed annavad arusaama, kuidas on võimalik virtuaalsete objektide taustal kuvada reaalsuse pilti. Virtuaalne maailm on pidevalt nähtav ning samal hetkel on võimalus kuvada teatud piirkonnas päriselu pilti, et tagada segareaalsuse tunnet.

PTSurfaceProjected ja PTProceduralMesh tasemed näitavad, kuidas on võimalik kuvada passthrough kihti mingi kindla geomeetrilise objekti peal. Nii on võimalik luua virtuaalset keskkonda, samas jättes alles akna reaalsesse maailma.

Kõik eelmainitud tasemed on tehtud Meta poolt, et anda ülevaadet Passthrough API võimalustest segareaalsuse rakenduste arendamisel. Autori poolt said need näited läbi proovitud ning nende seadistused ja koodid analüüsitud selleks, et nende abil saaks autor katsetada enda efektide loomist Unreal Engine 5 mängumootoris.

2.2.3 Unity-PassthroughCameraAPISamples

2025. aasta märtsis avaldati Meta poolt uut kaamera ligipääsu funktsiooni Passthrough Camera API. Kaamera ligipääs on tagatud läbi Android Camera2 API [22] ja pakub arendajatele palju uusi võimalusi rakenduste arendamiseks. Meta dokumentatsiooni poolt on mainitud, et kaamera pilti on võimalik näiteks kasutada masinõppe ja arvutinägemise mudelitega objektide tuvastamiseks, kirjeldamiseks ja keskkonna muutuste jälgimiseks [22]. Pildi töötlemine ja kasutus ei piirdu enam Meta tööriistade funktsionaalsusega, vaid võimaldab kasutada nüüd ka teisi tööriistu, mis nõuavad sisendiks pilti.

Selleks, et näidata, kuidas ligipääs kaamerale toimib ja missugust funktsionaalsust see pakub, on Meta arendajate poolt loodud GitHubi näidisprojekt Unity-PassthroughCameraApiSamples [29]. Projekt näitab, kuidas saab pääseda ligi kaamera andmetele läbi WebCamTexture klassi kasutades abistavaid klasse WebCamTextureManager ja PassthroughCameraUtils [23]. Lisaks ligipääsule on projektis viis näidist kaamera andmete kasutusest:

- CameraViewer näidis annab aimu võimalusest saada ligipääsu kaamerale ning kuvada eraldi kaamera pilti tekstuurina objektile;
- CameraToWorld näidis näitab, kuidas on võimalik kahemõõtmelist pilti paigaldada kolmemõõtmelisse keskkonda;
- BrightnessEstimation näidis tutvustab, kuidas saab pildi andmetelt hinnata heledust ja kasutada seda näiteks kirja valgustamisel;
- MultiObjectDetection näidis näitab kaamera andmete kasutamist eeltreenitud mudelitega. Näidises kasutati YOLO mudelit, mis identifitseeris 80 klassi;

- ShadeSample näidis tutvustab võimalust luua erinevaid efekte kasutades kaamera andmeid ja mängumootori võimalusi.

Lisaks näidistele on Meta näidisprojekti toodud välja ka hetkel esinevad kaamera ligipääsuga ja kasutusega piirangud. Selle lõputöö kirjutamise ajaks on võimalik korrigeerida saadava ligipääsu ainult ühele kaamerale korrigeerida ning kättesaadava pildi resolutsioon on samuti piiratud 1280x960 pikslini. Eeltreenitud mudelite puhul on soovitatud kasutada väiksemaid mudeleid, et vältida seadme jõudlusprobleeme.

2.3 Efektide loomine

Näidisprojektide uurimise ja testimise käigus loodi autori poolt erinevaid efekte. Efektide tegemise eesmärgiks on saada aru, kuidas on võimalik olemasolevat funktsionaalsust kasutada, mis võimalusi see pakub ja missugused piirangud esinevad. Selleks, et vältida probleeme pluginate versioonidega, kasutati varasemalt mainitud Meta näidisprojekte. Näidisprojektide põhjal loodi eraldi kaustadesse efekte, koos vajalike skriptide- ja alusfailidega. Tulemusena valmis autori poolt kümme efekti Unreal Engine 5 näidisprojekti baasil ning kolm efekti ja üks näidis eeltreenitud mudeli kasutamisest Unity näidisprojekti alusel. Valminud tulemused on üleslaaditud autori GitHub repositooriumi MixedRealityEffects (Lisa 1). Repositoorium võimaldab pääseda ligi mõlemale näidisprojektile. Lisaks on repositooriumis tehtud ülevaade ja lühike kirjeldus iga valminud näite kohta.

