



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Innendørs navigasjon ved bruk av AR (utvidet virkelighet)

Mikkel Espolin Birkeland, Truls Martinsen og Espen Haugland Mæland

Informasjonsteknologi/Dataingeniør

Institutt for datateknologi, elektroteknologi og realfag

Rogardt Heldal

02.06.2020 Innleveringsdato

Jeg bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle kilder som er brukt i arbeidet er oppgitt, jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 10.

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel: Innendørs navigasjon ved bruk av AR(utvidet virkelighet)</i>	<i>Dato: 02.06.2020</i>
<i>Forfatter(e): Mikkel Espolin Birkeland, Truls Martinsen og Espen Haugland Mæland</i>	<i>Antall sider u/vedlegg: 46</i>
	<i>Antall sider vedlegg: 3</i>
<i>Studieretning: Dataingeniør og Informasjonsteknologi</i>	<i>Antall disketter/CD-er: 0</i>
<i>Kontaktperson ved studieretning: Rogardt Heldal</i>	<i>Gradering: Ingen</i>
<i>Merknader: Ingen</i>	

<i>Oppdragsgiver: Bouvet Norge AS avd Bergen</i>	<i>Oppdragsgivers referanse: Ingen</i>
<i>Oppdragsgivers kontaktperson: Fredrik Chrislock</i>	<i>Telefon: 97762104</i>

<p><i>Sammendrag:</i></p> <p><i>Kryssplattform mobilapplikasjon for innendørs navigasjon ved bruk av utvidet virkelighet(AR).</i></p>

Stikkord:

Augmented Reality (AR)	Azure Spatial Anchors	Navigasjon
------------------------	-----------------------	------------

1. INNLEDNING	5
1.1 Motivasjon og mål	5
1.2 Kontekst	5
1.3 Avgrensninger	6
1.4 Ressurser	6
1.4.1 Arbeidsplass	6
1.4.2 Tilgjengelig utstyr	7
2. PROSJEKTBEKRIVELSE	8
2.1 Praktisk bakgrunn	8
2.1.1 Prosjekteier	8
2.1.2 Tidligere arbeid	8
2.1.3 Initielle krav	8
2.1.3 Initiell løsnings-idé	9
2.2 Litteratur om problemstillingen	9
3. DESIGN AV PROSJEKT	11
3.1 Forslag til løsning	11
3.1.1 Gjenkjenning via kamera	11
3.1.2 Trilaterasjon innendørs	11
3.1.3 Interaktiv panorama	12
3.2 Diskusjon av alternativene	12
3.2 Valgt løsning	13
3.3 Valg av verktøy	13
3.4 Prosjektmetodikk	14
3.4.1 Utviklingsmetodikk	14
3.4.2 Prosjektplan	16
3.4.3 Risikovurdering	16
3.4.4 Risikoanalyse	17
3.5 Evalueringsplan	18
4. DETALJERT LØSNING	19
4.1 Forklare scener	19
4.1.1 Meny	20
4.1.2 Navigasjon	21
4.1.2 Konfigurering av ankerpunkt	21
4.2 Spatial Anchors	22
4.2.1 Hva er et Spatial Anchor?	22
4.2.2 AR i relasjon til Spatial Anchors	24
4.3 Navigering og kalibrering av navigasjon	25
4.3.1 Navigering	25

4.3.2 Kalibrering av navigasjon	26
4.4 Visualisering av navigasjon	29
5. EVALUERING	31
5.1 Evaluering av kode	31
5.2 Evaluering av egenskaper	31
5.2.1 Metode og oppsett	31
5.2.2 Resultat	32
Kalibreringstid	32
Nøyaktighet	34
Avstand fra ankerpunkt	36
Oppsummering	36
6. DISKUSJON	38
6.1 Bruk av Unity	38
6.2 Microsoft Hololens	38
6.3 Manglende testmiljø	38
6.4 Evaluering av løsningen	39
7. KONKLUSJON OG VIDERE ARBEID	40
7.1 Kryssplattform	40
7.2 Nøyaktighet	40
7.3 Brukervennlighet	40
7.4 Fremtidsplaner	41
7.4.1 Nettbasert løsning	41
7.4.2 Andre tjenester	41
8. REFERANSER	43
9. APPENDIX	46
9.1 Gantt diagram	46
9.2 Evalueringsresultat	46
9.2.1 Kalibreringstid	46
9.2.2 Nøyaktighet	48
9.2.3 Distanse fra ankerpunkt	49

1. INNLEDNING

1.1 Motivasjon og mål

Utendørs navigasjon er utbredt på markedet. Teknologiske giganter som Apple og Google tilbyr navigasjonsassistanse via apper på mobile enheter. Dette har bidratt til at tradisjonelle kart i stor grad har blitt lagt til side, til fordel for slike applikasjoner. Innendørs navigering er imidlertid et område som i liten grad er utforsket og utarbeidet. Tradisjonelle kart er fortsatt et sentralt verktøy for navigasjon innendørs. Det eksisterer dermed et behov for å utvikle effektiv og tilgjengelig teknologi på dette området.

Utvidet virkelighet (Augmented Reality, heretter forkortet til AR) er ny teknologi som er på vei fremover i markedet. AR er tilgjengelig i de fleste smarttelefoner, og vil kunne bidra til en enklere og mer intuitiv innendørs navigering.

Målet er å lage en mobilapplikasjon som tilbyr navigasjon innendørs ved hjelp av AR. Applikasjonen skal la brukeren velge mellom tilgjengelige destinasjoner i nåværende bygg. Ved hjelp av AR vil piler plasseres langs bakken for å veilede brukeren til valgt destinasjon. Applikasjonen skal være lett tilgjengelig og fungere på operativsystemene Android og iOS.

1.2 Kontekst

Bouvet er et stort konsulentselskap, og møter kontinuerlig med potensielle kunder. For å opprette nye samarbeid, er det nødvendig at selskapet behersker den nyeste teknologien, evner å tenke innovativt og produserer nyttige produkter på markedet.

Studenter er også en viktig målgruppe for selskapet. Studenter har ervervet seg ny kunnskap i fagområdet, og er en ressurs på arbeidsmarkedet. Dersom selskapet gir et godt inntrykk, kan de nyutdannede ønske å ansettes i selskapet. Bekjentskapet kan eksempelvis dannes gjennom deltakelse i messer eller bedriftspresentasjoner, hvor selskaper presenterer sine prosjekter. Bouvet har tidligere hatt stor suksess med et VR-spill, som flere studenter har utforsket under slike settinger.

I dette prosjektet vil studenter og Bouvet samarbeide om en AR-applikasjon. Det er mulighet for at denne applikasjonen også vil kunne videreutvikles etter endt bacheloroppgave. Videreutviklingen vil kunne være en ressurs for Bouvet i relasjon med studenter og kunder.

1.3 Avgrensninger

Applikasjonen utvikles i første omgang til mobile enheter på plattformene Android og iOS. Funksjonalitet på dedikert AR-maskinvare som Microsoft Hololens vil ikke stå i fokus.

Prosjektet utvikles som en prototype. Dette innebærer at flere områder ikke vektlegges, deriblant brukerhåndtering, tilgangskontroll og distribusjon av applikasjonen.

Gruppen vil ta i bruk skytjenester på plattformen Azure. Azure er utviklet av Microsoft, og Azure tilbyr betalte tjenester der kostnad baseres på bruksmengde. Dette betyr at det er økonomiske begrensninger på hvor kraftige, og hvor mange ressurser gruppen har tilgjengelig for bruk. Det vil ikke fokuseres på hvilke andre eventuelle alternativer som eksisterer med tilsvarende produkter.

1.4 Ressurser

Intern veileder fra HVL, Rogardt Heldal, vil bistå med tilbakemeldinger og rådgivning på områder som arbeidsmetodikk, planlegging og skriftlig rapport. I den avsluttende fase av prosjektet bistod også Richard Kjepso fra HVL med tilbakemeldinger på rapporten.

Fredrik Chrislock er gruppens veileder fra Bouvet og har tidligere profesjonell erfaring som spillutvikler. Han har også utviklet tidligere prosjekter for Bouvet som tok i bruk virtuell virkelighet. Fredrik er dermed en viktig ressurs på områder som omhandler verktøy og teknologi som gruppen tar i bruk. Eksempel på dette er utviklingsplattformen Unity, og teknologien AR.

Gruppen har tilgang til Bouvet sin interne Slack kanal. Her kan gruppen nå frem til mer spesifikk kompetanse fra andre ansatte i Bouvet enn dem som er direkte involvert i dette prosjektet.

1.4.1 Arbeidsplass

Fullverdige arbeidsplasser ble stilt til disposisjon hos Bouvet sine kontorer i Bergen. Her var det plass til alle gruppens medlemmer ved siden av hverandre i et åpent kontorlandskap. Denne arbeidssituasjonen var ideell for godt samarbeid. Her var det planlagt at gruppen skulle jobbe sammen fire til fem ganger uken. Bouvet gjorde tidlig grep da COVID-19 situasjonen utartet seg i Norge. Dette gjorde at gruppen ikke lenger kunne benytte de disponerte arbeidsplassene på kontoret. Gruppen jobbet dermed adskilt hjemme hos hver enkelt. Samarbeidet og koordinasjon av arbeid gikk over internett.

1.4.2 Tilgjengelig utstyr

Gjennom prosjektet vil personlige datamaskiner bli brukt til utviklingen. Følgende operativsystem er brukt:

- Windows 10
- macOS Catalina

Følgende (personlige) enheter og operativsystem for testing av applikasjonen ble brukt:

- iPhone 11 Pro Max, iOS 13
- iPad Pro 12.9, iOS 13
- Samsung S8, Android 9.0
- Google Pixel, Android 10.0

Det ble vurdert å bruke Microsoft HoloLens, med tilgang via Bouvet. Tilgangen ble begrenset grunnet COVID-19, og det ble derfor besluttet å ikke anvende dette utstyret.

