



TALLINNA TEHNIKAÜLIKOO
INSENERITEADUSKOND
Tartu kolledž

LIITREAALSUSE NAVIGATSIOONIRAKENDUSE LOOMINE PUIESTEE 80A BIM MUDELI PÕHJAL

DEVELOPMENT OF AN AUGMENTED REALITY NAVIGATION APPLICATION BASED ON PUIESTEE 80A BIM MODEL

RAKENDUSKÕRGHARIDUSTÖÖ

Üliõpilane: Lauri Lind

Üliõpilaskood 178519EDTR

Juhendaja: Rinaldo Rüütli, insener
Yevhen Bondarenko, doktorant-
nooremteadur

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

Jaanuar 2023.a

Autor: Lauri Lind

/ allkiri /

Töö vastab bakalaureusetöö/magistritööle esitatud nõuetele

"....." 20.....

Juhendaja:

/ allkiri /

Kaitsmisele lubatud

"....."20... .

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Lauri Lind

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

„LIITREAALSUSE NAVIGATSIOONIRAKENDUSE LOOMINE PUIESTEE 80a BIM MUDELI PÕHJAL“,

mille juhendajad on Rinaldo Rütli ja Yevhen Bondarenko

1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

_____ (kuupäev)

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Tartu kolledž

LÕPUTÖÖ ÜLESANNE

Üliõpilane: Lauri Lind 178519EDTR

Õppekava, peaariala: EDTR17/17 – Telemaatika ja arukad süsteemid

Juhendaja(d): Insener, Rinaldo Rütli

Doktorant-nooremteadur, Yevhen Bondarenko

Lõputöö teema:

(eesti keeles) Liitreaalsuse navigatsioonirakenduse loomine Puiestee 80a BIM mudeli põhjal

(inglise keeles) Development of an augmented reality navigation application based on Puiestee 80a BIM model

Lõputöö põhieesmärgid:

1. Luua liitreaalsuse rakendus Puiestee 80a hoones navigeerimise hõlbustamiseks.
2. Analüüsida rakenduse toimimist Puiestee 80a hoones ning selle kasutamise arusaadavust rakenduse potentsiaalse sihtgrupi jaoks.

Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Rakenduse keskkonna loomine	30.10.22
2.	Rakenduse funktsionaalsuse arendamine	30.11.22
3.	Projekti testimine	20.12.22

Töö keel: Eesti keel **Lõputöö esitamise tähtaeg:** 04. jaanuar 2023.a

Üliõpilane: Lauri Lind ".....".....20.....a

/allkiri/

Juhendaja: ".....".....20.....a

/allkiri/

Konsultant: ".....".....20.....a

/allkiri/

Programmijuht: ".....".....20.....a

/allkiri/

SISUKORD

SISUKORD	5
EESSÕNA	6
Lühendite ja tähiste loetelu	7
SISSEJUHMATUS	8
1. KIRJANDUSE ÜLEVAADE	9
1.1 Navigatsioonisüsteemid liitreaalsuse abil	10
1.2 Liitreaalsuse seadmed ning arendusplatvormid	14
1.2.1 Liitreaalsuse seadmed	14
1.2.2 Liitreaalsuse arendusplatvormid	15
2. RAKENDUSE LOOMINE	16
2.1 Metoodika ja töö keskkonna valik	16
2.1.1 Unity	16
2.1.3 Keskkonna valik	17
2.2 Rakenduse loomine	17
2.2.1 Puiestee 80a hoone mudel ja rakenduse keskkond	18
2.2.2 Rakendust toetava koodi kirjutamine	20
2.2.3 Kasutajaliidese loomine	21
3. RAKENDUSE ANALÜÜS JA EDASIARENDUSE VÕIMALUSED	24
3.1 Rakenduse kasutamine	24
3.2 Rakenduse testimine	25
3.3 Olemasolevate funktsioonide täiendamine	30
3.3.1 QR-koodi kuju analüüsimine	30
3.3.2 Klassiruumidesse navigeerimise funktsiooni lisamine	31
3.3.3 Rakenduse keskkonna laiendamine	31
3.4 Potentsiaalsed lisafunktsioonid	33
KOKKUVÕTE	34
SUMMARY	35
KASUTATUD KIRJANDUSE LOETELU	36
LISAD	Tõrge! Järjehoidjat pole määratletud.

EESSÕNA

Lõputöö teema sõnastati Tallinna Tehnikaülikooli inseneri Rinaldo Rüütli poolt. Teemaks oli liitreaalsuse rakenduse loomine, mis kasutab algandmeteks Puiestee 80a hoone BIM mudelit. Navigeerimisrakenduse loomise sõnastas töö autor ning rakenduse loomiseks vajalike meetodite kasutamist konsulteeris Tallinna Tehnikaülikooli doktorant-nooremteadur Yevhen Bondarenko. Töö praktiline osa valmis töö autori poolt.

Töö autor soovib tänada oma juhendajaid, Rinaldo Rüütli ning Yevhen Bondarenkot. Samuti soovib autor tänada innustuse ja abi eest sõpru ning elukaaslast.

Märksõnad: Liitreaalsus, Navigatsioonirakendus, Unity, ARCore, rakenduskõrgharidustöö.

Lühendite ja tähiste loetelu

BIM – mudelprojekteerimine: 3D infomudeli loomise protsess. (ingl k building information modeling)

LiDAR – laserkiiri kasutatav skaneerimisseade (ingl k light detection and ranging)

QR-kood – ruutkood, kahemõõtmeline mustvalge maatrikskood (ingl k Quick response)

iOS – operatsioonisüsteem Apple telefonidele

SISSEJUHATUS

Navigeerimine on ülesanne, mida tuleb enamik inimestel lahendada igapäevaselt. Enamasti saadakse sellega hakkama oma pead kasutades, kuid tihti läheb vaja ka abivahendeid. Kaarte ja teeviitasid on navigeerimiseks kasutatud juba aastasadu, tänapäeval on ilmunud ka kaasaegsemaid lahendusi inimeste sihtkohta juhendamiseks. Vabas õhus positsioneerimiseks kasutavad tänapäeval enamik seadmed satelliitnavigatsioonil põhinevaid, peamiselt GNSS, süsteeme. Siseruumidesse on satelliitide signaali puuduse või nõrkuse tõttu suunistamise süsteemide ülesseadmine keerulisem. Rahvarohketes keskustes kasutatakse tihti statsionaarseid majajuhte: nii interaktiivseid süsteeme kui ka tavalisi kaarte. Seoses nutitelefonide riistvaralise arenguga on avanenud ka võimalus juhendada inimesi läbi liitreaalsuse navigatsioonisüsteemide.

Virtuaal- ja liitreaalsus on kiiresti kasvavad valdkonnad. Eelkõige kasvab arendajate huvi laiendatud reaalsuse platvormidele rakendusi luua. Suurimaks takistuseks selle juures on senimaani olnud arendusvõimaluste vähesus ja laiendatud reaalsusele arendamiseks vajalike eriteadmiste vajalikkus. Mängumootorid nagu näiteks Unity ja Unreal Engine, mille kasutamine enda tarbeks on arendajatele tasuta, on suurendanud viimastel aastatel virtuaal- ja liitreaalsusele arendamise võimalusi ja oskusi [1]. Üks suur liitreaalsuse kasutusvaldkondi on navigeerimiserakendused. Liitreaalsuse navigatsioonirakenduse eelis tavalise kaardiga navigeerimise ees on kasutamismugavus. Rakenduse kasutaja ei pea end kahemõõtmeliselt kaardilt üles leidma ja kaardi suunasid päris maailmaga sobitama, vaid saab rakendust kasutades läbi liitreaalsuse näha päris maailma pindade ja objektide peale kuvatuid virtuaalseid suunaviitasid või muid juhiseid.

Antud lõputöö eesmärk oli luua baasrakendus, mis, rakendades liitreaalsuse võimalusi, aitab kasutajal navigeerida Puiestee 80a õppehoones. Kasutaja juhendamine toimub visuaalselt läbi telefoniekraani, kuvades päris maailma keskkonda virtuaalset teekonda kasutaja asukohast tema valitud sihtkohani. Rakendus on kättesaadav kõigile, kes omavad tänapäevast nutiseadet ning rakenduse kasutamiseks ei pea olema vajalik omada erialaseid teadmisi. Rakenduse põhiline sihtgrupp on kõik Tallinna Tehnikaülikooli Tartu Kolledži külalised ning üliõpilased, kes ei ole veel hoonega tuttavad. Lisaks on loodav rakendus baas, mida tulevikus edasi arendada, lisades navigeeritavat ala ja suurendades sihtkohtade loetelu või siis muutes positsioneerimise meetodit kasutajale mugavamaks kasutamiseks. Projekti loomiseks kasutatakse Unity mängumootorit, sellesse integreeritud ARCore ning ARFoundation liideseid ning AutoDesk 3ds Max tarkvara.

1. KIRJANDUSE ÜLEVAADE

Liitreaalsuseks (ingl. k. Augmented Reality) nimetatakse reaalse maailma ja virtuaalse maailma kombinatsiooni, kus päris maailma keskkonda kuvatakse reaalajas arvuti poolt genereeritud virtuaalseid objekte [2]. Üldiselt on liitreaalsuse rakenduste suurim väljakutse virtuaalsete objektide kuvamise täpsus ehk virtuaalse objekti kuvamine täpselt sinna, kuhu tulenevalt rakenduse eesmärgist seda kuvada on vaja. Reaalsuste märkamatu integratsiooni võti on seadme liitreaalsust toetav riist- ja tarkvara, mille eesmärk on võimalikult täpselt jälgida seadme asukoha ja orientatsiooni muutusi. Erinevus liitreaalsuse ja lihtsalt kaamerapildile objektide kuvamise vahel on see, et kui kasutaja seadet liigutab ja kaamerapilt muutub, jääb liitreaalsust kasutades mulje, et kuvatud virtuaalne objekt jääb päris maailmas samale kohale, kus see esialgu kuvatud oli. Muutuste jälgimiseks on tänapäeva seadmetes ühendatud GPS'i, güroskoobi, kiirendusmõõduri ja magnetomeetri riistvaralised lahendused ning kaamerapilti analüüsivad tarkvaralised lahendused [3]. Virtuaalse maailma sidumine päris maailmaga reaalajas on muutunud järjest populaarsemaks. Liitreaalsus on üks laiendatud reaalsuste kategooriatest (Joonis 1.1), olles lähemal reaalsusele kui segarealsus (ingl. k. Mixed Reality) ja liitvirtuaalsus (ingl. k. Augmented Virtuality) [4].