Järgmiselt kirjeldatakse iga mängumootori kohta valminud efekte ning nende loomisel kasutatud funktsionaalsust.

2.3.1 Unreal Engine 5

Unreal Engine 5 näidised on loodud Meta Unreal-PassthroughSample näidisprojekti põhjal [28]. Kasutati Unreal Engine 5.5.1 ja Meta XR v74 versiooni näidisprojekti. Kõik autori poolt kasutatud ja loodud failid on salvestatud projekti Content kausta alamkausta MixedReality_VladimirMakarencov (Lisa 1).

Lisaks efektidele on kaustas olemas autori poolt loodud Pawn klass VRPTPawn_VladimirMakarencov, mille abil seadistatakse kontrollid ja peaseade

töötamiseks mängumootori objektidega ja komponentidega. Loodud Pawn klass omab eraldi kaamera komponenti, mis vastab peaseade töö eest ja kaks kontrolleri komponenti, mis vastavad parema ja vasaku kontrolleri eest. Kaamera komponendile on lisaks määratud alamkomponendina OculusXRPassthroughLayer kiht, mis võimaldab näha Meta Quest 3 kaameratest reaalselt maailma. Reaalse maailma pildi andmete kasutamine ja salvestamine eraldi ei ole võimalik, mille pärast pildi töötlemine toimub läbi OculusXRPassthroughLayer kihi enda seadete ja parameetrite kaudu. Pawn klassi kontrolleri komponendid olid samuti seadistatud vastavalt kasutama Meta Quest 3 kontrolleri komponente. Kontrollerite sisendite töötlemiseks oli loodud eraldi sisendite kaardistamise klass ning seadistatud vastavalt projekti vajadustele.

Iga efekti jaoks loodi eraldi Unreal Engine tase. Igal tasemel on eraldi tehtud skript, mis pääseb ligi OculusXRPassthroughLayer kihi funktsionaalsusele. Järgmiselt loetletakse valminud efekte (Joonis 1) ja lühidalt kirjeldatakse nende tööpõhimõtet.

RedFlip LUT (Joonis 1A) tase kasutab värvide kaardistamise funktsionaalsust, kus värvide kaardistamise tüübiks määratakse Color LUT ja allikaks RedFlip tekstuuri. Tulemusena näeb kasutaja ainult punaseid värve ning ülejäänud värvid on nähtavad hallides toonides.

RedFlip LUT changing weights (Joonis 1B) tase kasutab Color LUT värvide kaardistamise tüüpi ja allikaks samuti RedFlip tekstuuri. Skript määrab juhuslikult väärtuse Color LUT kaalu parameetritele. Tulemusena on näha, kuidas Color LUT efekti mõju pildile muutub juhuslikult.

Different LUT effects (Joonis 1C) tase demonstreerib Color LUT värvide kaardistamise tüübi erinevate allikate tööd. Skripti abil määratakse järjest projektis olemas olevad allikad Color LUT kaardistamise allikaks. Tulemusena näeb kasutaja erinevaid LUT efekte.

Only black edges (Joonis 1D) tase näitab tänu servade renderdamise funktsionaalsusele musta värvi ääri. Lisaks on kasutatud värvide kaardistamise funktsionaalsust, mille abil reguleeriti kontrastsust, heledust ja küllastust nii, et pilt jääks valgeks. Kombineerituna annavad need kaks funktsionaalsust efekti, kus on näha ainult musta värvi servad.

Only white edges (Joonis 1E) tase kasutab servade renderdamist ja värvide kaardistamise funktsionaalsust sarnaselt eelmisele tasemele, aga parameetrid määratakse vastupidi. Tulemusena tekib efekt, kus pilt on must ja kasutaja näeb ainult valget värvi servi.