2. PROSJEKTBEKRIVELSE

2.1 Praktisk bakgrunn

2.1.1 Prosjekteier

Bouvet ASA er et skandinavisk konsultentselskap med rundt 1500 ansatte og 13 kontorer i Norge og Sverige. Bouvet tilbyr konsulenttenester innen en rekke bransjer gjennom sine ansatte med bakgrunn som ingeniører, designere, tekstforfattere, økonomer, statsvitere, psykologer, pedagoger, matematikere og antropologer.

Prosjekteier blir forøvrig Bouvet avdeling Bergen. Det er dette kontoret applikasjonen skal utvikles med hensyn til i første omgang. Gruppen vil i all hovedsak bare forholde seg til denne avdelingen i Bouvet. Bouvet Bergen har på nåværende tidspunkt omtrent 120 ansatte. Flertallet av disse har kontor i Solheimsgaten 15.

2.1.2 Tidligere arbeid

Det er ikke gjort tidligere arbeid i forbindelse med dette prosjektet. Bouvet Bergen har derimot tidligere utviklet en applikasjon som tok i bruk virtuell virkelighet (Virtual Reality, heretter forkortet til VR) for å vise frem teknologi til potensielle kunder og studenter innenfor data og informasjonsteknologi. Erfaringer fra dette tidligere prosjektet vil være relevant for gruppen under utviklingen av navigasjonsapplikasjonen. Den tidligere VR-applikasjonen ble utviklet ved hjelp av Unity. Dette er den samme utviklingsplattformen som gruppen tar i bruk for å utvikle sin egen applikasjon. Utvikleren av den tidligere VR-applikasjonen, Fredrik Christoffer Chrislock, er gruppens nærmeste rådgiver hos Bouvet.

2.1.3 Initiale krav

For at applikasjonen skal løse problemstillingen og anvendes, stilles flere krav.

Høy nøyaktighet

Ved innendørs navigasjon er nøyaktighet mer viktig enn ved utendørs navigasjon. En feilmargen på flere meter er ikke akseptabelt da dette kan plassere brukeren i feil rom, og/eller etasje.

Enkelt oppsett

Det må være enkelt å sette opp navigering for et nytt bygg. Det må kunne gjøres raskt og involvere minimale mengder utstyr. Ideelt sett, bare en AR-kompatibel smarttelefon.

Brukervennlig

Det må være enkelt for en bruker å velge mål for navigering. Veien til målet skal presenteres på en ryddig og intuitiv måte.

Smart valg av sti

Applikasjonen finner beste alternativ for sti gitt begrensninger bestemt av brukeren. Dette kan være å finne en sti som passer for rullestol eller at applikasjonen unngår spesielle områder i et bygg.

Automatisk oppdatert navigeringsmål

Enkelte interessepunkter vil flyttes over tid. Eksempelvis ansattes plassering dersom det ikke er faste arbeidsplasser. Det er verdifullt med automatisk oppdatering av posisjonen til disse punktene i applikasjonen.

2.1.3 Initiell løsnings-idé

Den initielle løsningen var at applikasjon skulle være en mobilapplikasjon. Følgende bruk av applikasjonen er planlagt.

1. Brukeren starter applikasjonen og velger et bygg, eller om mulig velger applikasjonen bygget automatisk basert på GPS
2. Brukeren velger målet for navigering
3. Veien til målet presenteres i applikasjonen
4. Veien til målet oppdateres underveis som brukeren beveger seg
5. Brukeren når målet og applikasjonen kan lukkes

Det er også et behov for en bruker å sette opp nye mål for navigering og/eller nye bygg. Dette vil skje via opplastning av en 3D-modell. Deretter kan mål for navigering plasseres av brukeren. Ikke alle brukeren vil ha mulighet til dette.

2.2 Litteratur om problemstillingen

Flere store tjenester anvender AR for å bistå i navigasjon. Sommeren 2019 lanserte Google Live View tjenesten gjennom Google Maps (Inman, 2019). Her benyttes AR teknologi for å vise brukeren veien til sin valgte destinasjon. Kameraet på telefonen vil registrere omgivelsene, og programvaren vil plassere piler på skjermen. Disse pilene vil på en intuitiv måte vise retningen brukeren bør følge for å nå destinasjonen.

Bruk av AR-teknologien vil unngå begrensninger ved tradisjonell kartnavigasjon, og dermed gi nye muligheter. Ved bruk av tradisjonelle kart må brukeren orientere seg i omgivelsene før

navigasjonen gitt av kartet vil ha nytteverdi. Denne teknikken vil heller ikke ha mulighet til å gi informasjon om plassering av viktige elementer, for eksempel et sikringskap. AR-teknologien gir imidlertid mulighet til disse aspektene ved navigasjon.

AR muliggjør at brukeren kan rette og opprettholde fokus på omgivelsene, samtidig som mer informasjon mottas. Denne styrken ved teknologien medførte at en av de første reelle applikasjonene var jagerfly, hvor piloten kunne opprettholde fokus på sitt ansvarsområde, samtidig som informasjon via AR kunne indikere om registrerte fly var fiendtlige eller ikke (Berryman, 2012). Teknologien har også senere hatt stor nytteverdi på dette forsvarsområdet.

I tillegg til å være essensielt i natur- og bymiljø, er navigasjon også viktig i store kontorlandskaper. I slike omgivelser kan det være vanskelig å orientere seg effektivt. Selskaper kan derfor ha nytte av tilgang til et verktøy som kan overkomme denne utfordringen. Studier tyder på at navigasjonen er mer effektiv ved hjelp av AR-teknologi enn ved tradisjonelle metoder (Rehman, 2016).

Dagens samfunn er preget av god tilgang på mobile enheter. Mobiltelefoner ser ut til å være like effektive verktøy som mer avanserte AR-briller, som Google Glass, i navigasjonsassistanse (Rehman, 2016). Ettersom de fleste enhetene støtter bruk av AR, vil mange brukere kunne dra nytte av AR-teknologien i hverdagen.

3. DESIGN AV PROSJEKT

Designet av prosjektet avhenger i stor grad av hvilke teknologier som skal brukes. I de fleste ledd av utviklingen vil det eksistere flere ulike verktøy som kan tas i bruk for å løse et problem. Valg av verktøy vil da også ofte påvirke videre valg under utviklingen av applikasjonen. Da dette prosjektet omhandler utvikling av en prototype eller MVP (minimum viable product) er ofte tilgjengelighet mer vektlagt enn for eksempel robusthet i valg av teknologier og verktøy.

3.1 Forslag til løsning

For å kunne løse problemstillingen ble det diskutert flere mulige løsninger. Disse løsningene er forklart under, samt de respektive fordelene og ulempene.

3.1.1 Gjenkjenning via kamera

Teknologier som f.eks. Microsoft sin Azure Spatial Anchors tilbyr tjenester som kan bruke kamera på en mobiltelefon til å gjenkjenne posisjoner. Dette kan kombineres med AR til å tilby en oversiktlig måte å navigere. På grunn av funksjonene brukt, støttes bare android versjoner fra og med Android 7.0. For iOS støttes det fra og med iPhone SE med iOS 11 installert (ARCore supported devices, 2020; Apple Device Compatibility, 2020).

Fordeler

- Høy nøyaktighet
- Enkelt oppsett
- Oversiktlig måte å navigere via AR

Ulemper

- Brukeren må bevege kamera for gjenkjenning
- Endring i utseende på bygg vil hindre gjenkjenning av posisjoner
- Støttes ikke av alle mobiltelefoner

3.1.2 Trilaterasjon innendørs

Trilaterasjon innebærer å bruke avstanden mellom flere punkter for å kalkulere posisjonen. Teknologier som for eksempel GPS bruker dette. Grunnet forstyrrelser fra vegger, tak og annet, gir ikke GPS like god nøyaktighet inne i bygninger som utendørs. Ved å plassere sender med faste posisjoner som kan kommunisere med mobile enheter, kan det derimot oppnås god nøyaktighet innendørs.

Fordeler

- Krever ikke kamera
- Ikke avhengig av utseende på bygget

Ulemper

- Forstyrrelser kan lett oppstå
- Flere fysiske enheter må plasseres rundt i bygget

3.1.3 Interaktiv panorama

Ved å sette sammen flere bilder kan en lage et interaktivt panorama. Denne teknikken brukes av blant annet Google Street View (Street View for Mobile, 2020). Ved å bruke en modell av et gitt bygg, kan navigering kalkuleres og stier kan visualiseres i det interaktive panoramaet.

Fordeler

- Krever ikke kamera
- Trenger ikke å lokalisere brukeren

Ulemper

- Mer manuelt for brukeren, ettersom lokalisering ikke blir gjennomført
- Krever mye manuelt arbeid for oppsett
- Krever lagring av mange bilder
- For optimalt resultat vil det være nødvendig med 360° kamera
- Bilder kan bli utdatert ved endring i bygg

3.2 Diskusjon av alternativene

De mulige alternativene har forskjellige fordeler og ulemper, og det er viktig å måle disse opp mot de kravene som ble lagt frem i 2.1.3.

Gjenkjenning via kamera vil ha høy nøyaktighet så lenge bygningen ikke endrer seg. En er også avhengig av å plassere ut gode referansepunkter som ikke endrer seg. Velges det punkter som endres, vil lokaliseringen bli dårlig. Denne løsningen vil kreve mindre forarbeid for å iverksettes i nye bygg enn de andre nevnte metodene. Samtidig er AR en ideell teknologi for å vise brukeren en sti under navigasjon. En ulempe ved å benytte AR er kompatibilitet. Per dags dato støtter ikke alle mobile enheter denne teknologien. Dette gjør at applikasjonen ikke vil være tilgjengelig for alle.