Joonis 1.1 Laiendatud reaalsuste skaala [5]

Liitreaalsuse populaarsusele andis suure tõuke, kui eelmises lõigus nimetatud riistvara ning liitreaalsusvõimekust toetav tarkvara muutus Android ja iOS telefonidel standardiks. Tänapäevaks on enamikel müüdavatel telefonidel kaasas liitreaalsuse kasutamise võimekus ning seda võimekust rakendatakse järjest enam. Kuigi liitreaalsuse populaarsus ühiskonnas algas Google otsingus kuvatavate liitreaalsuse loomade ning telefonimängudega nagu Pokémon GO, siis tänapäevaks kasutatakse liitreaalsuse lahendusi ka paljudes tõsisemates valdkondades. Kaitseväge õhu- ja maaväed üle maailma kasutavad liitreaalsust kiivrites, et kuvada sõdurile asjakohast infot ümbruse kohta. Meditsiinis kasutatakse liitreaalsust näiteks kirurgide poolt: kirurgiaõpilased harjutavad kontrollitud keskkonnas operatsioonide tegemist, kirurg

saab lisainfot patsiendi kohta läbi MRT või kompuutertomograafi kasutades liitreaalsust. Turismivaldkonnas eelistatakse järjest enam liitreaalsuse lahendusi, mis kuvavad turistile lisaks infote objektide kohta ka täiendavaid visuaale. Liitreaalsust kasutatakse ka tööstusvaldkonnas, käsilolevast objektist parema või täieliku vaate saavutamiseks. Navigeerimine liitreaalsusega näib naturaalse kasutusvaldkonnana liitreaalsusrakenduste seas. Kasutajale saab järgimiseks kuvada lihtsasti arusaadava teejuhise läbi seadmekraani. [6]

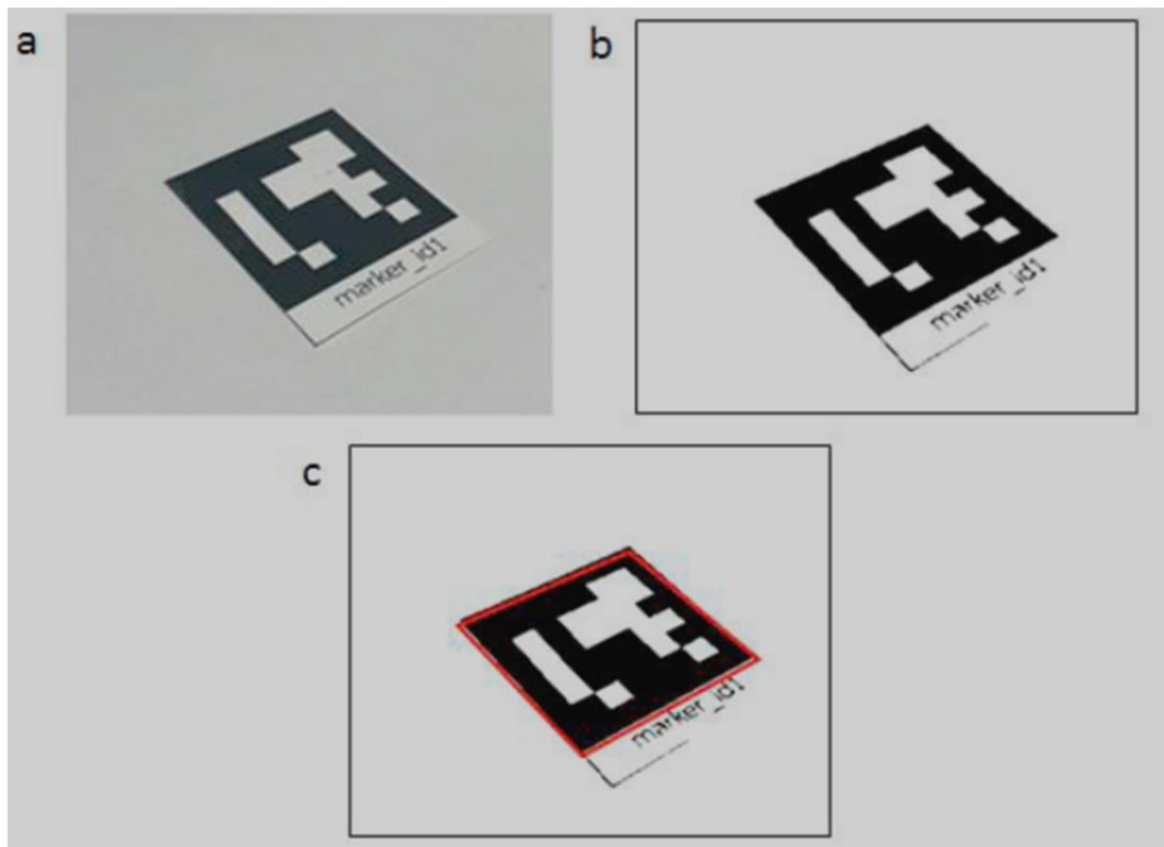
1.1 Navigatsioonisüsteemid liitreaalsuse abil

Kasutajal on liitreaalsust kasutava lahendusega lihtsam seostada navigeerimisjuhendeid päris maailmaga kui 2D navigeerimissüsteeme kasutades. Arusaadava kasutajaliidesega liitreaalsuse navigatsioonirakendusega ei pea kasutaja välja nuputama, kus ta kaardi peal asub ega kus on põhjasuund. Kasutaja saab avada liitreaalsuse rakenduse, läbida vajadusel rakenduse positsioneerimismeetodi ning talle kuvatakse päris maailma objektide ning pindade peale juhised navigeerimiseks. Juhend navigeerimiseks on tihtipeale joon. Joon võib olla pidev kui ka katkendlik, näiteks nähtamatule joonele kuvatud nooled. Navigeerimisjoon kuvatakse keskkonda ja see algab kasutaja asukohast. Kui kasutaja hakkab keskkonnas liikuma, muutub ka joone kuvamise alguspunkt vastavalt kasutaja asukohale keskkonnas. Navigatsioonirakendustes kasutatakse tihti ka rakenduse keskkonna 3D mudeleid selleks, et simuleerida paremini päris maailma käitumist. Näiteks selleks, et kasutaja ei näeks joont, mis keerab seina taha. Sellise lahenduse eesmärk on lisada rakendusse realismi ja teha joone asukoht päris maailmas ja selle järgimine arusaadavamaks. [7]

Probleem liitreaalsuses navigeerimisega on kasutaja positsioneerimine siseruumis. Erinevalt hoonest väljaspool navigeerimisest, ei saa siseruumis efektiivselt kasutada GNSS-süsteeme, sest hoone struktuur blokeerib satelliidilt saadavat infot, mille põhjal seadme positsioneerimine toimub. Seetõttu on leitud alternatiivsed viisid, kuidas positsioneerida liitreaalsuse seadet siseruumis. Siseruumides kasutaja asukoha leidmiseks, on üldiselt rakendatud kolme laiemat meetodit: markereid, punktipilvede sobitamist ja majakaid, mis kommunikeerivad seadmega läbi andmeside. [7]

Kõige kergem viis kasutaja asukoha leidmiseks on lasta kasutajal skaneerida mõnda objekti ehk markerit, mis on andmebaasis asukohainfoga seostatud. Selleks võivad olla ruumidesse spetsiaalselt paigutatud objektid nagu näiteks QR-koodid. Võib ka kasutada juba hoones olemasolevaid kaamerapildilt selgelt eristuvaid objekte nagu reklaamtahvlid, plakatid, eristuvad arhitektuurilised detailid. Suurim probleem

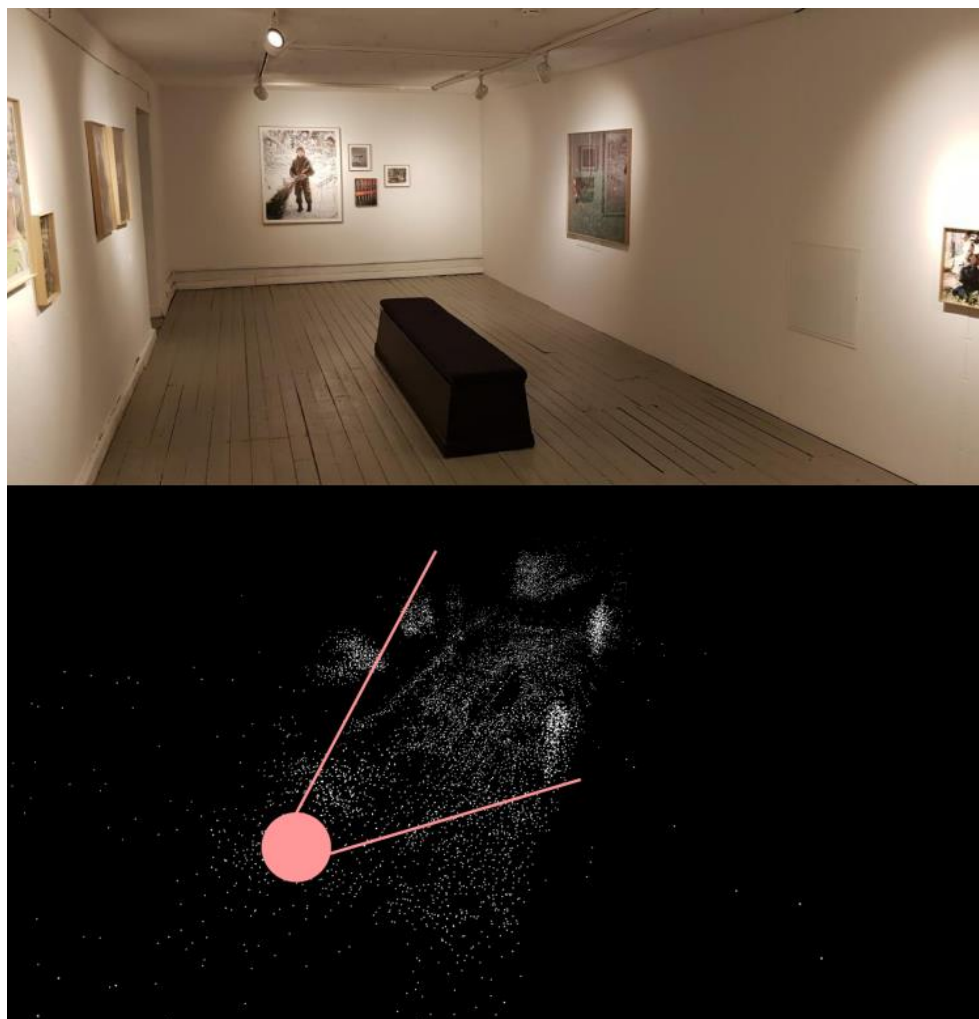
markeriga asukoha tuvastamisel on kasutaja seadme hoidmise viis. Rakendusel võib olla raskusi aru saada, mis nurga alt ja millisest suunast skaneeritakse markerit ja selle tulemusel võivad navigatsioonijuhised olla reaalsusega võrreldes nihkes või vale nurga all. Lahenduseks võib olla markeri analüüsimine: lihtsamal juhul kandilise markeri kuju ja suuruse analüüsi põhjal arvutada välja, mis suunast ja kui kaugelt markerit skaneeritakse (joonis 1.2). [8]



Joonis 1.2 QR-koodi kuju analüüsimine [8]