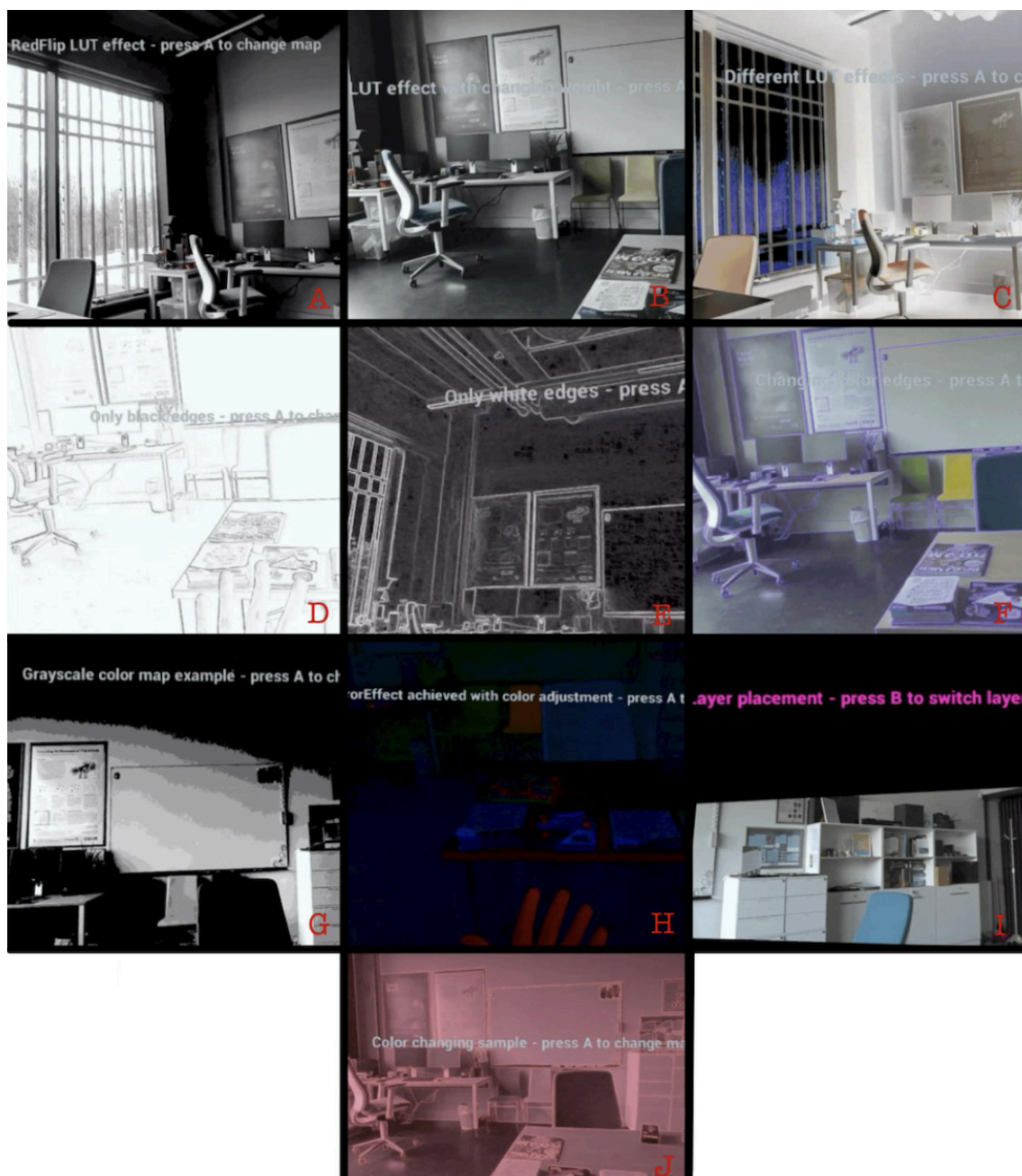
Changing color edges (Joonis 1F) tase kasutab servade renderdamise funktsionaalsust. Servade värv on määratud juhuslikult kasutades skripti. Teisi parameetreid ei muudeta, tänu millele kasutaja näeb pidevat äärte värvide vahetust.

Grayscale color mapping (Joonis 1G) tase näitab Grayscale To Color värvi kaardistamise tööd. Muudeti heleduse, kontrastsuse ja posteriseerimise parameetreid. Tulemusena näeb kasutaja hallides toonides pilti koos posteriseerimise efektiga.

Horror effect (Joonis 1H) tase kasutab värvide kaardistamise funktsiooni, kus reguleeritakse pildi heledust, kontrastsust ja küllastust. Erinevate väärtuste katsetamisel saadi pimedates ja punastes toonides efekti.

Layer placement (Joonis 1I) tase kasutab OculusXRPassthroughLayer kihi paigutuse funktsionaalsust. Selle funktsionaalsuse demonstreerimiseks sai loodud virtuaalne kast, milles kasutaja paikneb. Kontrolleri abil saab muuta Passthrough kihi paigutust alam- või ülemkihina. Alamkihina jäävad virtuaalsed objektid nähtavana, ülemkihina on virtuaalsed objektid peidetud.

Color changing (Joonis 1J) tase näitab värvi kaardistamise abil värvide muutmise funktsiooni kasutust. Skripti abil luuakse juhuslik värv ning seda kasutades muudetakse värvi kaardistamise värvi. Lisaks kasutatakse servade renderdamist ning antakse juhuslikult loodud värv parameetriks, mis muudab servade värvi. Tulemusena näeb kasutaja pildi ja servade värvide juhusliku muutumist.