For å løse problemstillingen krever produktet høy nøyaktighet. Ved å bruke trilaterasjon kan en risikere forstyrrelser som ødelegger nøyaktigheten. Grad av forstyrrelse vil kunne variere ettersom midlertidige objekter kan blokkere signaler. Trilaterasjon vil kreve fysiske enheter som gruppen verken har kunnskap om eller midler til å anskaffe.

Interaktiv panorama vil være en manuell løsning og har dermed ikke behov for lokalisering. Applikasjonen vil ikke forstå hvor i et bygg brukeren befinner seg. Det vil være opp til brukeren selv å velge riktig posisjon. Dette er ikke en optimal løsning. Om applikasjonen selv kunne ta seg av lokalisering vil en skape et mer brukervennlig produkt. Denne løsningen vil kreve store mengder forarbeid for hver bygning applikasjonen skal anvendes i. Dette er grunnet den store mengden med bilder en trenger for å skape selve panoramaet. Gruppen har heller ikke nødvendig utstyr tilgjengelig.

3.2 Valgt løsning

Gruppen har valgt å fokusere på den løsningen som tar i bruk kameraet på mobiltelefon og AR-teknologi. Denne løsningen oppfyller kravene til applikasjonen best. Gruppen har også flest ressurser rundt teknologien som brukes i dette alternativet. Den største ulempen ved denne løsningen er at applikasjonen ikke støtter alle mobile enheter. Gruppen har likevel valgt denne løsningen, da flertallet av mobile enheter på markedet i Norge støtter AR. Det satses også sterkt på denne teknologien fra store selskaper i mobilindustrien som Apple og Google. En kan dermed forvente at antallet kompatible enheter vil øke i fremtiden.

3.3 Valg av verktøy

Den valgte løsningen vil bruke teknologier som kan gjenkjenne posisjoner basert på informasjon fra kamera. Her finnes det flere alternativer:

- Google Cloud Anchors
- Azure Spatial Anchors

På grunn av tidligere erfaring med produkter fra Microsoft Azure, ble det valgt å bruke Azure Spatial Anchors.

Azure Spatial Anchors kan bli benyttet i applikasjoner laget med følgende teknologier (Azure Spatial Anchors overview, 2020):

- Unity
- Android
- iOS
- Xamarin

Det er ønskelig at applikasjonen skal være tilgjengelig og kunne kjøres på enheter som bruker operativsystemene Android og iOS. Om en da bygger applikasjonen direkte til disse plattformene vil det kreve to helt forskjellige kodebaser. Dette vil kreve mye tid og kjernekunnskap rundt to helt forskjellige operativsystemer.

Xamarin er en del av Microsoft sin .NET plattform. Verktøyet brukes til å lage kryssplattformapplikasjoner. Dette gjør at gruppen ikke behøver å lage applikasjonen i flere utgaver for de ulike plattformene den skal kjøres på. En vil kunne ha en kodebase for både Android og iOS. Ingen gruppemedlemmer har tidligere erfaring med dette verktøyet, hvilket vil gjøre læringskurven bratt.

Unity vil i likhet med Xamarin tillate gruppen å utvikle applikasjonen til både iOS og Android ved hjelp av en kodebase. Unity brukes hovedsakelig til å utvikle spill på ulike plattformer, og støtter dermed AR og behandling av 3D-modeller. Rådgiver fra Bouvet, Fredrik Chrislock, har lang erfaring med denne plattformen. Det ble derfor besluttet at Unity ville være det mest egnede verktøyet å ta i bruk for å utvikle applikasjonen.

Følgende er en liste over andre verktøy og teknologier brukt under utviklingen av applikasjonen.

- Azure Spatial Anchors
- Unity
- Visual Studio
- Raider
- XCode (kreves for bygging av applikasjon til iOS)
- Github
- Trello

3.4 Prosjektmetodikk

Det er viktig å finne den riktige prosjektmetodikken for et gitt prosjekt. Metodikken er et grunnlag for prosjektet, som vil veilede utviklingen. Etersom ethvert prosjekt er unikt, vil også tilpasset metode variere. Det er viktig å ha en evalueringsprosess rundt valg av metodikk.

3.4.1 Utviklingsmetodikk

Det er manglende erfaring med flere av teknologiene som brukes, hvilket vil gjøre det utfordrende å estimere tidsbruk tidlig i utviklingen. Av den grunn vil en iterativ og smidig metode for prosjektplanlegging trolig fungere bedre for gruppen. Det er derfor valgt å ta i bruk en smidig utviklingsmetodikk i dette prosjektet.

Scrum er et rammeverk som definerer hvordan en gruppe skal jobbe sammen for å løse vanskelige oppgaver på en effektiv måte (What is Scrum?, 2020).

Ettersom gruppen kun består av tre medlemmer, synes det ikke å være behov for en fullstendig Scrum-løsning. Denne avgjørelsen er tatt på grunnlag av antakelsen om at en liten gruppe ikke vil ha nytteverdi av Scrum sine aspekter, som omhandler organisering av større grupper.

Gruppen bruker sprinter med et tidsrom på en arbeidsuke, som i dette tilfelle tilsier rundt 30 timer fordelt på 5 dager. I en sprint arbeides det med et avtalt sett av oppgaver, og i slutten av sprinten skal gruppen sitte igjen med en fungerende iterasjon av produktet. Dette vil si at en fungerende versjon av produktet alltid eksisterer, og ny funksjonalitet kontinuerlig legges til.

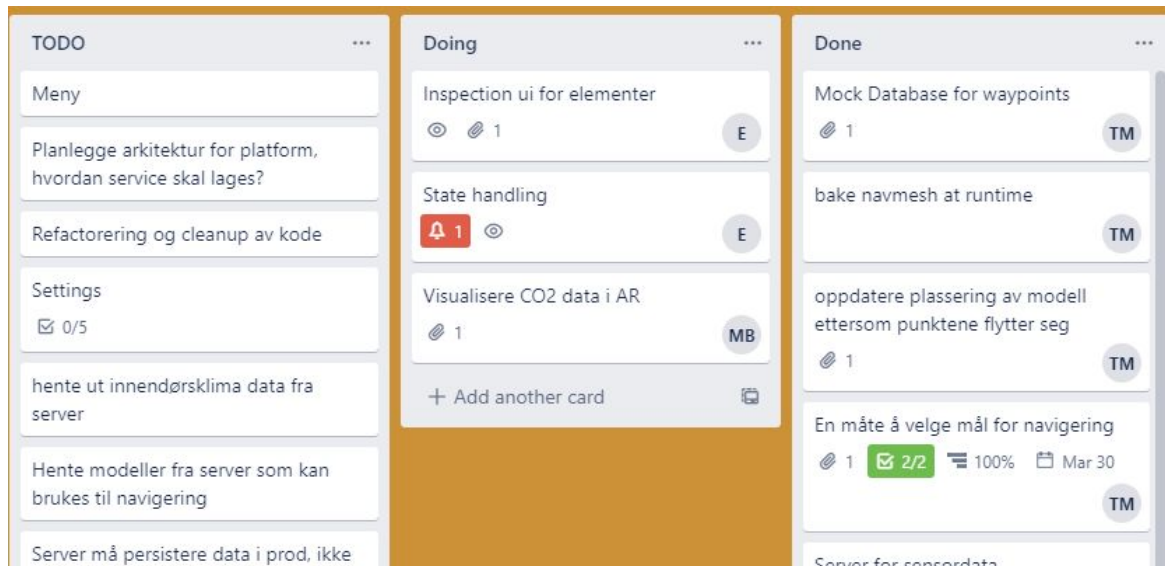
Det settes opp daglige møter mandag til fredag. Møtet begynner i utgangspunktet klokken 12:00, og er av ubestemt varighet. Møtetidspunkter er fleksible og kan endres fortløpende, med hensyn til gruppemedlemmenes eksterne forpliktelser. Møtene tar sted over internett ved bruk av kommunikasjonsverktøyet Discord. Her finnes en persistert kanal for både tekst og tale.

Gruppen anvender en digital tavle med oversikt over egendefinerte oppgaver, samt hvilken fase de befinner seg i. Det er valgt å bruke tre faser. Den første fasen er oppgaver i "TODO". Det vil si oppgaver som ikke har påbegynt arbeid, men som er tilgjengelig for å settes i gang. Her legges det til oppgaver fortløpende fra alle gruppens medlemmer etterhvert som en ser behov for at noe må gjøres. Oppgavene kan knyttes til en eller flere av gruppens medlemmer, eller de kan holdes åpne. Slike åpne oppgaver kan når som helst velges av en som ønsker å utføre arbeidet i oppgaven.

Når en oppgave er påbegynt flyttes den over i den andre fasen på tavlen, "Doing". I denne fasen er det plassert oppgaver som arbeides med for øyeblikket. For at en oppgave ikke skal bli liggende i «Doing»-fasen over lenger tid, er oppgavene relativt små i omfang. Ethvert gruppemedlem har ansvar for å flytte oppgavene de arbeider med i riktig fase.

Den siste fasen i tavlen er "Done"-fasen. Her skal ferdigstilte oppgaver plasseres. Gruppen har valgt å utelate en "test"-fase eller andre lignende evalueringsfokuserede faser. Siden prosjektet omhandler en prototype anses hurtighet av utvikling viktigere enn robustheten til produktet. Dermed har ikke denne typen faser av utviklingsprosessen blitt inkludert på tavlen.