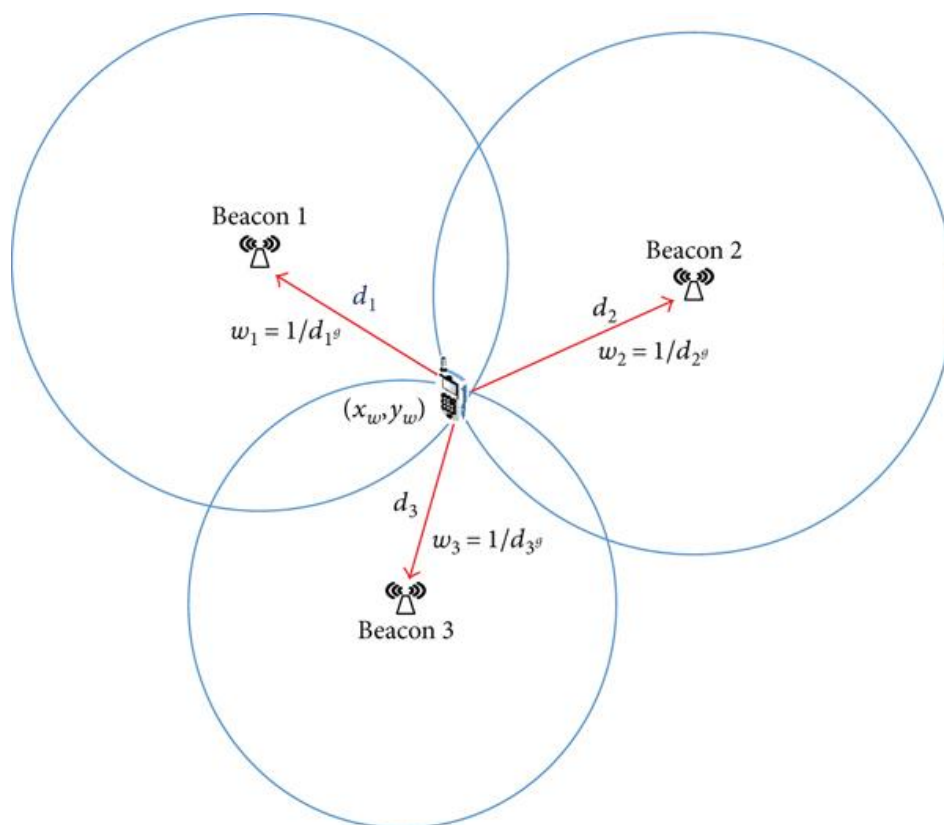
Kasutades punktipilvede sobitamise meetodit, puudub kasutajal vajadus otsida ruumis objekte skaneerimiseks. Seadmel on kättesaadaval keskkonnast varasemalt moodustatud punktipilv ja selle läbi protseduuriliselt kaamerapildist või LiDAR seadmega tehtava punktipilvega võrdlemise, sobitatakse kasutaja keskkonnas tema asukohale vastavasse kohta. Sobiva seadme ning detailsete algandmetega on punktipilvede sobitamise meetod kõige täpsem ning mugavam meetod kasutajale. Miinusteks on punktipilve loomise keerukus: nii algandmete jaoks mudeli koostamine kui ka protseduuriline punktipilve loomine. Täpsemaid punktipilvi luuakse LiDAR seadmega, kuid enamikel tänapäeva turul olevatel nutiseadmetel LiDAR võimekus puudub.

Punktipilve saab luua ka ilma LiDAR seadmeta. Selleks on loodud algoritme, mis tuvastavad objekte ja pindu läbi seadme kaamerapildi ja püüavad saadud kaamerapildist luua punktipilve, kuid tulenevalt kaamerapildi kahemõõtmelisusest esineb loodud punktipilves ebatäpsusi ja müra. Samuti on sellise meetodiga protseduuriline punktipilve loomine on nutitelefonil jaoks väga ressursimahukas. Joonisel 1.3 on näide sellise algoritmi tööst Pallase kunstisaalis. Samuti nõuab punktipilvede sobitamine palju arvutusjõudlust, mis vähendab programmi töötamiskiirust. Punktipilve meetodil on suur potentsiaal olla tulevikus, kui nutitelefonide arvutusjõudlus ning riistvaraline võimekus kasvab, põhiline positsioneerimise meetod siseruumides [9]



Joonis 1.3 Siseruum ja sellest protseduuriliselt loodud punktipilv:
Antud punktipilvel on suur müra põrandast punktipilve luues, näha on seintel asuvaid objekte ning ruumi keskel asuvat pinki [9]

Ka majakaid kasutades ei pea kasutaja enda seadme positsioneerimiseks tegema muud, kui rakenduse aktiveerima. Majakaks kasutatakse liitreaalsuse seadmega läbi mõne andmesideviisi suhtlevat seadet. Tüüpiliselt kasutatakse selleks madala energiakuluga Bluetooth seadmeid. Rakendus vajab kasutaja positsioneerimiseks vähemalt kahte majakat, tüüpiliselt kasutatakse vähemalt kolme. Rakendus saab positsioneerimise sisendiks kaugused majakatest, mille abil arvutatakse välja seadme asukoht keskkonnas nagu näidatud joonisel 1.4. Kuigi sellise meetodiga, ainult majakaid kasutades, saab väga täpselt teada seadme asukoha, pole võimalik selle meetodiga määrata seadme suunda. Seadme suund on vajalik, kui tegemist on liitreaalsuse navigatsioonisüsteemiga, et kuvada kasutajale navigatsioonijuhiseid korrektselt. Seetõttu tuleb majakate meetod kombineerida mõne teise lahendusega, kui rakenduses läheb vaja seadme suunda. [10]

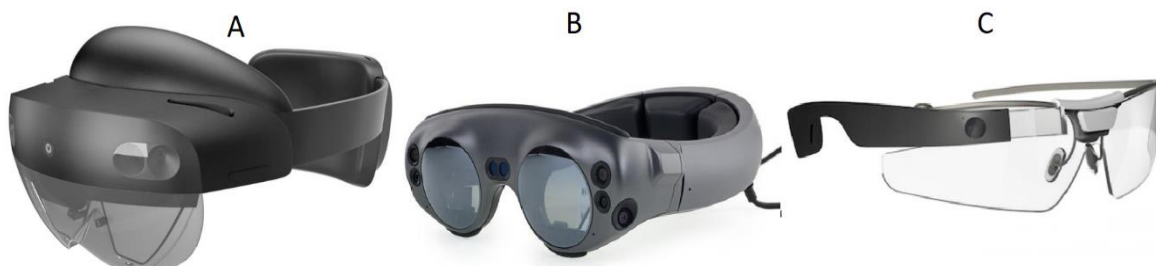


Joonis 1.4 Majakatel baseeruv positsioneerimine:
 Joonisel on 3 majakat Beacon 1, Beacon 2 ning Beacon 3, mis jagavad joonise keskel asuva telefoniga oma kaugusi telefonist d_1 , d_2 , ja d_3 , mille järgi telefon saab end positsioneerida. [11]

1.2 Liitreaalsuse seadmed ning arendusplatvormid

1.2.1 Liitreaalsuse seadmed

Liitreaalsuse kasutamiseks on laias laastus kaks valikut: Android ja iOS nutitelefonid ning erinevad liitreaalsust võimaldavad nutiprillid. Populaarsemad liitreaalsuse prillide platvormid aastal 2020 olid Microsoft HoloLens 2, Magic Leap ja Google Glass 2 (Joonis 1.5). Microsoft HoloLens 2 on kõrgema kvaliteedi liitreaalsuse seade ning võimaldab kasutaja silmade vaatesuuna jälgimist. HoloLens võimaldab kõrge kvaliteediga 3D objektide kuvamist ja nendega interakteerumist ning on mõeldud kasutama professionaalidele. Magic Leap sarnaneb HoloLens'iga, kuid on kergema kaaluga, mõeldud kasutamiseks nii professionaalidele kui ka kodukasutajatele. Magic Leap võimaldab virtuaalse keskkonnaga käeviibetega interakteerumist. Nii Magic Leap kui ka HoloLens 2 võimaldavad kasutajale 50 kraadist vaatenurka. Google Glass 2 on kaalult oluliselt kergem kui HoloLens 2 ja Magic Leap ning on mõeldud kasutamiseks kodukasutajale. Google Glass 2 on loodud juhiste, teavituste ja muu sarnase info kuvamiseks. [12]



Joonis 1.5 Populaarsed liitreaalsuse prillid:

A: Microsoft HoloLens 2, B: Magic Leap 1, C: Google Glass 2

Seadme valik sõltub paljuski arendatava rakenduse eesmärgist. Spetsiifilise tarkvara arendamine kindlale suunale nagu näiteks meditsiin või tööstus, on tõenäoliselt parem kasutada liitreaalsuse prille. Prillid võimaldavad suuremat mobiilsust tulenevalt kasutaja vabadest kätest ning liitreaalsuse objektide kuvamist 360 kraadi ümber kasutaja. See võimalus puudub nutitelefoni kasutades: objektid küll genereeritakse 360 kraadi ulatuses, kuid kasutaja näeb neid läbi telefoniekraani seda õigesse kohta suunates. Arendades tarkvara, mille eesmärk on olla kätte saadav laiemale publikule nagu näiteks turisminduses ja navigeerimiseks, tasuks arendusseadmeks valida nutitelefon, millele saab vajaliku tarkvara alla laadida iga sobiva seadme kasutaja. [13]

1.2.2 Liitreaalsuse arendusplatvormid

Nutitelefonile arendades on kolm tarkvaraplatvormi: ARCore, ARKit ning Vuforia, mis pakuvad arendajale tööriistu liitreaalsuslahendusi implementeerida. Kui ARCore ja Vuforia on mõeldud nii Android kui ka Windows seadmetele arendamiseks siis ARKit on loodud Apple poolt iOS tarkvarale. Need platvormid pakuvad kolme põhilist komponenti, mis toetavad liitreaalsuse arendamist: keskkonna tuvastus, keskkonna jälgimine ja objektide kuvamine keskkonda. Keskkonna tuvastamise peamine eesmärk rakenduses on aru saada, kus asuvad erinevad pinnad: põrandad, seinad, laed, mööblipinnad. Keskkonna jälgimise eesmärk on tuvastada läbi kaamerapildi ja seadme sensorite võimalikult täpselt kuhu seade keskkonnas liigub ja kuhu see suunatud on. Objektide kuvamine kasutab esimest kahte komponenti, et kuvada rakenduse eesmärgist sõltuvalt kasutajale reaalsesse keskkonda virtuaalseid objekte. [14]

2. RAKENDUSE LOOMINE

2.1 Metoodika ja töö keskkonna valik

Rakenduse seadmeks olid töö autoril kättesaadavateks seadmeteks liitreaalsuse prillid Microsoft HoloLens 2 ja Android nutitelefon. Kuna töö eesmärgist lähtuvalt peab rakendus olema kättesaadav kõigile üliõpilastele ning külalistele ja liitreaalsuse prillidele loodud rakendus on saadaval vaid vastavate prillide kasutajatele, otsustati rakendus luua Android operatsioonisüsteemiga nutitelefonile.

Nutitelefonidele liitreaalsuse rakenduste arendamiseks on turul domineerimas kaks mängumootorit: Unity ja Unreal Engine. Mõlemad mängumootorid toetavad nii ARCore, ARKit kui ka Vuforia platvorme ning nii Android, kui ka iOS süsteemidele arendamist. Populaarsem platvorm arendajate seas on Unity ning kuna ka töö autor sellega tuttavam oli, sai arendamiskeskonna valikuks Unity. [15]

2.1.1 Unity

Unity on mängumootor, loodud aastal 2005, mis võimaldab eri tüüpi rakendusi luua töö loomise ajal 25. eri platvormile. Unity on väga populaarne Android ja iOS rakenduse loojate hulgas: 2019 septembri seisuga oli 1000 kõige populaarsemast mobiilimängust 52% loodud Unityt kasutades. Unity pakub võimalusi arendamiseks nii 2D, 3D, liitreaalsus, segareaalsus kui ka virtuaalreaalsus rakendustele. Unity sobib väga ka algajatele, sest arendaja eest tehakse palju tööd ära 3D objektide ja keskkonna renderdamises. [16]

Unity pakub liitreaalsusele arendamiseks ka paketti nimega ARFoundation. ARFoundation võimaldab Unity's nii keskkonnas liitreaalsusega seotud objekte kasutada kui ka C# funktsioone kirjutades liitreaalsusega seotud klasse kasutada.