Joonis 1. Unreal Engine 5 mängumootori alusel loodud efektid. RedFlip LUT (A), RedFlip LUT changing weights (B), Different LUT effects (C). Only black edges (D), Only white edges (E) ja Changing color edges (F). Grayscale color map (G), HorrorEffect (H), Layer placement (I) ja Color changing (J).

2.3.2 Unity

Unity näidised on loodud Meta Unity-PassthroughCameraApiSamples [29] näidisprojekti põhjal. Unity versiooniks kasutati 6000.0.38f1 ja Meta XR versiooniks v74. Autori poolt

kasutatud ja loodud failid on salvestatud Assets kausta VladimirMakarencovSamples alamkausta (Lisa 1).

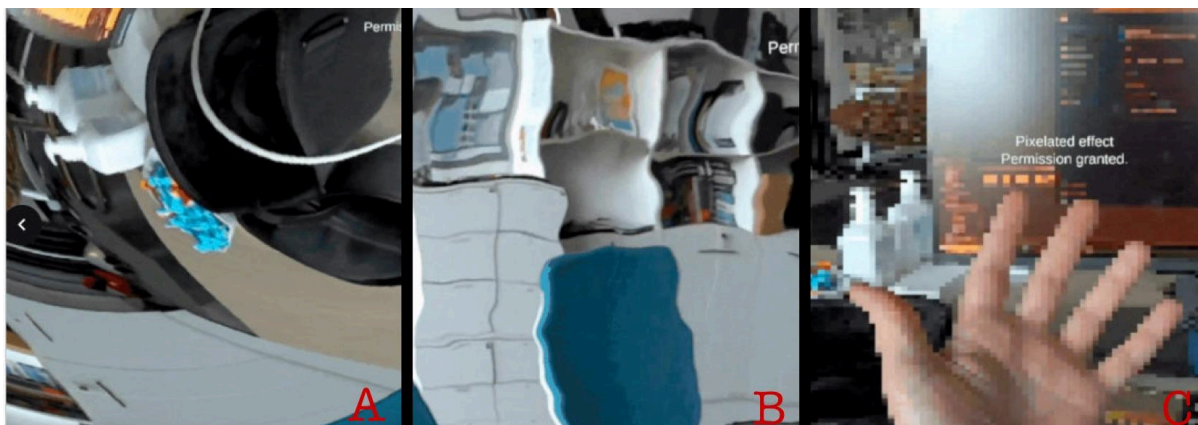
Iga valminud näidise jaoks on loodud eraldi Unity stseen, kus on seadistatud kõikide komponentide asukoht ja hierarhia. Näidised jagunevad efektideks ja eeltreenitud mudeli kasutamise näidiseks. Iga näidis kasutab ligipääsu Meta Quest 3 kaamerale ja selle andmetele. Kaamera andmete ligipääsuks kasutatakse varasemalt mainitud WebCamTexture klassi. Järgmisena loetletakse valminud näidiseid ja lühidalt kirjeldatakse nende tööpõhimõtet.

Efektide tegemisel kasutati Unity Shader Graph funktsionaalsust, mille abil loodi materjalid. Materjalid ühendati objektiga ning kasutades WebCamTexture klassi saadi Meta Quest 3 kaamerast pildi tekstuuri, mida määrati materjalide tekstuuriks. Tulemusena näeb kasutaja peaseadme kaamerast saadud pilti vastava materjali efektiga. Seda meetodit kasutades sai autori poolt loodud kolm efekti (Joonis 2).

Twirl efekt (Joonis 2A) paneb kaamera pilti keerlema, mis võib tekitada kasutajal enesetunde halvenemist. Selle tõttu efekt on seadistatud kujul, mis tekitab kasutajatel vähem probleeme enesetundega.