Trello ble tatt i bruk for å lage en digital oppgavetavle. Plattformen gjør det mulig å definere ønskede faser på tavlen. Disse fasene kan umiddelbart bli tatt i bruk og oppdaterer grensesnittet i sanntid.



Figur 3.1, Trello tavle med de tre faser oppgavene kan befinne seg under, TODO/backlog, doing og done.

3.4.2 Prosjektplan

Prosjektplanen er laget i form av et Gantt skjema (9.1). Her finner en milepæler og fokusområdet gjennom prosjektet. Skjemaet fungerer som en pekepinn på hvordan gruppen jobber med prosjektet. Da det blir tatt i bruk en smidig utviklingsmetode er det ikke hensiktsmessig å følge et planlagt oppsett. Dermed vil det kunne forekomme avvik fra Gantt-skjemaet.

3.4.3 Risikovurdering

Risikohåndtering- og vurdering er viktig for ethvert prosjekt. I dette prosjektet anvendes teknologi gruppemedlemmene har lite erfaring med. En risikoanalyse er viktig for å sikre at tiltak er tilgjengelige ved en eventuell uforutsett hendelse.

Analysen er satt opp i et tabellformat med sannsynligheter og konsekvenser. Sannsynligheter og konsekvenser rangeres fra 1-5, og risikofaktoren dannes ved å multiplisere disse. En risikofaktor på under 10 blir regnet som uproblematisk, 10-20 er middels problematisk og alt over 20 er en risiko en bør ta tak i.

En risiko kan forekomme i forskjellige faser gjennom et prosjektløp. Ved å identifisere de største risikoene i enhver fase, kan en forebygge og forberede eventuelle skader.

3.4.4 Risikoanalyse

S : Sannsynligheten for at risikoen oppstår

K : Konsekvensen om risikoen inntreffer

RK : Risikofaktor = S*K

1-5 : 1 er minst og 5 er størst.

Suksessfaktorer	S	K	RF	Tiltak	Fase
Ikke lære Unity	2	5	10	Bruke mer tid på å lære Unity og få opplæring av veileder fra Bouvet.	Oppstart
Spatial Anchors fungerer ikke for vår løsning	1	3	3	Finne et annet alternativ lignende Spatial Anchors..	Oppstart
Problemer med å bygge applikasjonen på iPhone	4	2	8	Kjøre Apple Utvikler lisens	Oppstart
Klarer ikke å bruke Unity sin innebygde navigasjon	4	5	20	Finne ut hvordan sti algoritmen fungerer og lage vår egen implementasjon av denne	Oppstart
Tar lang tid å lokalisere brukeren	3	5	15	Plassere punkter på faste plasser med en unik markering, eksempelvis med QR kode eller et bilde.	Midtveis
Medlemmer blir syke av COVID-19	4	5	20	Gruppen kan jobbe godt sammen slik at om noen blir syke, kan andre medlemmer ta over arbeidet til de har mulighet å jobbe igjen.	Hele Prosjektet
Problemer med server hosting og tredjeparts programmer	5	1	5	Finne andre som tilbyr lignende tjenester eller hoste selv	Hele Prosjektet
Spatial Anchors blir ikke plassert nøyaktig	3	5	15	Feilsøke hvorfor ankrene ikke er så nøyaktige og prøve å plassere ankrene plasser som fungerer bedre	Hele prosjektet

3.5 Evalueringsplan

For å evaluere løsningen, måle løsningens egenskaper og sikre kvaliteten, vil det bli brukt flere metoder for testing.

Integrasjonstesting

For å sikre funksjonaliteten i løsningen vil det bli brukt integrasjonstester. Denne typen tester vil sjekke at de ulike delene av løsningen fungerer korrekt sammen.

Brukertesting

For at løsningen skal være brukervennlig er det viktig at det er flere brukere som tester løsningen. Dette vil gi viktige tilbakemeldinger som forbedrer løsningen.

Måling av egenskaper

Løsningen vil ha flere egenskaper og verdier som kan måles. Dette kan være hvor langt tid det tar før brukeren kan navigere etter oppstart, eller hvor nøyaktig løsningen er. Under utviklingen vil det bli oppdaget flere andre egenskaper som er av interesse å måle.

4. DETALJERT LØSNING

Dette kapitlet beskriver designet av prosjektet og hvordan de forskjellige delene jobber sammen. Det blir forklart hvordan navigasjonen fungerer og hvordan dette blir satt opp i Unity. I tillegg blir også teknologiene brukt forklart i detalj.

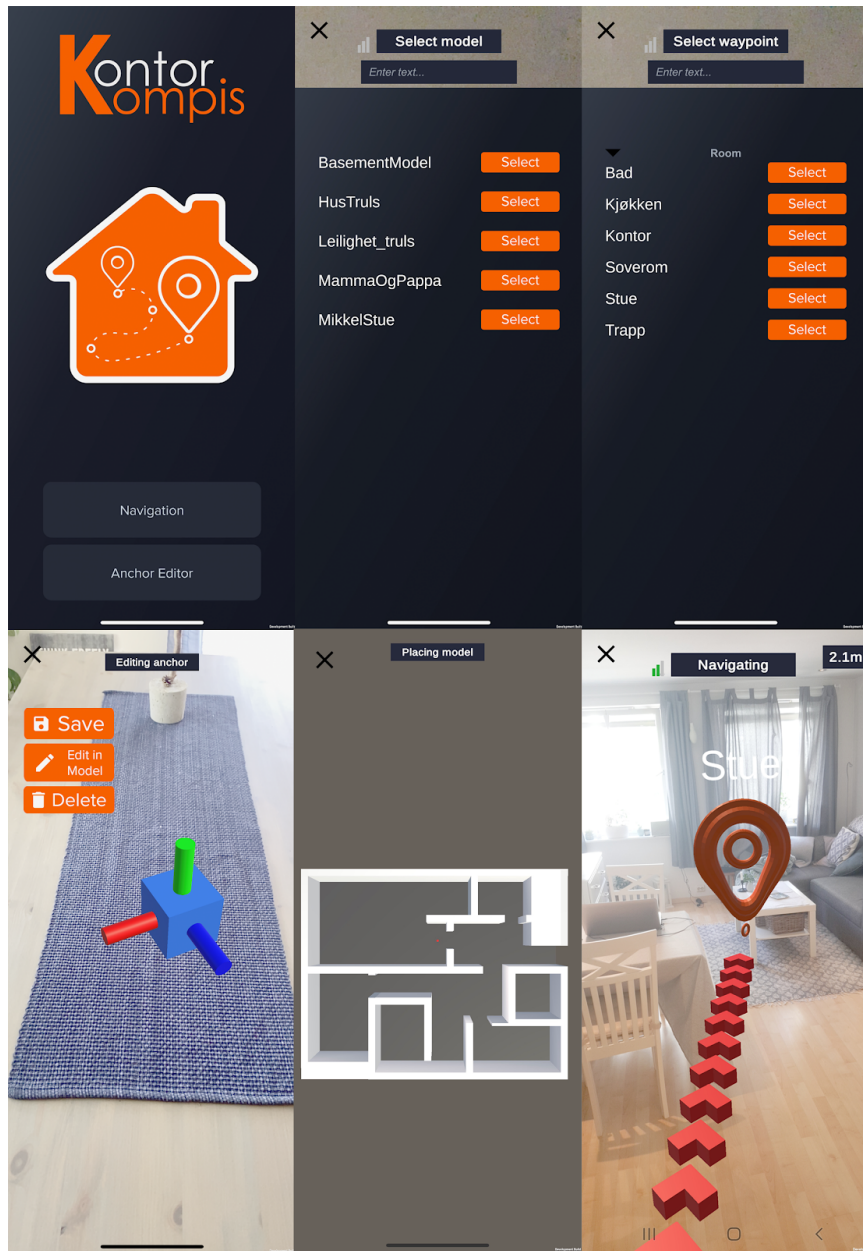
4.1 Forklare scener

I Unity brukes et konsept kalt Scener. Unity har ikke en konkret definisjon for scener, men kan bli sett på som egne miljø. Et miljø kan for eksempel være et spill nivå, innstillingene til spillet eller spill menyen. Scenen bruker da komponenter som er delt på tvers i prosjektet, men bruker egne konfigurasjoner og innstillinger i komponentene. Dette betyr at en komponent kan oppføre seg annerledes i forskjellige scener.

Applikasjonen starter med Meny scenen. Menyen er koblingen mellom navigasjonen og verktøyet for konfigurering av ankerpunkt. Selv om scenene deler flere komponenter blir de brukt forskjellig ved hjelp av tilstander. En tilstand forteller applikasjonen hva den skal gjøre og hva brukeren kan gjøre. Eksempelvis behøves det en modell å navigere i, for å velge et navigasjonspunkt. Hvilke tilstander som eksisterer i hver enkelt scene og hvilken påvirkning dette har på applikasjonen, vil forklares.

Følgende scener blir tatt i bruk:

- Meny
- Navigasjon
- Verktøy for konfigurering av ankerpunkt



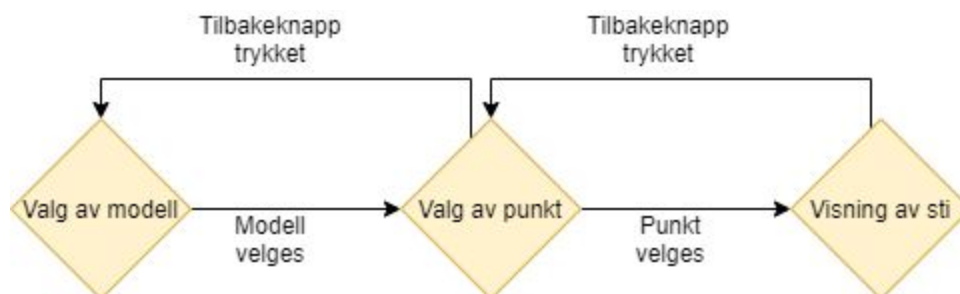
Figur 4.1.1, skjermbilder av løsningen i bruk

4.1.1 Meny

Denne scenen er broen mellom verktøyet for konfigurering av ankerpunkt og navigasjonen. Scenen har to knapper som starter hver sin scene. Dette er også oppstartsmenyen til applikasjonen.