Arendades läbi Unity liitreaalsus-rakendust Android platvormile, on valikus kaks liitreaalsus tarkvaraplatvormi: ARCore ja Vuforia. Mõlemad on vabavaralised tarkvarad, kuid neil on erinevad eelised teineteise ees. Vuforia eelis ARCore ees on piltide, sümbolite ja pindade tuvastamine läbi kaamera, ARCore edestab Vuforia't keskkonna jälgimises ehk aru saamises kuhu kasutaja koos telefoniga liigub [14]. Arvestades, et navigeerimiskrakendus toetub tugevalt keskkonnas liikumisele ja virtuaalne keskkond peab olema võimalikult täpselt päris keskkonnaga sünkroonis, otsustati rakenduse loomisel ARCore platvormi kasuks.

2.1.3 Keskkonna valik

Rakenduse loomise keskkonnaks pakkus lõputöö juhendaja välja ühe TalTech Tartu kolledži hoone. 3D BIM Mudelid olid kättesaadaval nii Puiestee 80a hoonest kui ka Puiestee 78 ja 76 ehk vastavalt kolledži B ja C hoonetest. Tulenevalt hoone ehitusest, väljapääsude arvust ning mudeli failiformaadist, valiti lõputöö loomise keskkonnaks Puiestee 80a hoone. Hoonel on kolm väljapääsu, kolm ligipääsetavat korrust, kolm tualettruumi ning enamik ruumidest on ligipääsetavad kõigile õpilastele. Hoonel on ka kaks trepikoda, mis lisavad navigeerimisele ja sihtkohtadeni tee leidmisele keerukust.

2.2 Rakenduse arendamine Unity's

Rakendus loomiseks kasutati Unity versiooni 2021.3.12f1. Rakenduse testimiseks ning kasutamiseks on vajalik kaameraga Android nutitelefon, millele on paigaldatud operatsioonisüsteem Android 7.0 või uuem.

Enne rakenduse projekti loomist lisati Unity *Hub*'is eelnimetatud versioonile järgnevad rakenduse tööks vajalikud moodulid:

- *Microsoft Visual Studio Community 2019*
- *Android Build Support*
 - *OpenJDK*
 - *Android SDK & NDK Tools*
- *Universal Windows Platform Build Support*
- *WebGL Build Support*
- *Windows Build Support (IL2CPP)*
- *Windows Dedicated Server Build Support*

Projekti loomiseks kasutati Unity poolt pakutud 3D malli.

Selleks, et rakendus kasutaks töö autori seadmega sobivaid sätteid tuli seadetes teha järgnevad muudatused:

- *Player Settings > Android > Other Settings > Auto Graphics API > Disable*
- *Player Settings > Android > Other Settings > Graphics APIs > eemaldati Vulkan*
- *Player Settings > Android > Other Settings > Color Gamut > Dynamic Patching > Enable*
- *Player Settings > Android > Other Settings > Color Gamut > Minimum API Level > Android 7*

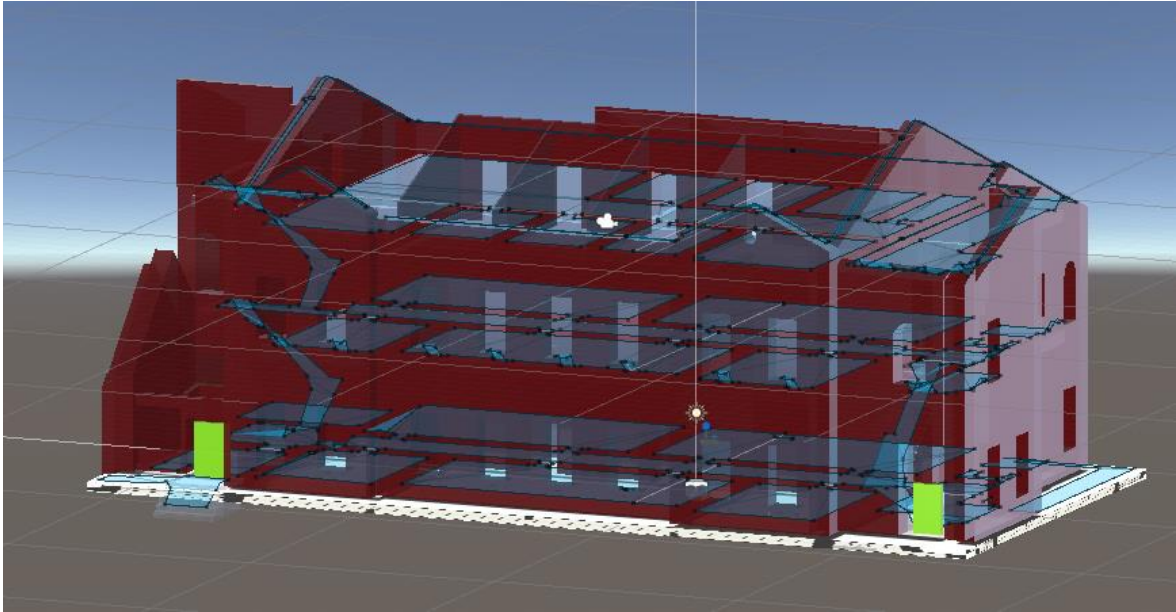
- *Player Settings > Android > Other Settings > Color Gamut > Scripting Backend > IL2CPP*
- *Player Settings > Android > Other Settings > Color Gamut > Target Architecture ARM64*
- *Build Settings > Platform > Android > Switch Platform*
- *Build Settings > Add Open Scenes*
- *XR Plugin Management > Install XR Plugin Management*
- *XR Plugin Management > Android > ARCore > Enable*

Unity *Registry* paketi haldurist lisati projektile AR Foundation pakett versiooniga 4.2.6. Seejärel kustutati stseenist antud rakenduse eesmärkideks ebavajalik objekt *main camera* ning lisati *AR Session Origin* ja *AR Session*.

2.2.1 Puiestee 80a hoone mudel ja rakenduse keskkond

Navigeerimiskrakenduse toimimiseks tuli luua navigeerimiseks sobilik keskkond. Vajalikud elemendid selleks on navigeerimise ala ning navigeerimise takistused, milleks selle töö kontekstis oli hoone korruste põrandad ning seinad. Lisaks oli rakenduse eesmärgiks vajalikud navigeerimise sihtkohtade objektid ning lähtepunktid.

Algandmetena oli töö autoril kasutada Puiestee 80a hoone 3D BIM mudel, mille failiformaat on *International Foundation Class* (.ifc) ning selleks, et töö Unity keskkonda importida, tuli fail ümber konverteerida *FilmBox* (.fbx) formaati. Selleks kasutati Autodesk 3ds Max 2023 tarkvara, kuhu oli lisatud Josef Wienerroither'i poolt loodud ja jagatud IfcMax plugin [17]. Seejärel imporditi hoone mudel Unity keskkonda, kus mudel pakiti lahti. Hoone detailid kategoriseeriti korrustele vastavalt. Kuna välisseinad ning esimese ja teise korruse siseseinad läbisid kõiki korruseid, lisati need eraldi kategooriasse. Korruste detailid jagati omakorda alakategooriatesse: korruse põrand, korruse siseseinad ning ülejäänud detailid. Rakendusele lisati *Window > AI > Navigation* komponent, mis arvutab välja valitud keskkonna põrandal navigeeritava ala, arvestades erinevaid takistusi mudelis (Joonis 2.1).



Joonis 2.1 Hoone keskkond visuaalse navigatsioonialaga:
Läbipaistev punane kolledži mudel, helesinine navigeerimisala, neonrohelist väljapääsu sihtkohaobjektid numbriga 1 ja 2.

Kõikide korruste põrandate objektidele lisati läbipaistev materjal selleks, et põrand ülalt-alla minikaardi kaamera vaatevälja ei blokeeriks. Seinte materjaliks määrati *VR/SpatialMapping/Occlusion Shader*, seega seina läbi liitreaalsuse kaamera rakenduses ei kuvata, kuid seinte taga asuvad objektid on kaamera eest varjatud. Seintele ja põrandale lisati vastavad sildid läbi mille määrati *AR Session -> AR Camera Culling Mask*'i seinad liitreaalsuse kaamerale nähtavaks ning põrand mitte.

Navigeerimise sihtpunktideks kasutati Unity *3D Object -> Cube* Objekte. Sihtpunktidele lisati silmatorkava värviga materjal ja need liigutati mudelis sihtpunktidele vastavatesse asukohtadesse. Rakenduse lähtekoha tähistamiseks loodi Unity *3D Object -> Sphere* Objekt nimega indikaator, millele lisati *Position Constraint* komponent, mille allikaks määrati *AR Camera* objekt, eesmärgiga siduda indikaatori objekt rakenduses liikuva ning kasutajat tähistava AR kaameraga. Indikaatori 3D objekti eesmärk oli kuvada rakenduse minikaardil rakenduse kasutaja asukohta. Indikaatorile lisati ka komponent *Line Renderer*, mille eesmärk on läbi liitreaalsuse kaamerapildi kuvada kasutajale joon kasutaja valitud sihtkoha objektini.

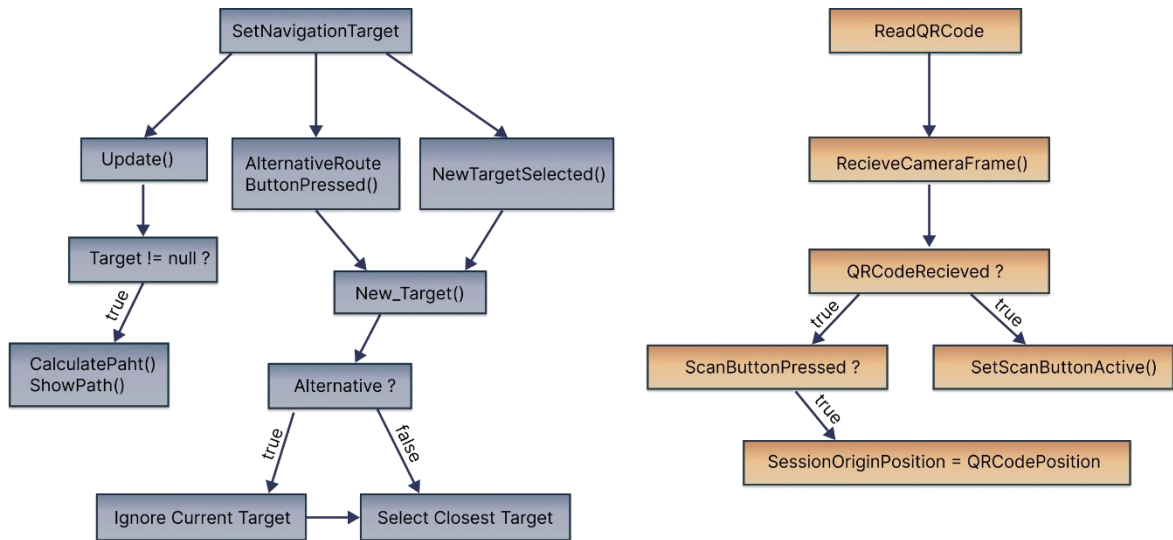
Sarnaselt sihtkoha objektidele lisati rakendusse QR-koodide asukohtade objektid. Selleks kasutati Unity *Create Empty GameObject* objekte. Objektid nimetati vastavalt nende asukohtadele korruste järgi ja liigutati mudelis asukohtadesse, kuhu päris hoones kleebiti seinale ka vastavad QR-koodid paberil. QR-koodide sisuks oli tekstiobjekt mis kattus selle QR-koodile vastava mudelis asuva QR-koodi objekti nimega.