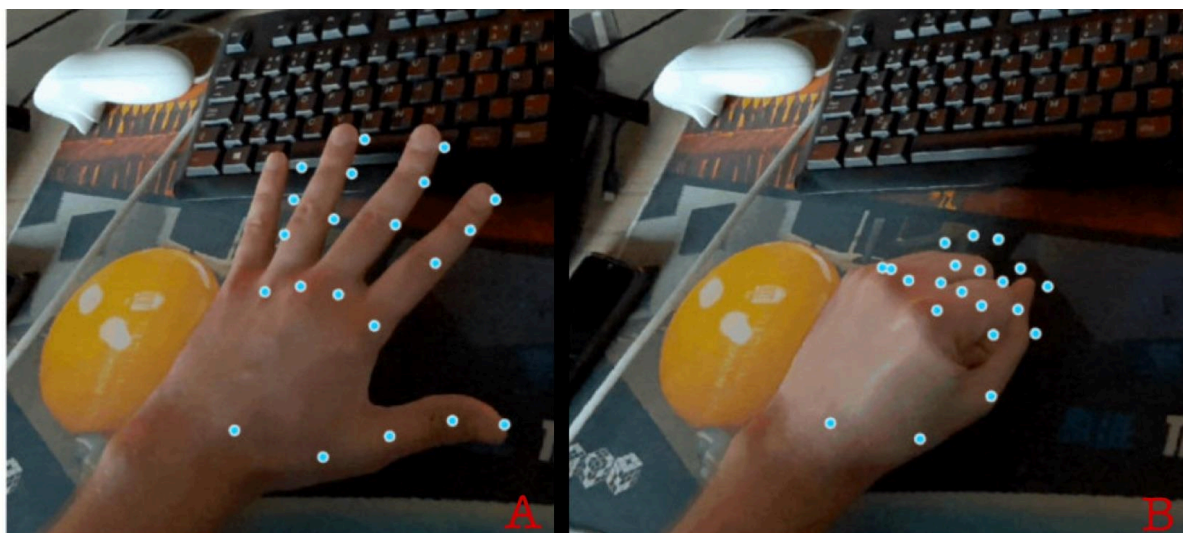
Waving efekt (Joonis 2B) paneb kaamera pilti lainetama. Lainetamise efekti on võimalik seadistada Unity Editoris, muutes lainetamise kiirust ja tugevust.

Pixelated efekt (Joonis 2C) muudab kaamera pilti pikseliseerituks. Pikseliseerumise tugevust saab muuta vastava stseeni materjali seadetes.



Joonis 2. Unity mängumootori alusel loodud efektid. Twirl efekt (A), Waving efekt (B) ja Pixelated efekt (C).

Eraldi oli loodud ka eeltreenitud mudelit kasutav näide Hand detection (Joonis 3). Näide põhineb Google Research poolt loodud vähekaalulisel käetuvastusmudelil BlazeHand [30]. Mudelit leiti Unity Sentiis näidisprojektist sentis-samples. Meta Quest 3 kaamera andmetega ühendamiseks kasutati BlazeHand näidisprojektis [30] olevaid prefabe, skripte ja teisi vajalikke andmeid. Kaamera tekstuuriga töötamiseks muudeti peamist skripti HandDetection.cs. Lisati WebCamTextureManageri klassi ning seostati skripti teised osad kasutama kaameralt saadud tekstuuri. Seadistati tuvastusmärkide ilmumist ja tööd, et nende paiknemine oleks korrektne kaamera pildil.



Joonis 3. Käetuvastuse mudel. Käetuvastuse mudel avatud käel (A) ja käetuvastuse mudel rusikas (B).

3. Analüüs

Selles peatükis analüüsitakse katsetatud näidiste ning loodud efektide põhjal Meta Quest 3 segareaalsuse rakenduste arendamise võimalusi ja piiranguid.

3.1 Piirangute analüüs

Uuritud Unreal Engine 5 mängumootoril tehtud Meta näidisprojekti kasutati Passthrough API funktsionaalsust. Kaamera töötlemata pildi andmete kättesaamine sellisel juhul ei ole võimalik ja töötlemine toimub ainult Meta arendajate poolt tehtud tööriistade kasutamise abil. Passthrough funktsionaalsuse kasutamise ajal ei ole kaamera andmed salvestatud ja edastatud [24]. Töö autori tehtud näidistes kasutati samuti Passthrough API funktsionaalsust, mida ühendati skriptidega, et efektide muutumine toimuks rakenduse töö käigus. Passthrough API pakub erinevaid võimalusi pildi modifitseerimiseks kuid nende mõju pildile jääb ühetüübiliseks. Autori poolt loodud efektid (Joonis 1) kasutasid Passthrough kihi töötlemiseks erinevat tüüpi parameetreid ja vaadates näidiste pilte on võimalik näha, et kokkuvõttes muutub enamasti ainult pildi värv. Võrreldes Unity mängumootori näidistega on Unreal Engine 5 projekti näidised piiratud Passthrough komponendi funktsionaalsuse ja parameetritega.

Varasemalt oli kasutajate poolt märgatud Passthrough pildi deformatsioone, mis tekitasid ebamugavusi kasutamise ajal [26]. Meta poolt üritati lahendada probleem Meta Quest v66 uuendusega [27]. Vaatamata pildi kvaliteedi parandamisele märkas autor lähedal asuvate objektide vaatamisel moonutusi ka uuematel versioonidel. Samuti teksti lugemine seadmete ekraanidelt on keeruline, mille pärast kaamera pildi kasutamine reaalse maailma täieliku asenduse tööriistana ei ole veel võimalik.