4.1.2 Navigasjon

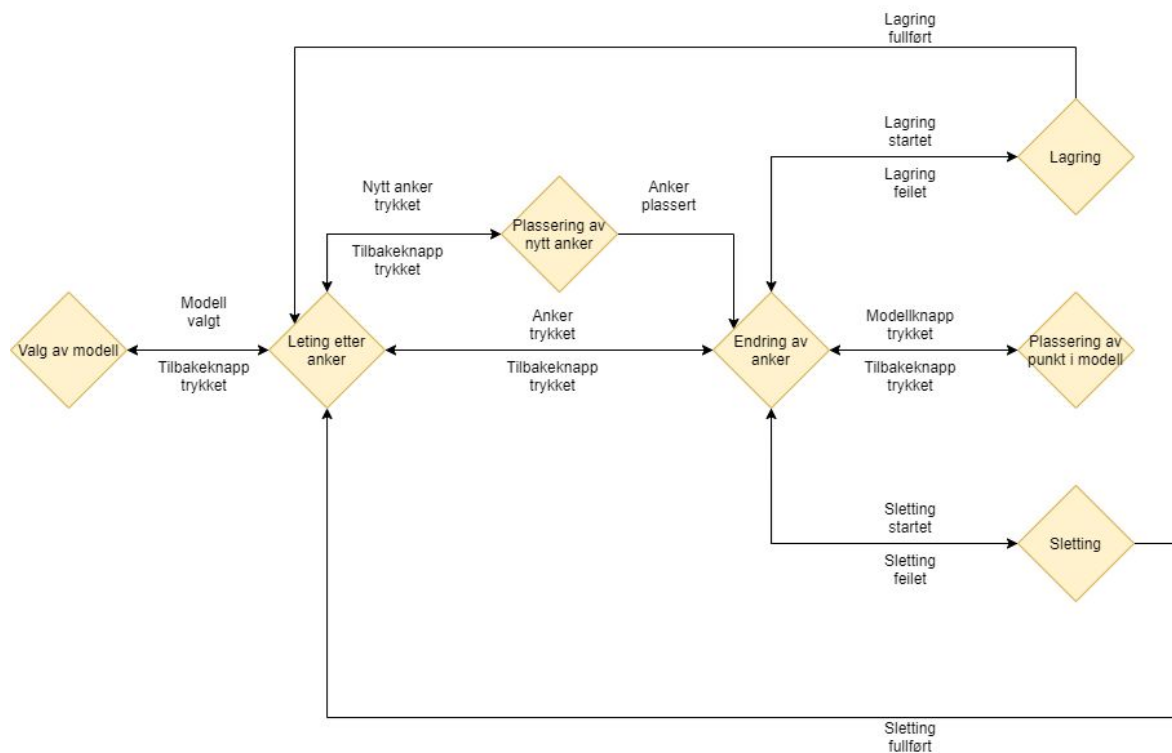
Brukerne av applikasjonen kommer hovedsakelig til å bruke scenen for navigasjon. Formålet med denne scenen er navigasjon. Brukeren velger modellen som skal navigeres i, og deretter en destinasjon. Når en destinasjon er valgt blir en sti visualisert til destinasjonen. Hvordan denne stien blir funnet og visualiseringen blir gjort, blir beskrevet i avsnitt 4.3 og 4.4. I tillegg gir brukergrensesnittet informasjon om avstand til destinasjon, samt interessepunkt i nærheten.



Figur 4.1.2, beskrivelse over alle tilstander i navigasjons Scenen

4.1.2 Konfigurering av ankerpunkt

Anker oppsett menyen er hvor alle endringene til spatial anchors blir gjort. Scenen har mange oppgaver som blir styrt av de forskjellige tilstandene i scenen. Figur 4.1.2 viser et diagram om tilstandene scenen tar i bruk. Tilstandene gjør det mulig å lage nye anker, flytte eksisterende anker, og lagre eller slette anker. Disse endringene blir synkronisert og kan etter lagring bli tatt i bruk i navigasjonsdelen.



Figur 4.1.3, beskrivelse av alle tilstander i scenen om konfigurasjon av anker

4.2 Spatial Anchors

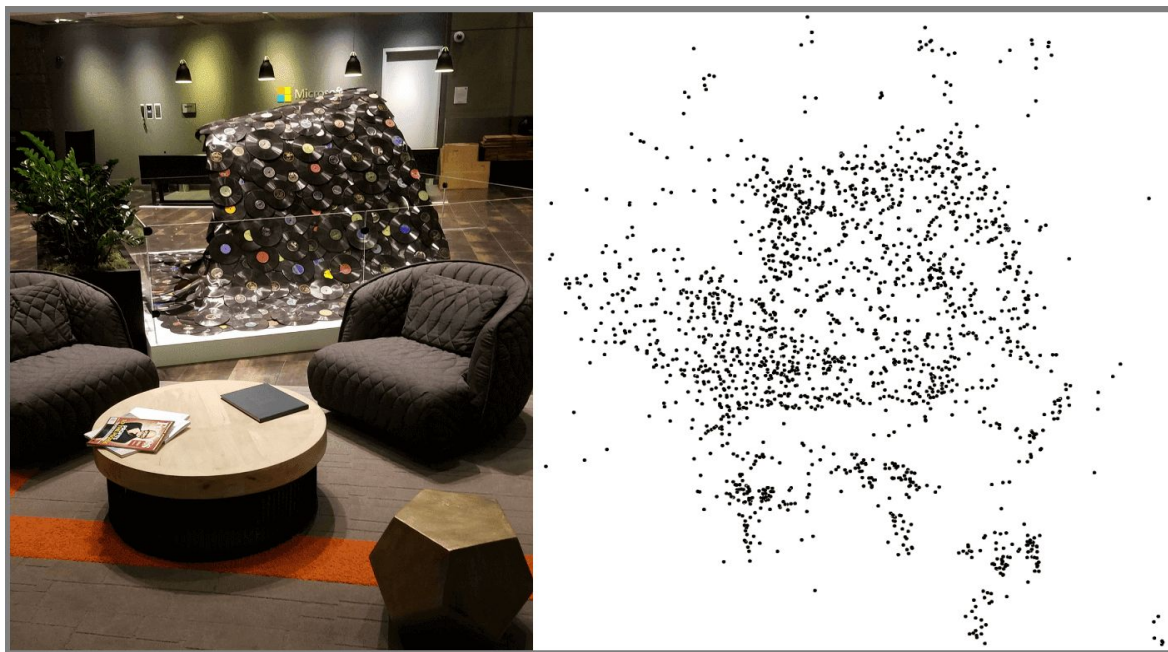
Spatial Anchors representerer et viktig punkt i den virkelige verden, som blir sporet over tid (Spatial anchors - Mixed Reality, 2020). Et anker har et eget koordinatsystem som justeres ettersom mer data samles. Dette betyr at koordinatsystemet kan brukes for å plassere gjenstander i virkeligheten. Etter et anker er plassert er det tilgjengelig for alle støttede mobiltelefoner i nærheten.

4.2.1 Hva er et Spatial Anchor?

For å forstå hva spatial anchor innebærer, bør begrepet «feature points» defineres. Apple Developer Documentation definerer feature points som følgende (RawFeaturePoints - ARFrame, 2020):

Desse punkta representerer markante detaljer funnet i kamerabildet. Posisjonen dems i 3D rommet blir ekstrapolert som en del av bilde analysen ARKit gjør for å nøyaktig bestemme enheten sin posisjon, orientering og bevegelse. Sammen blir disse punktene løst korrelert til konturene av virkelige objekter fra kamera sitt perspektiv.

Et anker inneholder derfor mange feature points som kombinert lager en representasjon av rommet i 3D. Med hensyn til hvordan informasjonen samles, er det fordelaktig å lage flere ankere i samme sesjon. Da vil samme feature points brukes for flere ankere i en sesjon. Dette gjør det mulig å beskrive posisjonen og distansen mellom de plasserte ankrene. Dersom et av ankrene lokaliseres, vil de resterende ankrene fra samme sesjon automatisk lokaliseres og plasseres korrekt. Dette styrker nøyaktigheten av ankrene. Eksempel på feature points er vist på Figur 4.2.1.



Figur 4.2.1: Et landskap og feature points som blir funnet, fra Microsoft
<https://docs.microsoft.com/en-us/azure/spatial-anchors/spatial-anchor-faq>

4.2.2 AR i relasjon til Spatial Anchors

AR blir definert som en direkte eller indirekte sanntidsvisning av et fysisk virkelig miljø som har blitt endret/forsterket ved å legge til virtuell informasjon (Furth, 2011). Teknologien ble, som tidligere nevnt, introdusert til iOS 11 og Android 7.0, og blir anvendt ved bruk av bevegelsessporing og miljø-forståelse.

Bevegelsessporing fungerer ved å bruke SLAM (simultaneous localization and mapping). Durrant-Whyte og Bailey (2006) forklarer at SLAM er en prosess hvor en robot (i dette tilfellet en mobil) genererer et kart og bruker dette til å finne egen posisjon. Miljø-forståelse står i relasjon til feature points. Furth (2011) forklarer sammenhengen mellom prinsippene på følgende måte:

Den bruker videosporingsevnen til å kalkulere i sanntid den virkelige kamera posisjonen og orienteringen relativt til fysiske markører. Da den virkelige kamera posisjonen er funnet, blir et virtuelt kamera plassert i akkurat samme posisjon og en 3D datagrafikk modell kan bli tegnet for å belegge markørene.