2.2.2 Rakendust toetava koodi kirjutamine

Rakenduse toimimiseks kirjutati kahte olulisemat funktsiooni täitvad programmikoodid: `SetNavigationTarget` ning `ReadQRCode`. `SetNavigationTarget` funktsiooni eesmärk on kuvada kasutajale navigeerimisjoon ning valida õige sihtkoha objekt. `ReadQRCode` funktsiooni eesmärk on lugeda QR-kood ning muuta kasutaja virtuaalset asukohta keskkonnas. Funktsioonide põhiliste toimingute skeemid on kujutatud joonisel 2.2.

Funktsioon `SetNavigationTarget` vastutab vastavalt kasutaja sisendile õige sihtmärgi valimise ja selleni *Line Renderer* komponenti kasutades joone kuvamise eest. Valides soovitud sihtkohatüüpi - kas väljapääsu või tualeti, leiab funktsioon seda tüüpi sihtkoha objekti milleni on kasutajast lühim tee. Kui kasutaja soovib alternatiivset sama tüüpi sihtkohta, eemaldatakse kõige lähemal asuv sihtkoha objekt ja sama funktsioon korratakse. Kui rakenduses on joone kuvamine aktiivne, toimub funktsioonis ka joone pidev uuendamine. Kui kasutaja on muutnud oma asukohta, liigutatakse joone alguspunkt kasutaja uude asukohta ja joon kuvatakse uuesti. Uuenduse käigus kordagi sihtkoha objekti ei muudeta. Kasutaja asukoha muutust jälgib ARCore tarkvara.

Funktsioon `ReadQRCode` kasutab ZXing tarkvara [18], mis loeb kaamerakaadritl QR-koode ning muudab vastavalt saadud infole kasutaja virtuaalset asukohta hoone mudelis. Funktsioon saab sisendiks ka SCAN nupu oleku väärtuse. Selleks, et kaamera QR-koodile suunatuna lõpmatult ei skaneeriks, on funktsiooni toimimiseks vajalik SCAN nupu aktiivne olek. Iga projektis kasutatud QR-kood sisaldab tekstisõnumit. Tekstisõnum koosneb kahest osast: antud rakenduse otstarbeks eristatav silt „QRcodeRecenter“ ning QR-koodi asukohta iseloomustav number. Näiteks esimese korruse koridori lähtepunkti iseloomustav number on 104. Funktsioonis võrreldakse QR-koodis sisalduvat tekstisõnumit rakenduse keskkonnas olevate QR-kood objektidega, mis on nimetatud sama loogikaga nagu QR-koodid. Nimede kattumisel kirjutatakse kasutaja asukohta tähistava *sessionOrigin* koordinaadid üle QR-kood objekti asukoha koordinaatidega.



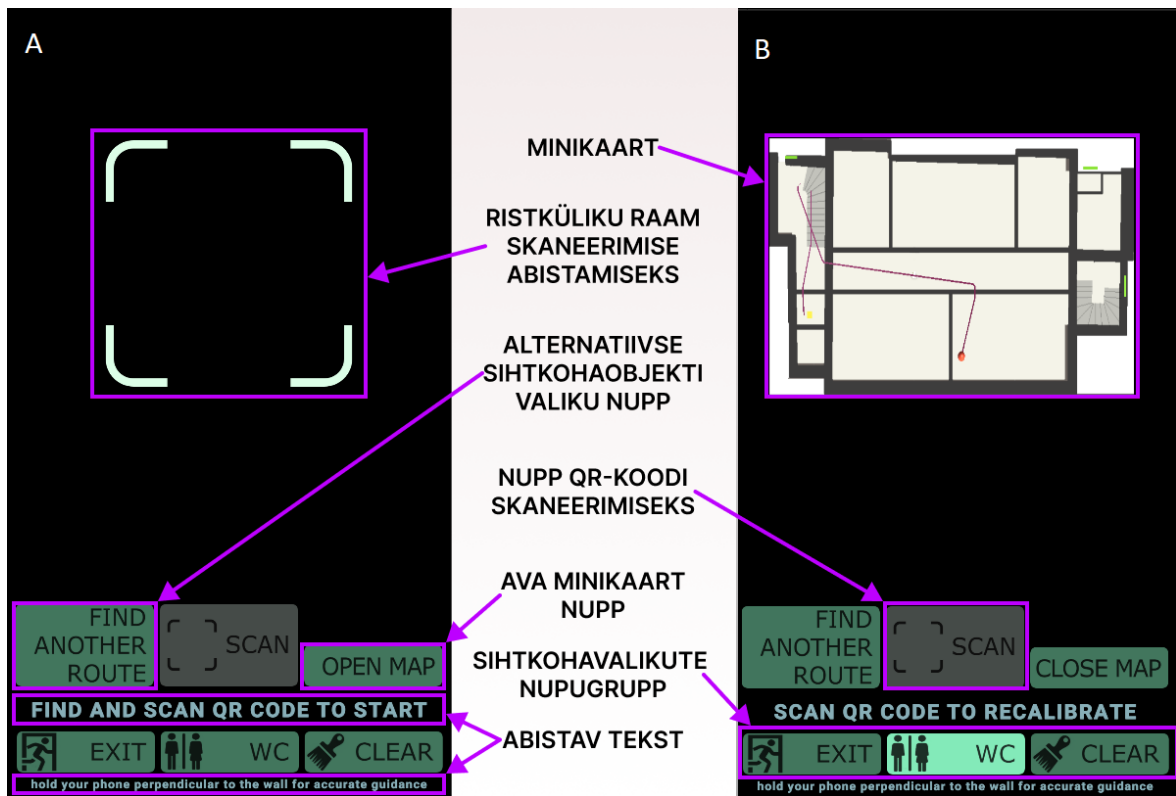
Joonis 2.2 Rakenduse funktsioonide skeem:
 () tähis näitab, et tegemist on funktsiooniga, ? näitab et tegemist on true/false konditsiooniga

2.2.3 Kasutajaliidese loomine

Kasutajaliidese loomiseks lisati stseeni *UI > Canvas* element, millele lisatud *UI* alamobjektid kuvatakse kasutaja telefoniekraanile. Kasutaja abistamiseks on ekraanile kuvatud kaks *TextMeshPro* tekstiobjekti (Joonis 2.3). Väiksem tekst ekraani allosas juhendab kasutajat skaneerima QR koodi seinaga risti selleks, et pärast kasutaja positsioneerimist ei oleks keskkond nihkes. Suurem tekst sihtkohaobjektide valikugrupi kohal juhendab rakendust esialgu avades kasutajal skaneerida QR-kood, et end positsioneerida. Kui kasutaja on esimese mudelis oleva QR-koodi skaneerinud, muudetakse suuremat tekstisõnumit. Kasutajat juhendatakse skaneerima uut QR-koodi, kui kasutaja soovib end uuesti positsioneerida. Antud tekstisõnumit rakenduse lahtioleku ajal rohkem ei muudeta. Lisaks juhendavatele tekstidele on rakenduse avades ekraanil ka ristküliku raam, kuhu sisse peaks kasutaja skaneeritava QR-koodi sihtima. Ristküliku raam aitab kasutajal skaneerida QR-koodi seinaga risti: selleks, et kandiline QR-kood sobiks ristküliku sisse, peab koodi skaneerima seinaga risti. Pärast seda, kui esimene mudelis olev QR-kood on skaneeritud, kaob ekraanilt ristküliku raam. Ristküliku raami all on samasuguse ristküliku raami sümboliga *SCAN* nupp, mis rakenduse avades on deaktiveeritud ja selle iseloomustamiseks halli värvi. Kui kasutaja telefoni kaamera kaadris on keskkonnas leiduv QR-kood, aktiveeritakse *SCAN* nupp ja selle iseloomustamiseks muutub nupp rohelist värvi. Kui kasutaja vajutab aktiivset *SCAN* nuppu, positsioneeritakse kasutaja rakenduse keskkonnas QR-koodile vastavasse asukohta ning kasutaja saab valida endale sobiva sihtkoha, kuhu navigeerimisjoon kuvada.

Ekraani alumisse serva lisati sihtkohavalikute nupugrupp (Joonis 2.3), kasutades *UI > Toggle* ehk lülitinupu valikut. Nupugrupis on kolm lülitinuppu, mis võimaldab igal nupul olla kahes seisundis. Nuppudeks on vastavalt järjestusele: *EXIT*, *WC*, *CLEAR*. Nupud *EXIT* ja *WC* võimaldavad kasutajal valida kahe sihtkohagrupi vahel, kuvades ekraanile *LineRenderer* joone lähima valitud sihtkohagrupi objektini. Nupp *CLEAR* puhastab eelmise valiku ning kaotab ekraanilt *LineRenderer* joone. Kõikidele nuppudele on lisatud funktsioon, mis antud nuppu vajutades deaktiveerib kõik teised nupud sihtkohavalikute nupugrupis. Kasutajamugavuse eesmärgil on lisatud igale sihtkohavaliku nupule ka nupu funktsiooni kirjeldav sümbol. Kui kasutaja soovib alternatiivset sihtkohaobjekti valitud sihtkohagrupist, saab kasutaja vajutada *FIND ANOTHER ROUTE* nuppu. Antud nupp muutub aktiivseks, kui kasutaja on juba valinud ühe sihtkohagrupi vahel ja navigatsioonijoon lähima sihtkohaobjektini on ekraanil kuvatud. Vajutades *FIND ANOTHER ROUTE* nuppu, kuvatakse kasutajale navigatsioonijoon kauguselt järgmise sihtkohaobjektini sama sihtkohagrupi valikute hulgas. Nuppu korduvalt vajutades ja sihtkohaobjektides kõige kaugema objektini jõudes, kuvatakse järgmise vajutusega taas kauguselt teine objekt. Selleks, et kuvada taas lähim sihtkohaobjekt, peab kasutaja vajutama taas vastavat sihtkohavaliku nuppu.

Lisaks on kasutajal võimalik enda asukohta hoones näha minikaardil. *SCAN* nupu kõrval (Joonis 2.3) asub nupp *OPEN MAP*, mida vajutades avaneb telefoni ekraanile Puiestee 80a hoonet kujutav majaplaan. Kui kasutaja on end QR-koodi skaneerides positsioneerinud, kuvatakse minikaardile ka kasutaja asukoht oranži täpina. Kui kasutaja on valinud ka sihtkoha, ja rakendus kuvab läbi kaamerapildi navigatsioonijoont, on navigatsioonijoon läbi terve hoone nähtav ka minikaardil.



Joonis 2.3 Rakenduse kasutajaliides ilma kaamerapildita:

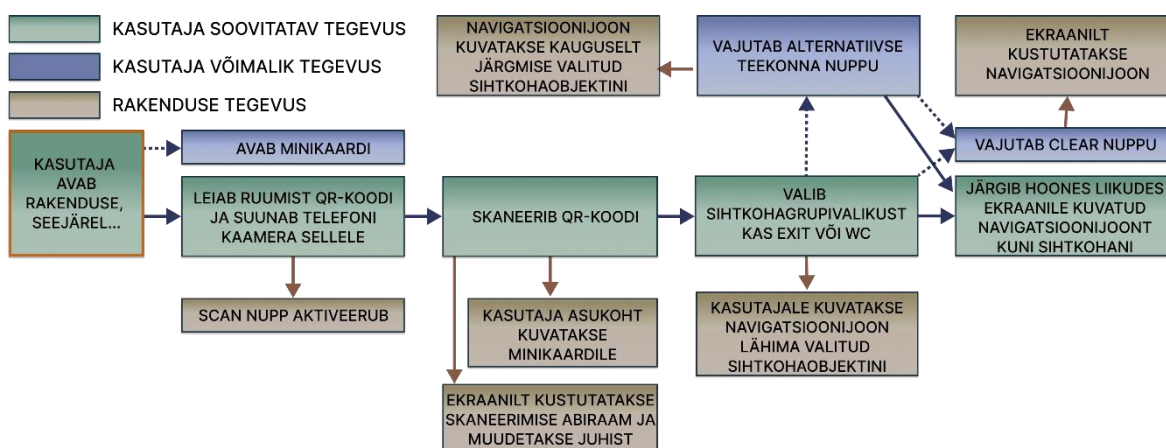
A: Kasutajaliides rakenduse avades, ükski valik pole selekteeritud ja ristküliku raam on veel

ekraanil. B: Kasutajaliides pärast QR-koodi skaneerimist, minikaart on avatud, WC sihtkohavalik on aktiivne ja minikaardil on näha navigeerimisjoont ning indikaatorobjekti.