Ligipääs Passthrough Camera API-le on 2025. aasta mai seisuga võimalik ainult kasutades Unity mängumootorit. Passthrough Camera API on avalikustatud 2025. aasta märtsis ja lõputöö kirjutamise hetkel on seda regulaarselt uuendatud. Meta GitHubi näidisprojekti Unity-PassthroughCameraAPISamples [29] on toodud välja hetkel esinevad piirangud.

Ligipääs on võimalik korraga ainult ühele kaamerale, sellelt saadud tekstuur on limiteeritud 1280x960 resolutsioonini ja selle kuvamine toimub 40-60 ms viivitusega [23]. Piirangute

pärast pildi modifitseerimisel või efektide tegemisel ei ole võimalik saavutada täiuslikku sukeldumist segareaalsusesse ning tagada maksimaalselt mugavat kasutajakogemust.

Kaamera pildiga töötavate eeltreenitud mudelite kasutamine Meta Quest 3 peaseadmel võib põhjustada jõudlusprobleeme, mistõttu on Meta arendajate poolt soovitatud kasutada võimalikult väikest ja kiiret mudelit [23]. Autori poolt kokku pandud käe tuvastamise näites (Joonis 3) esinevad jõudlusprobleemid, mille pärast ei ole kaamera pildi ja mudeli töö niivõrd sujuv. Arendajatel on vaja arvestada rakenduse töö ja mudeli optimeerimisega.

Meta XR v76 versioonist alates peavad Passthrough Camera API-t kasutavad rakendused läbima läbivaatamise protsessi, et neid oleks võimalik avaldada [25]. Varasemalt oli kaamera ligipääs peetud avalikult eksperimentaalseks ja seda kasutavate rakenduste avaldamine ei olnud võimalik.

3.2 Võimaluste analüüs

Pikemat aega oli Meta Quest 3 segareaalsuse rakenduste arendajatel võimalik kasutada ainult pluginate funktsionaalsust. Passthrough Camera API avaldamisega tekkis arendajatel võimalus luua oma rakendused ja tööriistad kaameralt saadud pildi töötlemiseks ja kasutamiseks. Kaamera pilti on võimalik saada kätte mängumootori jaoks äratuntaval kujul, tänu millele on võimalik ka kasutada mängumootori erinevaid tööriistu tekstuuriga töötamiseks. Autori poolt loodud Unity efektid (Joonis 2) põhinevad mängumootori Shader Graph tööriistal ning nende kuvamine toimub läbi Shader Graphist loodud materjali.

Pilti on võimalik kasutada ka sisendina erinevates eeltreenitud mudelites. Sarnaselt Meta Unity-PassthroughCameraApiSamples projektis loodud objektide tuvastamise näitele on ka autori poolt valminud käe tuvastamise näide Hand detection (Joonis 3). Tulemusena töötab käetuvastuse mudel reaalses ajas kasutades Meta Quest 3 kaameralt saadud andmeid ning näitab tuvastusmärke tuvastatud käel. Käetuvastuse töös esineb varasemalt piirangutes mainitud jõudlusprobleeme, mida võib üritada lahendada optimeerimisega. Tuvastusmärgid peavad olema joontega ühendatud, aga autori näites joonte töö ei olnud täiesti korrektne ja sellepärast olid nad välja jäetud.

Sarnaselt käe- ja objektide tuvastusega on arendajate poolt kasutatud Passthrough Camera API, et töödelda mudelite abil kaamera pilti. Rolando Masís-Obando kasutaja poolt loodud LinkedIn sotsiaalmeedias postitus [31] näitab, kuidas on võimalik kasutada SDXL-Turbo ja LCM mudeleid Passthrough kaamera pildi modifitseerimiseks. Postituse autor on välja toonud ka esinenud piiranguid nagu väike resolutsioon ja töötlemise viivitus, mis kattuvad varasemalt käsitletud piirangutega.