Kombinasjonen av miljøforståelse og bevegelsessporing er dermed teknologien som gjør spatial anchors mulig. Ankrene benytter seg av miljø- og bevegelsesdata som AR teknologien samler inn. Microsoft har da bygget spatial anchors som tjeneste på toppen av dette. En tjeneste som kan lagre nødvendig data slik at den på et senere tidspunkt kan plassere punktet på samme sted. Tjenesten forankrer derfor et virtuelt punkt til et punkt i virkeligheten, derav navnet spatial anchors.

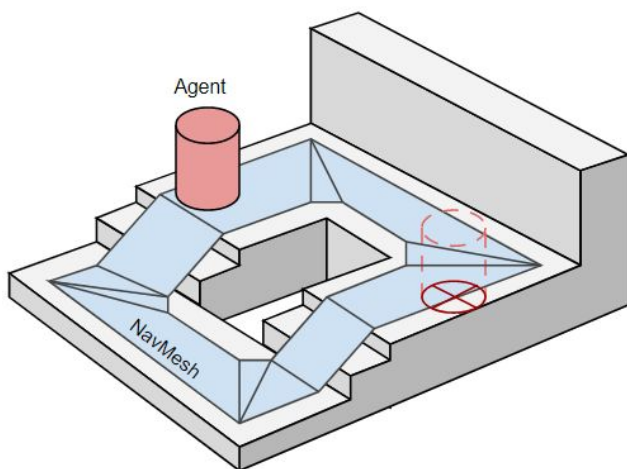
4.3 Navigering og kalibrering av navigasjon

For å kunne navigere innendørs kreves det informasjon om hvor vegger og andre hindringer befinner seg. Det er også nødvendig med informasjon om hvor interessepunkter er plassert. For å kunne ha denne informasjonen krever løsningen en 3D-modell av bygningen. Dette gir nødvendig informasjon om hindringer og om hvor interessepunkter befinner seg.

Det er to utfordringer som må håndteres for å navigere innendørs. Det må skapes en sti i 3D-modellen som brukeren kan følge, og posisjonen av 3D-modellen og virkeligheten må synkroniseres. Dette vil forklares i 4.3.1 og 4.3.2.

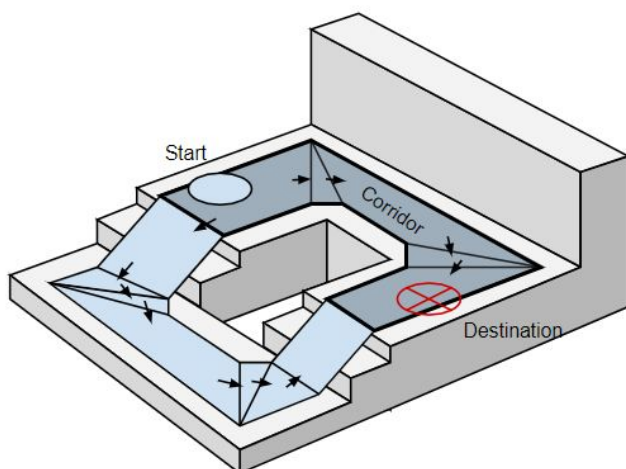
4.3.1 Navigering

For å skape en navigeringssti fra brukerens nåværende posisjon til målet, brukes Unity sitt navigeringssystem. Dette gir mulighet til å skape en «Nav Mesh», som lagrer informasjon om mulige bevegelsesmønstre i 3D-modellen og hvilke hindringer som eksisterer (Figur 4.3.1).



Figur 4.3.1, navigasjonsagent og "Nav Mesh" fra Unity,
<https://docs.unity3d.com/Manual/nav-InnerWorkings.html>

For å kunne skape en sti til et bestemt punkt brukes en navigasjonsagent (Figur 4.3). Flere egenskaper ved navigasjonsagenten påvirker stien som lages. Det kan settes begrensninger på trinnhøyde, vinkel på ramper, bredde og høyde. Dette kan brukes for skape en sti som er tilpasset brukeren.



Figur 4.3.2, navigasjonssti på en "Nav Mesh" fra Unity,
<https://docs.unity3d.com/Manual/nav-InnerWorkings.html>

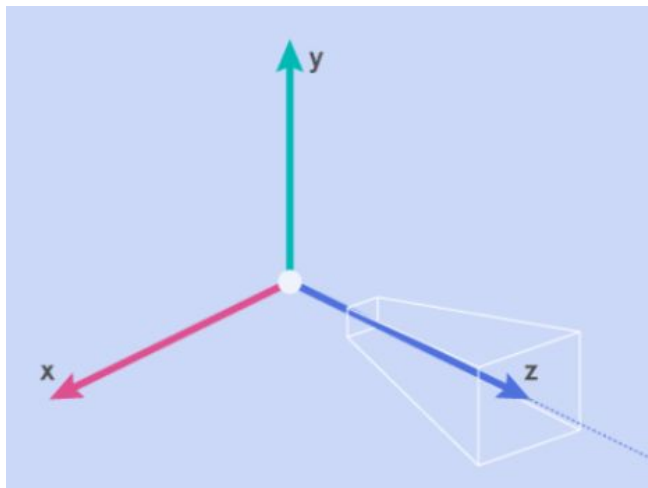
Vanligvis vil området hvor navigasjonsagenten kan bevege seg i, kalkuleres på forhånd. Etersom denne løsningen skal fungere for en gitt 3D-modell, vil det imidlertid være nødvendig å skape dette området under kjøring. For å løse dette ble det brukt et tilleggsværktøy av Unity, som enda er under utvikling (Unity-Technologies, 2020).

4.3.2 Kalibrering av navigasjon

For å kunne synkronisere posisjonen av 3D-modellen med virkeligheten, brukes Spatial Anchors. Ankeret er definert med et punkt i virkeligheten og et punkt i modellen, hvilket gir et «posisjonspar». Disse posisjonsparene gir mulighet til å kalibrere navigering ved å synkronisere 3D-modellen og virkeligheten. Dette oppnås ved å beregne en rotasjon og en forskyvning som brukes til å oversette koordinater i modellen til koordinater i virkeligheten.

Det er viktig å påpeke at plasseringen av punkter i virkeligheten og i modellen har begrensninger, deriblant grad av nøyaktighet. Denne feilmarginen kan reduseres ved å bruke flere punkter, ettersom et gjennomsnitt vil kunne gi et bedre estimat for rotasjon og forskyvning.

For å avgjøre rotasjonen til 3D-modellen er kun rotasjonen rundt y-aksen relevant (Figur 4.3.3). Det er antatt at rotasjonen rundt x-aksen og z-aksen er null. Dette gjør det mulig å avgjøre rotasjonen med kun 2 posisjonspar. Dette forutsetter at vektoren mellom koordinatene i virkeligheten eller i modellen ikke er parallell med y-aksen.



Figur 4.3.3, aksene i Unity

Rotasjon

For å finne modellens rotasjon defineres vektorer mellom posisjonsparene. En vektor mellom koordinatene i virkeligheten, og en vektor mellom koordinatene i modellen. Deretter finner en rotasjonen mellom disse to vektorene. Dersom det eksisterer mer enn 2 posisjonspar, brukes den gjennomsnittlige rotasjonen. En implementasjon av dette i C# med Unity sine bibliotek kan være slik (Figur 4.3.4).

```
Quaternion CalculateModelToRealRotation(List<PositionPair> positionPairs)
{
    var rotations = new Vector3();
    for (int i = 0; i < positionPairs.Count - 1; i++)
    {
        var realRotation = (positionPairs[i].real - positionPairs[i + 1].real).normalized;
        var modelRotation = (positionPairs[i].model - positionPairs[i + 1].model).normalized;

        var pair_rotation = Quaternion.FromToRotation(modelRotation, realRotation);

        rotations += pair_rotation.eulerAngles;
    }

    var rotation = rotations / (positionPairs.Count - 1);

    return Quaternion.Euler(rotation);
}
```

Figur 4.3.4, Algoritme for å kalkulere rotasjonen ut fra posisjons parene

Forskyving

For å finne den nødvendige forskyvningen, undersøkes avstanden mellom punktene i virkeligheten og punktene i modellen. Punktene i modellen er rotert med rotasjonen mellom posisjonsparene. Rotasjonen utregnet i forrige avsnitt brukes. En implementasjon av dette i C# med Unity sine bibliotek kan være slik.

```
Vector3 CalculateModelToRealOffset(List<PositionPair> positionPairs, Quaternion modelToRealRotation)
{
    var positions = new Vector3();
    foreach (var pair in positionPairs)
    {
        positions += pair.real - modelToRealRotation * pair.model;
    }
    return positions / positionPairs.Count;
}
```