3. RAKENDUSE KASUTAMINE JA TESTIMINE

3.1 Rakenduse kasutamine

Rakenduse kasutamine algab rakenduse installeerimisest ja selle avamisest. Rakenduses ei kuvata eraldi kasutusjuhust, selle asemel on lisatud abistavad kirjad ning rakendusele on loodud lihtne ja kasutajale arusaadav kasutajaliides. Rakendus on loodud eeldusega, et kasutaja on tuttav liitreaalsuse rakenduse toimimise põhimõtetega ning saab aru inglisekeelsest kasutajaliidesest. Rakendus toimib põhimõttel, et rakenduse põhifunktsiooni täitmiseks läbib kasutaja õiges järjekorras viis etappi, mis on toodud joonisel 3.1 rohelist värvi kastidega. Lisaks põhifunktsioonile, saab kasutaja ka valida ka lisafunktsioone, joonisel tähistatud sinist värvi kastidega, nagu näiteks minikaardi avamine ja sulgemine ning alternatiivsete teekondade kuvamine. Vastavalt erinevate funktsioonide kasutamise võimalikkusele, muutuvad funktsiooniga seotud nupud kasutajaliideses aktiivseks. Nupu aktiveerimine tähendab rakenduses selle nupu värvi erksamaks muutumist ja selle eesmärk on juhendada kasutajat soovitud tegevusi läbima. Lisaks nuppude aktiveerimisele toimuvad vastavalt kasutaja valikutele rakenduses ka muud muutused, joonisel tähistatud pruuni värvi kastidega, nagu näiteks ekraanile objektide kuvamine ja kustutamine. Rakenduse kasutamiseks on joonisel 3.2 pilt rakenduse kasutamise hetkest.



Joonis 3.1 Rakenduse kasutamislõogika vooskeem:

Vooskeemil on kujutatud jämedate nooltega vasakult paremale rakenduse optimaalseim kasutuskulg. Rakenduse kasutamine algab oranži äärisega plokist. Optimaalse kasutuse plokkidest üles- ja allapoole on toodud sündmused, mis kasutaja tegevustest sõltuvalt rakenduses toimuvad ja võimalikud lisategevused, mida kasutaja erinevates etappides teha saab.



Joonis 3.2 Pilt rakenduse kasutamisest Puiestee 80a teisel korrusel: Pildil on sihtkohaobjektiks valitud WC ning navigatsioonijoon juhatab kasutaja teise korruse tualetini.

3.2 Rakenduse testimine

Rakendust testiti õppehoones kokku kolmel korral: kahel korral rakenduse arendamise käigus ning ühel korral pärast rakenduse valmimist. Esimesel testimisel testis rakendust vaid töö autor, teisel testimisel testis rakendust lisaks ka töö juhendaja ning kolmandal testimisel testis rakendust lisaks töö autorile ka erialaväline testija. Lisaks kolmele Puiestee 80a hoones toimunud testimisele toimus ka pidev kasutajaliidese ning rakenduse funktsioonide testimine arendamise käigus väljaspool õppehoonet. Paljude rakenduse funktsioonide töötamist oli võimalik testida ka Unity mängumootori siseselt.

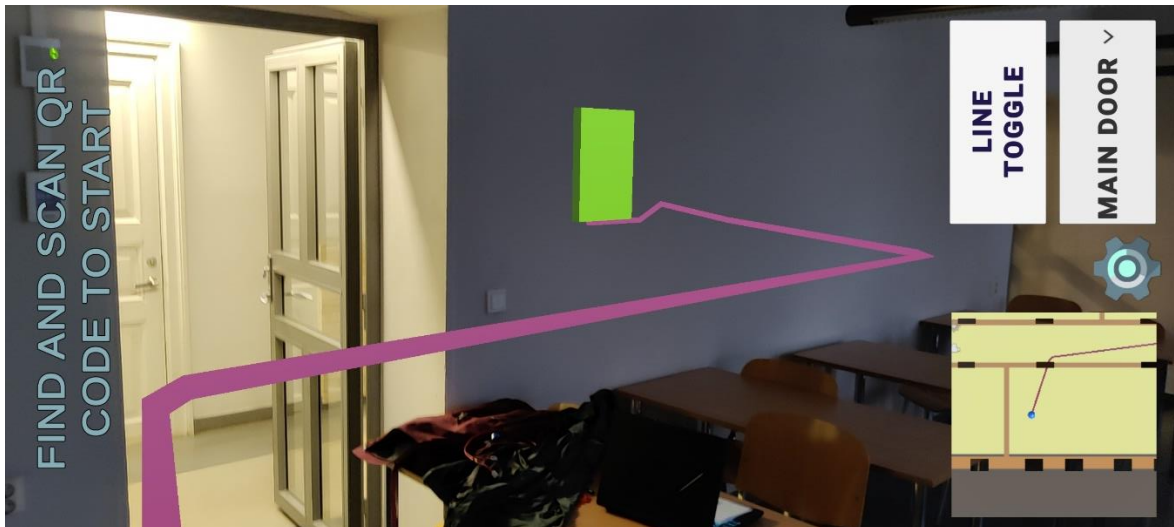
Esimene testimine õppehoones toimus rakenduse vähendatud funktsioonidega. See tähendas, et kasutajaliides ja funktsioonide toimimine polnud veel täielikult välja arendatud: kasutaja pidi ripploendist valima manuaalselt iga sihtkohaobjekti vahel, navigatsioonijoon ning sihtkohaobjektid olid nähtavad läbi seinte. Esimese testimise põhieesmärgid olid järgnevad:

- Tuvastada, kas virtuaalse keskkonna mõõtmel kattuvad õppehoone päriseluliste mõõtmeltega.
- Leida QR-koodidele optimaalsed asukohad õppehoones.
- Leida puudused funktsioonide töös, navigatsioonijoonte kuvamises, kasutajamugavuses ja QR-koodide skaneerimises.

Esimese testimine toimus vaid töö autori poolt. Testimise käigus asetati seintele ajutistele kohtadele QR-koodid ning katsetati navigeerimist väljapääsu sihtkohaobjektideni õppehoone esimese korruse auditooriumist, koridorist ning teise korruse koridorist. Joonisel 3.2 on kuvatõmmis ühest esimese testi katsest. Esimese testimise tulemused olid järgnevad:

- Tuvastati joone kuju, pikkust ja asukohta analüüsid, et virtuaalse keskkonna mõõtmel kattuvad õppehoone päriseluliste mõõtmeltega.
- Tuvastati QR-koodide optimaalne hulk ning asukohad õppehoones.
- Tuvastati probleem, kus rakendus jäi QR-koodi korduvalt skaneerima, kui telefon oli QR-koodi poole suunatud.

Esimese testimise tagajärjel lisati rakendusele SCAN nupp ja paigutati keskkonda vastavalt testimise tulemusele kümme QR-koodi. QR-koodid paigutati järgnevalt: 3 koodi hoone esimese korruse auditooriumidesse, 3 koodi teise korruse auditooriumidesse, üks kood mõlema korruse koridori keskele ning kaks koodi kolmanda korruse põhiruumi.



Joonis 3.3 Kuvatõmmis rakenduse esimesest testimisest Puiestee 80a hoones: Kasutaja seisab hoone esimese korruse auditooriumis, on skaneerinud QR-koodi ning on valinud sihtkohaobjektiks hoone peaukse. Navigatsioonijoon on kuvatud lilla värviga ning peauks on tähistatud rohelise risttahukaga.

Teine testimine õppehoones toimus samuti rakenduse vähendatud funktsioonidega. Testimist viis läbi töö autor ning testimisel osales ka töö juhendaja, kes oli rakenduse esmakordne kasutaja. Teise testimise põhieesmärgid olid järgnevad:

- Testida esimeses testimises välja tulnud puuduste paranduste toimimist.
- Testida rakenduse toimimise arusaadavust rakendusele võõra testija kaasamisega.
- Kontrollida QR-koodide asetuste täpsust.

Teise testimise protseduur jälgendas esimese testimise protseduuri, lisatud oli kaastestija. Teise testimise tulemused olid järgnevad:

- Tuvastati, et esimese testimise tulemuste põhjal lisatud SCAN nupp täitis eesmärgi ning skaneerimine toimus vaid nuppu vajutades.
- Tuvastati probleem, kus QR-koodi nurga all skaneerides paigutatakse kasutaja virtuaalses mudelis nurga alla ja sellest tulenevalt kuvatakse kasutajale objektid valesse kohta.

- Tuvastati probleem, kus teatud asukohtades (ebamäärase kontuuriga ja ebapiisava valgusega ruumides) rakenduse liitreaalsuse platvorm ARCore, ei suuda kasutaja liikumist täpselt tuvastada ning virtuaalne keskkond ei ühti enam päris maailma keskkonnaga.

Teise testimise tagajärjel lisati rakenduse kasutajaliidesele kasutajat abistav tekst, mis juhendas kasutajat skaneerima QR-koodi nii, et telefoni kaamera oleks suunatud seinaga risti.

Rakenduse kolmas testimine toimus töö autori läbiviimisel ning kaasatud oli erialaväline testija. Testijale ei antud rohkem infot, kui rakendus ise seda kuvab, eesmärgiga aru saada, kas rakenduse kasutamine on uuele kasutajale arusaadav. Testimine toimus lõpetatud rakendusega. Kasutaja sai valida sihtkohagruppide vahel ning valida ka alternatiivsete sihtkohaobjektide vahel. Kolmanda testimise põhieesmärgid olid järgmised:

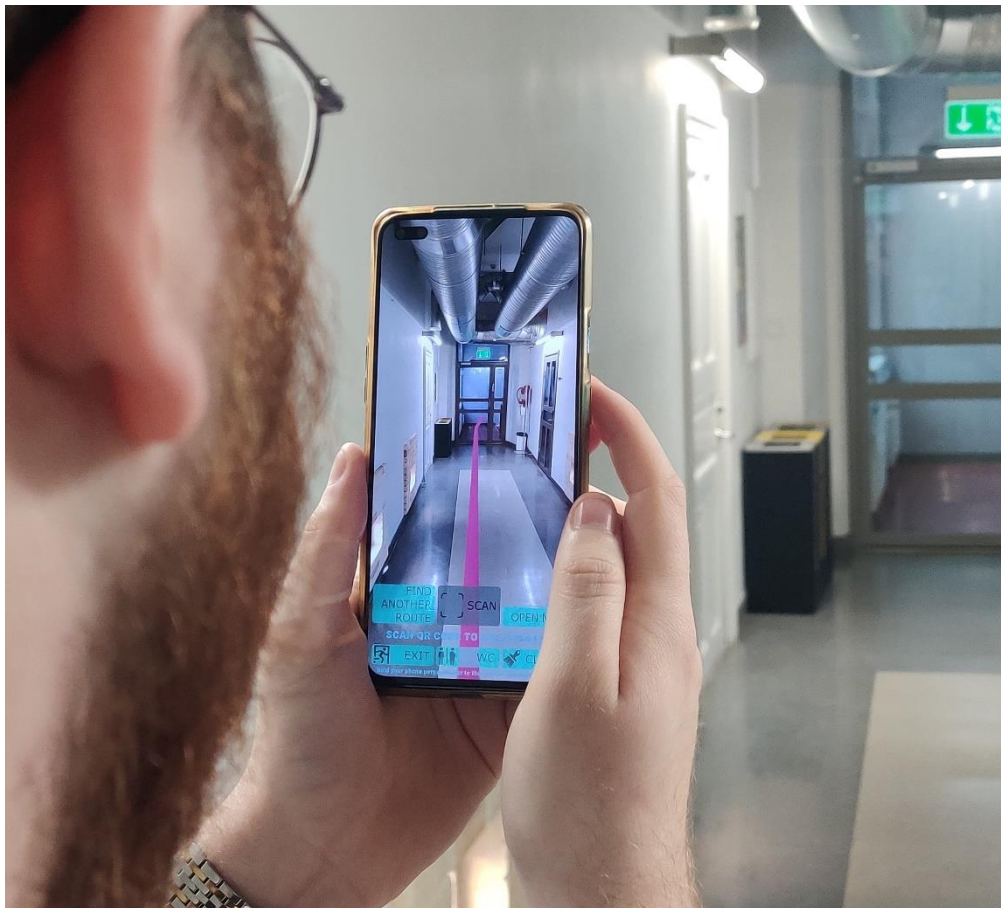
- Tuvastada, kas erialaväline inimene saab rakenduse kasutamisega hakkama ilma eelneva infota rakenduse kohta.
- Testida lõpetatud rakenduse üldist navigeerimisjoonte kuvamise täpsust ja järjepidevust.
- Testida alternatiivsete teede leidmist ja kuvamist.

Kolmanda testimise käigus valiti navigeerimise alguspunktiks üks esimese korruse auditoorium, üks teise korruse auditoorium ning kolmanda korruse ruum. Testija sai ilma lisainfota rakenduse kasutamisega hakkama pärast kiiret rakendusega tutvumist erinevaid valikuid vajutades ning telefoni eri suundadesse osutamise. Navigeeriti valitud alguspunktidest esialgu tualetini, seejärel väljapääsuni. Seejärel testiti samadest alguspunktidest alternatiivsete teede valimise võimalust. Joonisel 3.4 on näha testijat teise korruse koridoris liikumas väljapääsu poole. Testija on pildil valinud alternatiivse väljapääsu. Kolmanda testimise tulemused olid järgnevad:

- Tuvastati, et QR-koodi skaneerimine toimus rakenduses ebastabiilselt ehk vajutades SCAN nuppu, ei toimunud alati esmakordselt skaneerimine ning kasutaja ümber positsioneerimine. Lisaks kui telefoni liigutada QR-koodi skaneerimise ajal, võis kasutaja virtuaalses keskkonnas siiski olla nihkes.

- Leiti, et kui skaneerimine toimus korrektselt, kuvati kasutajale navigeerimisjooned õiges suunas ning järjepidevalt.
- Leiti, et kasutaja soov võib olla kasutusjuhise rakenduse avades või paremat järgnevat valikut märgatavust.
- Tuvastati, et alternatiivse teekonna kuvamine toimub eesmärgipäraselt.

Kolmanda testimise tulemusel leiti, et QR-koodi skaneerimise protseduur vajab parandamist nii skaneerimise sujuvamaks muutmise kui ka potentsiaalselt QR-koodi kuju analüüsimise teel. Samuti leiti, et rakenduse kasutajasõbralikkusele aitaks kaasa kasutusjuhiste kuvamine kas rakenduse esmakordsel kasutamisel või kasutaja valikul.



Joonis 3.4 Kolmanda testimise testija liigub mööda Puiestee 80a teise korruse koridori alternatiivse väljapääsu suunas.

4. RAKENDUSE EDASIARENDUSE VÕIMALUSED

Tulenevalt ühest lõputöö eesmärgist luua baasrakendus, millele oleks tulevikus võimalik edasiarendusi teha, on oluline tuua välja edasi arendamise suunad. Edasi arendamise suunad on jagatud kaheks: rakenduse täiendamine ning funktsioonide lisamine.

4.1 Rakenduse funktsioonide parendamine ning täiendamine

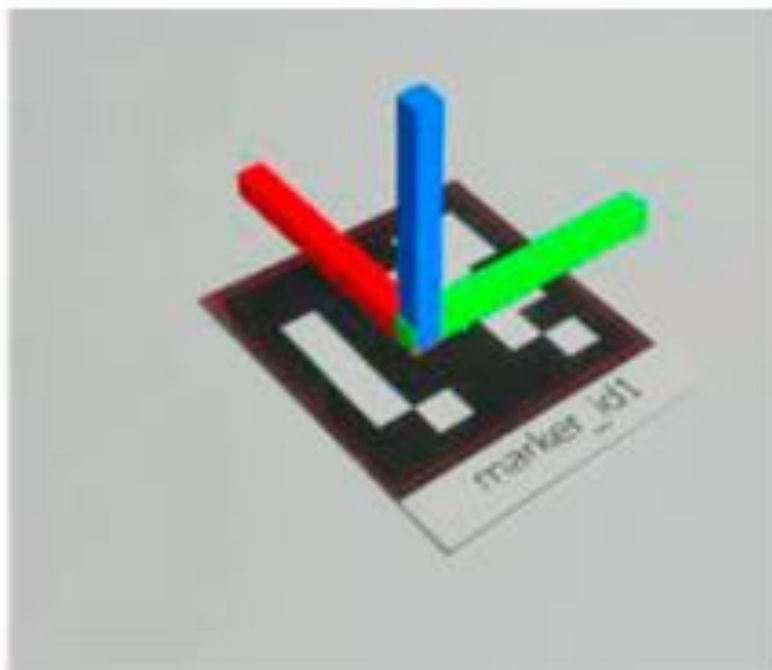
Olemasolevate funktsioonide parendamise eesmärk on jätta rakenduse tööpõhimõte samaks, kuid parandada olemasolevad puudused. Olemasolevatest funktsioonidest saaks parendada QR-koodi nurga alt skaneerimisest tulenevat keskkonna nihet. Selleks tuleks rakendada algoritmi, mis analüüsib QR-koodi kuju selle skaneerimise hetkel. Olemasolevate funktsioonide täiendamise all mõeldakse sihtkohaobjektide, sihtkohagruppide lisamist ning rakenduses navigeeritava keskkonna laiendamist. Funktsioonide täiendamine ei nõua rakenduse loogika muutmist ja on vähem töökulukas kui uute funktsioonide lisamine.

4.1.1 QR-koodi kuju analüüsimine

Selleks, et vältida olukorda, kus nurga all skaneeritud QR-kood põhjustab kogu keskkonna ning navigatsioonijuhiste nihkes oleku, on võimalik implementeerida QR-koodi kuju analüüsimise algoritmi. Algoritm tuleks implementeerida kas juba rakenduses kasutusel oleva ZXing [18] tarkvara funktsioonidesse või leida QR-koodi lugemise tarkvara, millesse on juba eelnevalt QR-koodi kuju analüüsimise algoritm implementeeritud.

QR-koodi kuju analüüsi põhimõte seisneb teadmisel, et QR-koodi, mida skaneeritakse, must keskmine osa, on päriselus ruudukujuline ning seda nurga all skaneerides jääb kaamerapildile trapetsi või rööpküliliku kujuline QR-kood (joonis 1.2). Analüüsidest antud nelinurga külgede pikkuste suhteid, külgede vahelisi nurki, on võimalik luua QR-koodi peale selle positsioonile vastav teljestik [8]. Saadud teljestiku suunad (Joonis 4.1) tuleb vastavusse panna virtuaalse mudeli telgedega. See tähendab, et telg, mis mudelis on risti QR-koodiga, tuleb panna vastavusse saadud teljestiku QR-koodiga risti oleva teljega ning arvutada telgede nurkade erinevus ning teha seda kõikide telgedega. Saadud telgede erinevused, tuleb anda sisendiks ReadQRCode funktsioonile, mis

kohandab kasutaja asukohta mudelis vahetades ARSession objektile antud positsiooni rotatsiooni koordinaate.



Joonis 4.1 QR-koodile x, y, ja z teljestiku loomine [8]

4.1.2 Klassiruumidesse navigeerimise funktsiooni lisamine

Rakenduse testimise tagasisidest ilmnis, et suure tõenäosusega üks kasutaja soov rakenduse funktsionaalsuse jaoks oleks sihtkohagrupidelle klassiruumide lisamine. Antud muudatus vajaks sihtkohaobjektide lisamist igasse klassiruumi vastava klassiruumi nimega. Lisaks oleks vajalik teha muudatus kasutajaliideses, mis võimaldaks kasutajal kas valida loendis klassiruumi number või trükkida see käsitsi sisse. Edaspidi võib rakenduse siduda Tartu kolledži tunniplaaniga, kus kasutaja peab vaid sisestama enda eriala ning aasta, vajadusel ka aine rühma. Rakendus saab seejärel kasutajale sihtkohagrupi valikute seast klassiruumi valides ise leida vastavalt saadud tunniplaanile ja majaplaanile ruumi, kuhu kasutajal tõenäoliselt vaja minna on.

4.1.3 Rakenduse keskkonna laiendamine

Kuna selle lõputöö rakendus on loodud ühe kolmest Tallina Tehnikaülikooli Tartu kolledži hoone näitel, oleks loogiline laiendada rakenduse keskkonda ülejäänud Tartu kolledži hooneid kaasates. Kolledži hooned asuvad ühe krundi peal ning hooned C ja B (joonis 4.2) on ühendatud vahekäiguga, seega tuleks projekti keskkonda lisada hoonete C ja B mudelid, mida ühendab mudelis hoonetevahelist kõnniteed kujutav horisontaalne pind.

Keskkonna mitmekordse suurendamise puhul tuleks ka analüüsida sellest tulenevat navigatsiooniala arvutamiskiiruse vähenemist ning kas see võib muutuda aeglasematel seadetel takistuseks.



Joonis 4.2 Tallinna Tehnikaülikooli Tartu kolledži hooned A, B ja C

4.1.4 Kasutusjuhise lisamine

Selleks, et rakenduse kasutamine, eelkõige esmakordne kasutamine, kulgeks lihtsamalt, oleks võimalik rakendusele lisada kasutusjuhise. Kasutusjuhise võib olla kas eraldiseisev kuva või rakenduse esmakordsel kasutamisel kuvatav lisainformatsioon.

Eraldiseisev kuva tähendaks seda, et kas enne rakenduse liitreaalsuse funktsiooni kasutamist või vastavalt kasutaja soovile kuvatakse talle uus ekraan, millelt saab kasutaja informatsiooni rakenduse kasutamise kohta. Eraldiseisvale kuvale saab lisada ka rakenduse toimimist selgitava videoklipi. Eraldiseisva kuva eelis kasutajajuhendi rakenduse toimimise ajal kuvamise ees on selgem ülevaade kogu rakenduse toimimise loogikast.