Roberto Coviello poolt (kasutajanimi xrdevrob) on loodud GitHubi näidisprojekt QuestCameraKit [32]. Projekt demonstreerib samuti Passthrough Camera API kasutusvõimalusi. Näide OpenAI vision model saab kasutaja häälpäringut ja kaamera pilti sisendina, edastab seda mudelile ja tagastab ette loetud tekstina vastuse. Päring on saadetud interneti kaudu OpenAI mudelile kasutades API võtit. Näidis tutvustab võimalust kasutada kaamera pildiga suuremaid mudeleid interneti ühenduse kaudu.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli anda eestikeelne ülevaade hetkeseisundist segareaalsuse arendamise võimalustest ja piirangutest Meta Quest 3 virtuaalreaalsuse seadme alusel. Töös kirjeldati olemasolevaid viise Meta Quest 3 Passthrough kaamera pildile juurdepääsemiseks, nendel viisidel tehtud näidisprojekte ja viiside funktsionaalsuse kasutamist. Parema arusaamise jaoks katsetas autor kahte viisi efektide loomiseks, mille tulemusena valmis eraldi projekt. Kasutades Meta arendajate näidisprojekte koos autori poolt loodud näidistega analüüsiti Passthrough API ja Passthrough Camera API funktsionaalsust. Tulemusena selgitati Passthrough API ja Passthrough Camera API võimalusi, piiranguid ja erinevusi.

Selgus Passthrough API võimaluste ühetüübilisus, sõltuvus Meta arendajate poolt pakutud funktsionaalsusest ja kaamera pildi andmetele ligipääsu piirang. Passthrough Camera API võimaldab saada ligipääsu kaamera pildi andmetele, aga see toimub hetkel samuti mitme piiranguga. Selgitati, kuidas on võimalik kasutada saadud pildi andmeid mängumootori tööriistadega. Samuti kirjeldati eeltreenitud mudelite tööd kaamera pildiga ning esinevaid piiranguid, mille pärast võib kasutaja segareaalsuse kogemus olla kehv. Töö käigus valminud näidised annavad samuti visuaalse ülevaate võimalustest ja kinnitavad eelmainitud piiranguid.

Väljaselgitatud võimalused ja piirangud saavad olla kasutatud segareaalsuse rakenduste arendamisel sobivama funktsionaalsuse valimiseks ja parema olukorra mõistmiseks. Tulevikus oleks võimalik uurida Passthrough Camera API edasist arengut ning uusi võimalusi selle kasutamiseks.

Viidatud kirjandus

- [1] Cayir D., Acar A., Lazzeretti R., Angelini M., Conti M., Uluagac S. Augmenting Security and Privacy in the Virtual Realm: An Analysis of Extended Reality Devices. *Security and privacy in the metaverse*, 2024, 1. <https://arxiv.org/pdf/2402.03114> (14.12.2024)
- [2] Rokhsaritalemi S., Sadeghi-Niaraki A, Choi S.-M. A Review on Mixed Reality: Current Trends, Challenges and Prospects. *Applied Sciences*. 2020. <https://www.mdpi.com/2076-3417/10/2/636> (14.12.2024)
- [3] Snyder S. Mixed reality. *Salem Press Encyclopedia of Science*. 2023. <https://research-ebSCO-com.ezproxy.utlib.ut.ee/linkprocessor/plink?id=5d7ae968-f1d2-3ee5-a57a-a91135fcf133> (14.12.2024)
- [4] Che F. S. , Muliati S., Kasim H. M. Unveiling the potential of mixed reality: opportunities, challenges and future prospects. *Journal of Theoretical and Applied Information Technology*, 2023. https://www.researchgate.net/publication/375372936_UNVEILING_THE_POTENTIAL_OF_MIXED_REALITY_OPPORTUNITIES_CHALLENGES_AND_FUTURE_PROSPECTS (14.12.2024)
- [5] Wu X.. A review of virtual reality technology. *Applied and Computational Engineering*, 2024. https://www.researchgate.net/publication/378435611_A_review_of_virtual_reality_technology (14.12.2024)
- [6] CDW Corporation. Virtual reality vs. Augmented reality vs. Mixed reality. 2023. <https://www.cdw.com/content/cdw/en/articles/digitalworkspace/virtual-reality-vs-augmented-reality-vs-mixed-reality.html> (23.01.2025)
- [7] Meta. Configure your headset for app development. <https://developers.meta.com/horizon/documentation/unreal/unreal-quick-start-config-headset#unknown-sources> (19.04.2025)
- [8] Heaney D. Quest 3 Full Specs - Compared With Quest 2, Quest Pro, Pico 4 & Apple Vision Pro. 2023. <https://www.uploadvr.com/quest-3-specs/> (19.04.2025)
- [9] Heaney D. Meta Quest 3 Review: An Excellent VR Headset With Barely Passable Mixed Reality. 2023. <https://www.uploadvr.com/quest-3-review/> (19.04.2025)
- [10] Jokikokko V. H. The Potential of Unreal Engine 5 in Game Development: Exploring the Capabilities of the Unreal Engine. 2023. <https://www.theseus.fi/handle/10024/790326> (21.04.2025)