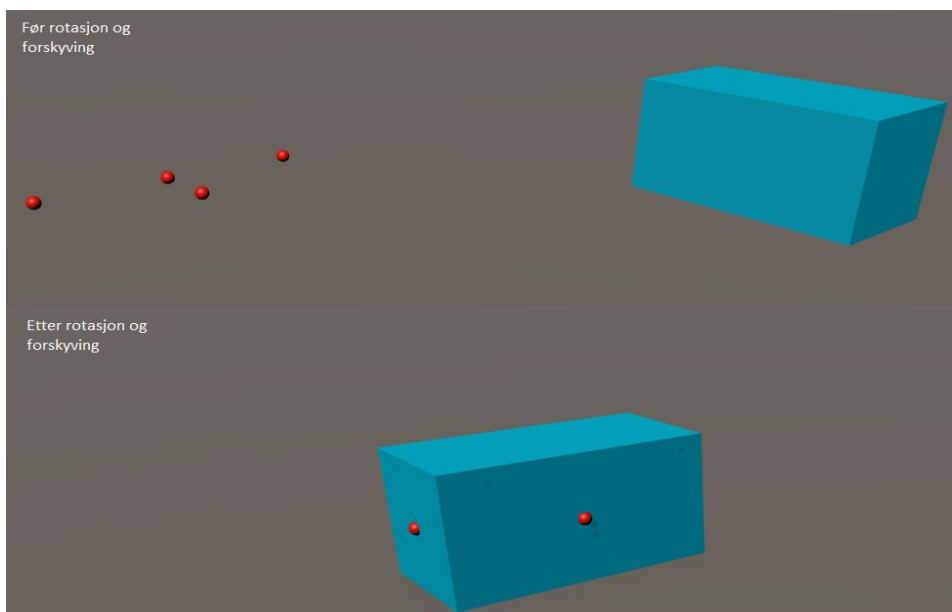
Figur 4.3.5, Algoritme for å kalkulere forskyving ut fra posisjons parene og rotasjon

Oversettelse av koordinater

Ved å bruke rotasjon og forskyving kan et punkt i modellen oversettes og plasseres i virkeligheten. Dette gjøres ved å rotere punktet og deretter legge til forskyvingen (Figur 4.3.6). Dette gjør det mulig å plassere interessepunkt i virkeligheten for brukeren å navigere til, samt at stien som lages av navigasjonsagent blir plassert riktig i virkeligheten.

```
var translatedPoint = modelToRealRotation * point + modelToRealOffset;
```

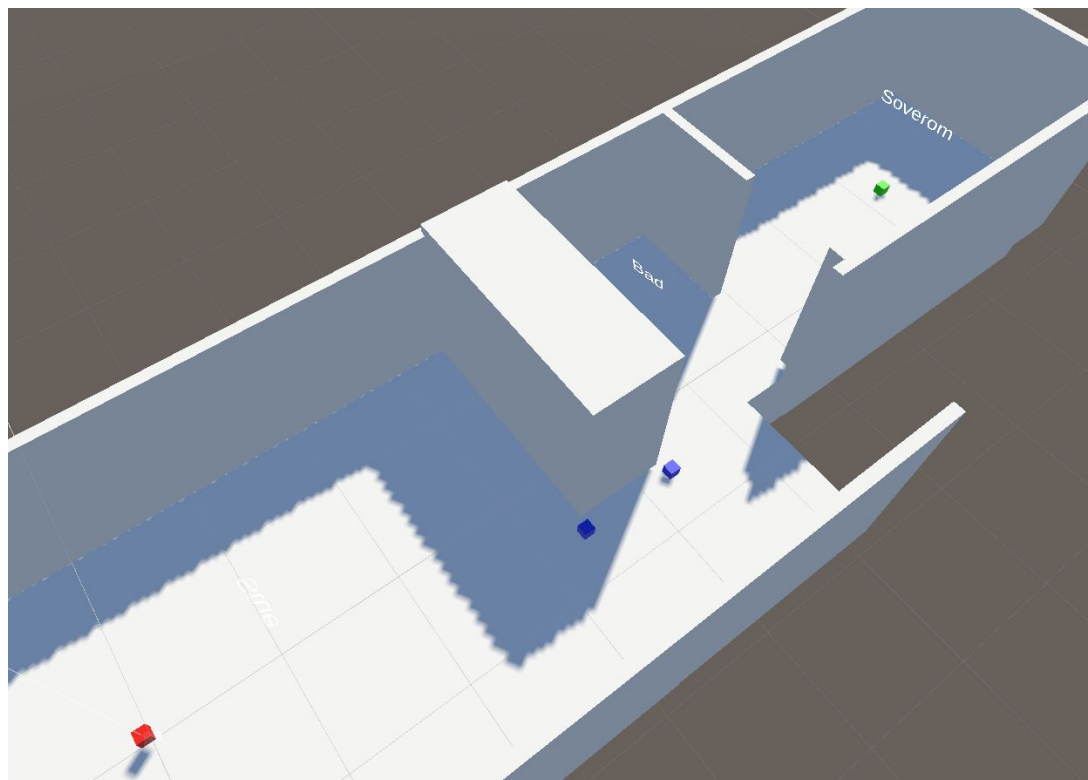
Figur 4.3.6, oversettelse av punkt fra koordinater i modellen til koordinater i virkelighet



Figur 4.3.7, et objekt før og etter rotasjons og forskyving

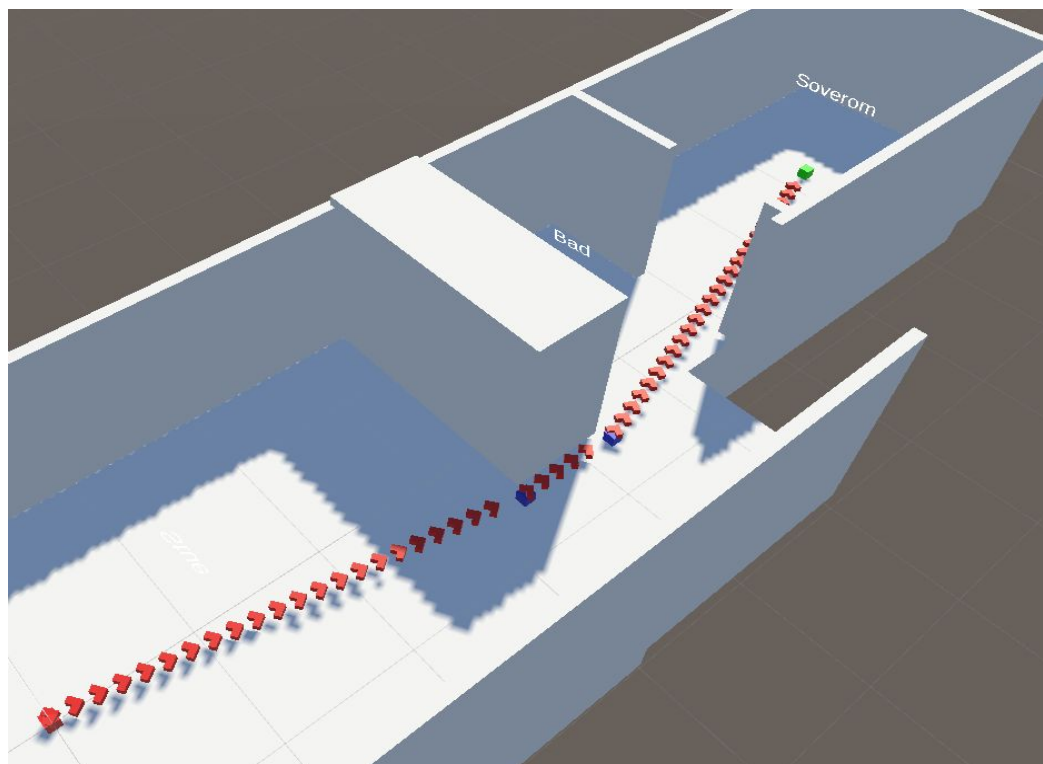
4.4 Visualisering av navigasjon

For å visualisere stien som navigasjonsagent gir må stien videre behandle. Stien som navigasjonsagent gir er kun hjørnene og gir ikke en fullverdig sti som brukeren kan følge. (Figur 4.4.1)

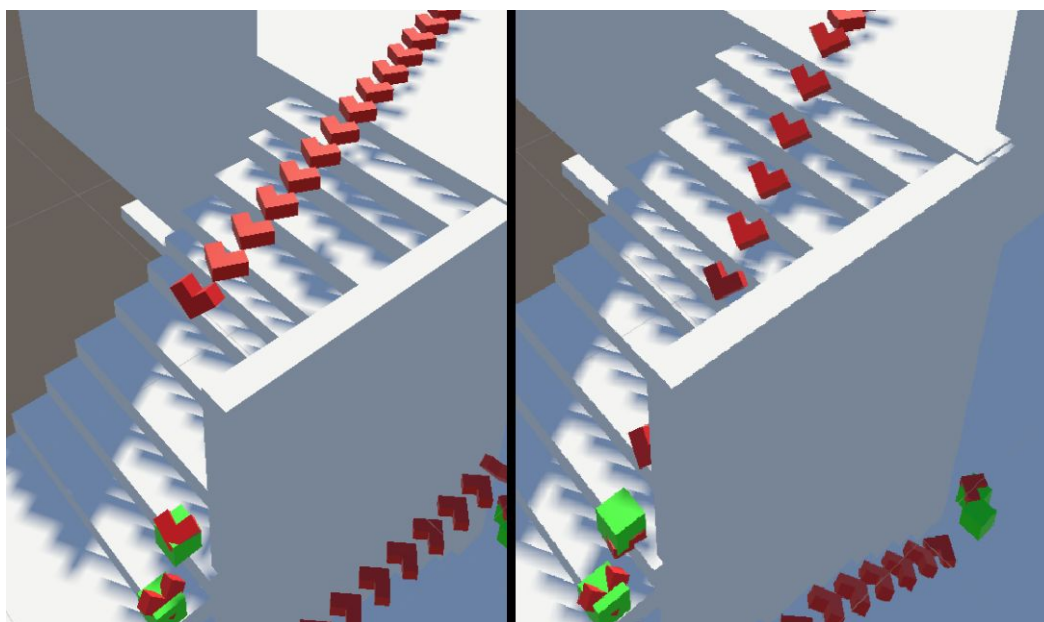


Figur 4.4.1, den røde kubene viser start for navigering og den grønne kubene målet, de blå kubene viser stien navigasjonsagenten gir

For å gi brukeren en sti som er lettere å følge lages det punkter i mellom hjørnene i stien. Det gjøres ved å regne ut en linje mellom hjørnene og plassere punkter på linjen med en gitt avstand mellom punktene (Figur 4.4.2). Dette fungerer kun hvis hjørnene er på samme høyde. For å plassere pilene i riktig høyde selv om hjørnene er i forskjellig høyde må det flere utregninger til. Ved å bruke en stråle som går parallelt med y-aksen nedover og finne punktet der strålen treffer modellen kan pilens posisjon forbedres. Strålen må starte i samme høyde som det høyeste hjørnet pilen er plassert imellom. Resultatet av å anvende dette er at pilene får riktig høyde i forhold til modellen. (Figur 4.4.3).