Kasutajajuhist saab kuvada ka rakenduse toimimise ajal. Kui rakendus selle tuvastab või kasutaja ise märgib, et rakendust kasutatakse esimest korda, lisatakse ekraanile järgnevaid tegevusi juhendavad sõnumid. Sõnumite loogika võiks tugineda peatükis 3.1 toodud vooskeemile joonis 3.1. Juhul, kui rakendus ise ei tuvasta esmakordset kasutamist oleks kasutusjuhise toimepõhimõtte sarnane minikaardi toimimisele ehk lülitinupuga saab kasutajajuhist sisse ja välja lülitada. Rakenduse toimimise ajal kasutusjuhiste kuvamise eelis eraldiseisva kuva ees on kasutaja informatsiooniga mitte üle koormamine ja see, et vajalikud juhised kuvatakse kasutajale õigel hetkel.

4.2 Potentsiaalsed lisafunktsioonid

Suur lisaväärtus rakendusele oleks eemaldada rakendusest markerid ja skaneerimine. Populaarsemad lahendused selleks on punktipilvede sobitamise ja majakate meetodid. Punktipilvede sobitamise meetod oleks vaid tarkvaraline ning ei vajaks päris maailma mingite objektide lisamist, seevastu majakate meetod nõuaks riistvaralist lahendust kolledži hoones.

4.2.1 Punktipilvede sobitamise meetodi rakendamine

Raskuskohtasid punktipilvede sobitamise meetodi rakendamisel oleks laias laastus kolm: kolledži hoonest detailse ettevalmistatud punktipilve loomine, protseduurilise punktipilve loomise algoritmi loomine või leidmine ning punktipilvede sobitamise algoritmi loomine või leidmine. Ettevalmistatud ja detailset punktipilve on võimalik teha tänapäevaste LiDAR seadmetega. Protseduurilise punktipilve loomine on vajalik, et reaalses sobitada ettevalmistatud ja genereeritud punktipilve. Kuna enamikel rakenduse kasutamiseks mõeldud seadmetel, nutitelefonidel, pole tänapäeval LiDAR võimekust, oleks vaja punktipilve genereerida, kuid see pole läbi kaamerapildi genereerides LiDAR'iga loodud punktipilvega võrdse kvaliteediga [9]. Punktipilvede sobitamise algoritmi on võimalik nii ise luua kui ka kasutada eelnevalt loodud punktipilvede sobitamise algoritmi, mis ei pruugi kattuda antud rakenduse keskkonnaga.

4.2.2 Majakate meetodi rakendamine

Sarnaselt punktipilve meetodile, puuduks vajadus skaneerimise ja markerite järele ka majakate meetodit kasutades. Projekti keskkonda tuleks lisada vähemalt kolm kasutaja seadmega andmesidet vahendavat majakat, mille järgi kasutajat positsioneerida. Majakate arv võib ka kasvada, olenevalt sellest, kui hästi tuvastab kasutaja seade majakatelt saadud informatsiooni läbi hoone seinte ja korruste.

KOKKUVÕTE

Antud töö põhieesmärk oli luua liitreaalsuse navigatsioonirakendus, mida saaksid kasutada kõik Tartu kolledži üliõpilased ja külalised, et navigeerida Puiestee 80a hoones. Lisaks oli eesmärgiks, et projektile oleks võimalik teha täiendusi ning parandusi, et laiendada rakenduse kasutamisalala ja kasutamisevõimalusi. Teine põhieesmärk oli analüüsida rakenduse toimimist ning selle kasutamise arusaadavust rakenduse potentsiaalse sihtgrupi jaoks.

Töö põhieesmärk sai täidetud: loodi liitreaalsuse navigatsioonirakendus Android nutitelefonidele. Rakendus loodi, kasutades Unity mänguplatvormi, mis omakorda võimaldab projektis kasutada ARFoundation ning ARCore platvorme. Rakendus võimaldab kasutajal Puiestee 80a hoones skaneerida erinevates ruumides asuvaid QR-koode, et alustada navigeerimist vastavalt valikule kas väljapääsuni või tualetini. Kasutajal on võimalik navigeerida kõigil kolmel korrusel ning kasutajat juhendatakse lähima sobiva sihtkohani. Töö lisaeesmärk, milleks oli luua rakendus, mis toimiks ka edasiarenduse baasina, sai samuti täidetud. Projekt on loodud lihtsasti mõistetava ja loogilise struktuuriga: kasutatud on liigendatust ja kirjeldavaid nimetusi erinevatele objektidele. Kasutades käesolevat tööd, on võimalik rakendusele lisada parandusi ja täiendavaid funktsioone ning võimalusi.

Rakenduse testimise käigus leiti, et kuigi kasutajal oleks lihtsam, kui talle kuvatakse kasutusjuhiseid, siis rakendust on piisavalt lihtne kasutada. Pärast kiiret rakendusega tutvumist saab ka erialaväline esmakordne kasutaja rakenduse põhifunktsioonidega hakkama.

Lisaks leiti töö käigus erinevaid kohti, mida saaks projektis parandada ning täiendada. Kasutajat juhendatakse ekraanil hoida seadet QR-koodi skaneerides koodiga ristuvast, et navigeerimine korrektselt toimiks, kuid antud lahendus teeb rakenduse kasutamise ebamugavaks ning ebatäpsusi siiski esineb. Seda probleemi saab lahendada tarkvaraliselt ning lahendust on kirjeldatud peatükis 3.3.1. Lisaks leiti peatükis 3.3 erinevaid täiendusi mida saab rakenduses ellu viia: klassiruumidesse navigeerimise funktsiooni lisamine ning rakenduses navigeeritava ala laiendamine. Töös on ka välja toodud alternatiivsed kasutaja positsioneerimise meetodid, mis eemaldaksid kasutajal vajaduse objekte skaneerida.

SUMMARY

The main aim of this thesis was to develop an augmented reality application, which could be used by all Tartu college students and visitors to help them navigate in the Puiestee 80a building. An additional goal was to make the project relatively easy to use as a base to add fixes, navigable area, or additional options for the user. The second main aim of the thesis was to analyse the functionality of the application and its accessibility.

The main goal for the thesis was completed: an augmented reality navigation application was created for Android smartphones. It was created using Unity game engine which in turn enabled the usage of ARFoundation and ARCore platforms. The user can use the application to scan different QR-codes scattered across the building to position themselves, then begin navigating to the nearest exit or bathroom through augmented reality. Navigation is possible in all three floors of the building and the user can choose to navigate to alternative exits or bathrooms if the closest one is unsuitable. The additional goal of making the application easy to use as a base project for an upgraded version was also completed. The project was made to be easily readable using descriptive names for objects and using a logically structured functionality. Developing the project further should be feasible with the help of this thesis.

It was concluded during testing that while the application would be easier to use for the first-time user if there was some kind of a tutorial when opening the application, it isn't entirely necessary. A first-time user can introduce themselves to how to use the application after a brief experimentation.

Different ways to improve the project were found during and after the development of this application. The user is being instructed to scan the QR-code perpendicular to the wall to position them into the environment at the right angle. Such practice is uncomfortable to the user and not entirely failproof. The solution can be done software wise and it is described in chapter 3.3.1. Some additional ways to improve the project were also described in chapter 3.3 including expanding of the navigable area to the whole campus and adding classrooms to the destination choices. Alternative ways to position the user such as using a point cloud matching algorithm and beacon usage markers are also introduced in the third chapter.

KASUTATUD KIRJANDUSE LOETELU

- [1] N. Ashtari, A. Bunt, J. McGrenere, M. Nebeling, ja P. K. Chilana, „Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities“, *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, apr 2020, lk 1–13. doi: 10.1145/3313831.3376722.
- [2] P. Cipresso, I. A. C. Giglioli, M. A. Raya, ja G. Riva, „The Past, Present, and Future of Virtual and Augmented Reality Research: A Network and Cluster Analysis of the Literature“, *Front Psychol*, kd 9, nov 2018, doi: 10.3389/fpsyg.2018.02086.
- [3] E. Marino, F. Bruno, L. Barbieri, ja A. Lagudi, „Benchmarking Built-In Tracking Systems for Indoor AR Applications on Popular Mobile Devices“, *Sensors*, kd 22, nr 14, lk 5382, juuli 2022, doi: 10.3390/s22145382.
- [4] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang, ja J.-C. Liang, „Current status, opportunities and challenges of augmented reality in education“, *Comput Educ*, kd 62, lk 41–49, märts 2013, doi: 10.1016/j.compedu.2012.10.024.
- [5] Brassea Michael, „Augmented Reality vs Virtual Reality for Manufacturers | Tacton“, 2021. <https://www.tacton.com/cpq-blog/augmented-reality-vs-virtual-reality/> (vaadatud dets 13, 2022).
- [6] Tim Perdue, „Applications of Augmented Reality“, veebr 12, 2020. <https://www.lifewire.com/applications-of-augmented-reality-2495561> (vaadatud dets 14, 2022).
- [7] B.-C. Huang, J. Hsu, E. T.-H. Chu, ja H.-M. Wu, „ARBIN: Augmented Reality Based Indoor Navigation System“, *Sensors*, kd 20, nr 20, lk 5890, okt 2020, doi: 10.3390/s20205890.
- [8] S. Boonbrahm, P. Boonbrahm, ja C. Kaewrat, „The Use of Marker-Based Augmented Reality in Space Measurement“, *Procedia Manuf*, kd 42, lk 337–343, 2020, doi: 10.1016/j.promfg.2020.02.081.
- [9] Karl - Walter Sillaots, „Mobile AR Point Cloud Matching“, UNIVERSITY OF TARTU, Tartu, 2021.
- [10] P. K. V. Jayananda, D. H. D. Seneviratne, P. Abeygunawardhana, L. N. Dodampege, ja A. M. B. Lakshani, „Augmented Reality Based Smart Supermarket System with Indoor Navigation using Beacon Technology (Easy Shopping Android Mobile App)“, *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAFS)*, dets 2018, lk 1–6. doi: 10.1109/ICIAFS.2018.8913363.
- [11] S. Subedi ja J.-Y. Pyun, „Practical Fingerprinting Localization for Indoor Positioning System by Using Beacons“, *J Sens*, kd 2017, lk 1–16, 2017, doi: 10.1155/2017/9742170.

- [12] EDUARD LUTA, „AR & VR Glasses for Business: Comparison and Overview“, aug 26, 2020. <https://augment-it.com/en/ar-vr-glasses-for-business-comparison-and-overview/> (vaadatud jaan 03, 2023).
- [13] David Shapira, „Augmented Reality: A User’s Guide to Smart Phones vs. Smart Glasses“, sept 14, 2020. <https://blog.dormakaba.com/augmented-reality-a-users-guide-to-smart-phones-vs-smart-glasses/> (vaadatud jaan 03, 2023).
- [14] A. Hanafi, L. Elaachak, ja M. Bouhorma, „A comparative Study of Augmented Reality SDKs to Develop an Educational Application in Chemical Field“, *Proceedings of the 2nd International Conference on Networking, Information Systems & Security - NISS19*, 2019, lk 1–8. doi: 10.1145/3320326.3320386.
- [15] XR Bootcamp, „Comparing Unity vs Unreal for VR, MR or AR Development Projects“, mai 06, 2022. <https://xrbootcamp.com/unity-vs-unreal-engine-for-xr-development/> (vaadatud dets 16, 2022).
- [16] Marie Dealessandri, „What is the best game engine: is Unity right for you?“, jaan 16, 2020. <https://www.gamesindustry.biz/what-is-the-best-game-engine-is-unity-the-right-game-engine-for-you> (vaadatud dets 16, 2022).
- [17] Josef Wienerroither, „IfcMax importer plugins for 3ds Max 2015-2023 available“. <https://www.frogsinspace.at/?p=3454> (vaadatud jaan 02, 2023).
- [18] Sean Owen, „ZXing (‘Zebra Crossing’) barcode scanning library for Java, Android“, nov 02, 2022.