- [11] Singh S., Kaur A. Game Development using Unity Game Engine. *2022 3rd International Conference on Computing, Analytics and Networks (ICAN)*. 2022. <https://research-ebSCO-com.ezproxy.utlib.ut.ee/linkprocessor/plink?id=b7e32ba4-9a85-3012-82dd-b97ef021b29b> (23.04.2025)
- [12] Epic Games, Inc. Hardware and Software Specifications. https://dev.epicgames.com/documentation/en-us/unreal-engine/hardware-and-software-specifications-for-unreal-engine?application_version=5.0 (23.04.2025)
- [13] Meta. Prerequisites for installing Unreal Engine. <https://developers.meta.com/horizon/documentation/unreal/unreal-quick-start-install-prerequisites#create-dev-account> (23.04.2025)
- [14] Meta. Installing Unreal Engine. <https://developers.meta.com/horizon/documentation/unreal/unreal-quick-start-install-unreal-engine/> (23.04.2025)
- [15] Meta. Oculus Platform SDK. <https://developers.meta.com/horizon/downloads/package/oculus-platform-sdk/71.0> (23.04.2025)
- [16] Meta. Installing the Meta XR plugin. <https://developers.meta.com/horizon/documentation/unreal/unreal-quick-start-install-metaxr-plugin/> (23.04.2025)
- [17] Meta. Hello World. <https://developers.meta.com/horizon/documentation/unity/unity-tutorial-hello-vr>
- [18] Meta. Passthrough API Overview. <https://developers.meta.com/horizon/documentation/unreal/unreal-passthrough-overview>
- [19] Meta. Get Started with Passthrough. <https://developers.meta.com/horizon/documentation/unreal/unreal-passthrough-overview-gs>
- [20] Meta. Depth API Overview. <https://developers.meta.com/horizon/documentation/unreal/unreal-depthapi-overview/>
- [21] Meta. Passthrough Samples. <https://developers.meta.com/horizon/documentation/unreal/unreal-sf-passthrough>
- [22] Meta. Overview. <https://developers.meta.com/horizon/documentation/unity/unity-pca-overview> (29.04.2025)
- [23] Meta. Implementation using WebCamTexture. <https://developers.meta.com/horizon/documentation/unity/unity-pca-documentation> (30.04.2025)

- [24] Meta. How to Use VR Passthrough? 2024.
<https://forwork.meta.com/blog/what-is-vr-passthrough/> (06.05.2025)
- [25] Meta. Meta XR Core SDK (UPM). Version 76.0.
<https://developers.meta.com/horizon/downloads/package/meta-xr-core-sdk> (07.05.2025)
- [26] Hayden S. New Update Fixes One of Quest 3's Most Noticeable Issues. 2024.
<https://www.roadtovr.com/quest-3-passthrough-warping-fix-update-v66/> (08.05.2025)
- [27] Meta. Meta Quest v66 Update: Reduced Passthrough Distortion, Background Audio Support, and More. 2024.
<https://www.meta.com/en-gb/blog/meta-quest-v66-software-update-reduced-passthrough-distortion-background-audio/> (08.05.2025)
- [28] Oculus Samples. Unreal-PassthroughSample.
<https://github.com/oculus-samples/Unreal-PassthroughSample> (02.05.2025)
- [29] Oculus Samples. Unity-PassthroughCameraApiSamples.
<https://github.com/oculus-samples/Unity-PassthroughCameraApiSamples> (02.05.2025)
- [30] Unity Technologies. sentis-samples.
<https://github.com/Unity-Technologies/sentis-samples/tree/main/BlazeDetectionSample/Hand>
 (03.05.2025)
- [31] Masis-Obando R. Post on LinkedIn. 2025.
https://www.linkedin.com/posts/rmasiso_big-news-in-xr-last-week-meta-dropped-camera-activity-7307797312158789632-qMjn?utm_source=share&utm_medium=member_desktop&rcm=ACoAADvSLP0BtIEWi4ZP-prxF9VwN4xAUmsnz3A (9.05.2025)
- [32] Coviello R. QuestCameraKit. 2025.
<https://github.com/xrdevrob/QuestCameraKit?tab=readme-ov-file#license> (9.05.2025)

Lisad

Lisa 1. GitHub repositooriumi MixedRealityEffects

<https://github.com/VladimirM1/MixedRealityEffects/tree/main>

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Vladimir Makarenkov ,
(*autori nimi*)

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

Segareaalsuse seadmed: piirangud ja võimalused

(*lõputöö pealkiri*)

mille juhendaja on Madis Vasser ,
(*juhendaja nimi*)

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;

2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni;
3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile;
4. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Vladimir Makarenkov

09.05.2025