Figur 4.4.2, lik som figur 4.4.1, men med piler mellom hjørnene



Figur 4.4.3, før og etter høydejustering av piler

5. EVALUERING

Evaluering er basert på evalueringsplanen lagt fram i kapittel 3.5. Dette kapittelet vil vise til konsekvensene av valgene for evalueringsmetodene, samt resultatet av evalueringen.

5.1 Evaluering av kode

Under utviklingen ble det anvendt integrasjonstesting og brukertesting. For å få optimalt utbytte av testmetodene, ble det lagt vekt på at integrasjons- og brukertesting skulle være enklest mulig. Dette hadde påvirkning på hvordan løsningen ble designet.

For å tilrettelegge for integrasjonstester, ble løsningen dannet av flere komponenter. Dette gjorde det enklere å gjenbruke komponenter. Muligheten til gjenbruk og modulær kode medførte til økt kodekvalitet i løsningen.

Ved bruk av brukertesting underveis i utviklingen ble det registrert problemer, som senere ble løst. Disse problemene innebar hovedsakelig visuelle aspekter ved navigering og utformingen av applikasjonens grensesnitt. Grunnet COVID-19 ble det utfordrende å utføre større brukertester, og testingen er derfor avgrenset til gruppens medlemmer.

5.2 Evaluering av egenskaper

For å evaluere løsningen er det flere egenskaper som må måles. Dette er viktig for å kunne sammenligne løsningen med andre løsninger, og for å se løsningens begrensninger. Fullstendig resultat av forsøkene er vedlagt i 9.2.

5.2.1 Metode og oppsett

For å sikre integriteten av resultatet, introduseres det flere kontrollerte variabler. I tillegg gjentas forsøket flere ganger. Egenskapene som skal måles er følgende:

Nøyaktighet

Løsningen bruker ankerpunkt for å kalibrere navigeringen. Nøyaktigheten til disse ankerpunktene bestemmer derfor nøyaktigheten på løsningen. Ankeret blir plassert på et bestemt punkt, men enheten kan under bruk få en annen plassering på dette ankeret. Avstanden mellom den opprinnelige plasseringen og den nåværende plasseringen måles. Det er ønskelig at denne avstanden er lavest mulig.

Kalibreringstid

Ved oppstart vil applikasjonen lokalisere ankerpunkter. Minimum to punkter må plasseres for at navigeringen kan starte. Tiden det tar fra applikasjonen åpnes til tilstrekkelige punkter er plassert, er definert som kalibreringstiden. Det er ønskelig at kalibreringstiden er kortest mulig.

Distanse fra ankerpunkter

En viktig egenskap ved applikasjonen er mulig avstand mellom ankerpunktene og brukeren/den mobile enheten. Denne avstanden måles ved å undersøke hvor langt brukeren kan bevege seg vekk fra ankerpunktene før navigeringen forstyrres. Det er ønskelig at avstanden er størst mulig. Denne avstanden gir grunnlag for hvor tett ankerpunktene må være plassert.

De kontrollerte variablene som introduseres er følgende:

Miljø

Det brukes ulike miljøer for å undersøke hvordan løsningen håndterer variasjon, eksempelvis i forhold til størrelse, gjenskinn i materiale og antall gjenstander i et område.

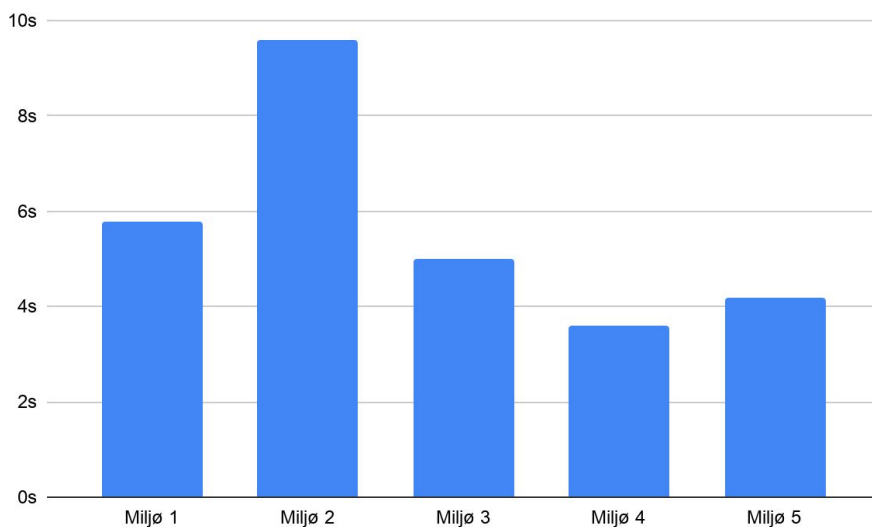
Lysforhold

For å undersøke konsekvensen av ulike lysforhold i løsningen, settes lysforhold som en kontrollert variabel. Sterkt lys er definert som lys på, mens normalt lys er definert som lys av. I tillegg til dette er det naturlig lys fra vinduer.

5.2.2 Resultat

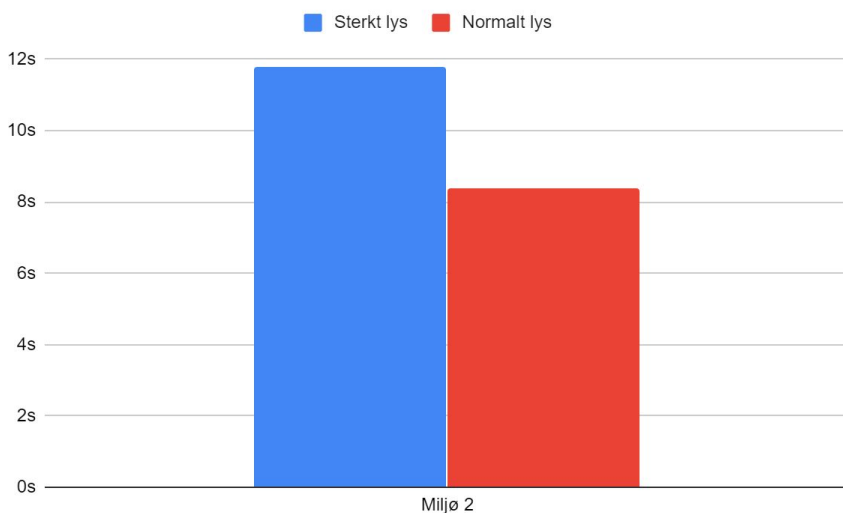
Kalibreringstid

Gjennom forsøkene ble det tydelig at miljø var en viktig faktor. Gjennomsnittlig kalibreringstid (Figur 5.2.1) viser at noen miljøer er mer utfordrende enn andre. Eksempelvis i miljø 2, hvor det er flere blanke flater, hvilket kan være utfordrende for lokaliseringen. Gjennomsnittstiden i alle miljøene for kalibrering var 5.6 s, og median kalibreringstid var 4.0s.



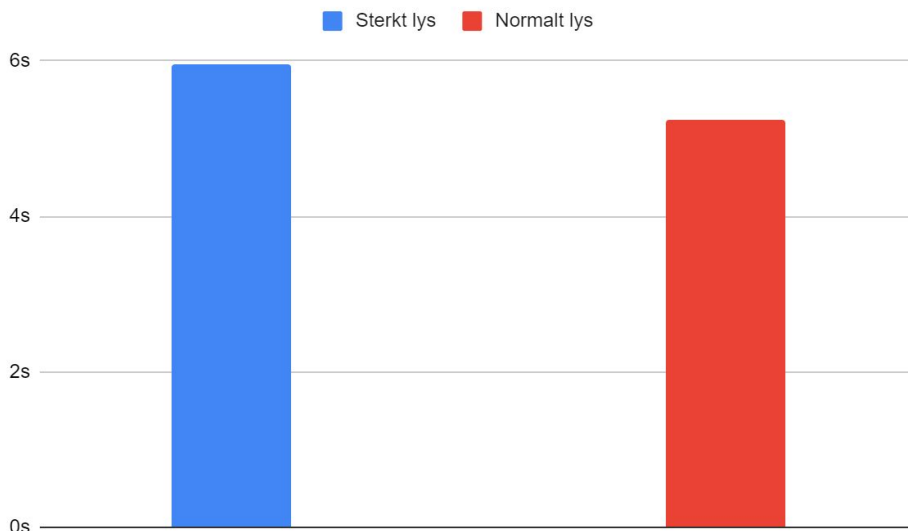
Figur 5.2.1, Gjennomsnittlig kalibreringstid for ulike miljøer

Antakelsen om at blanke flater kan være utfordrende for kalibreringen, styrkes av funnet at gjennomsnittlig kalibreringstid med sterkt lys er større enn ved svakt lys (Figur 5.2.2). Dette kan skyldes at sterkt lys vil skape mer gjenskin. På nåværende tidspunkt er det ikke tilstrekkelig data til å bekrefte denne antakelsen, men det er verdt å bemerke muligheten.



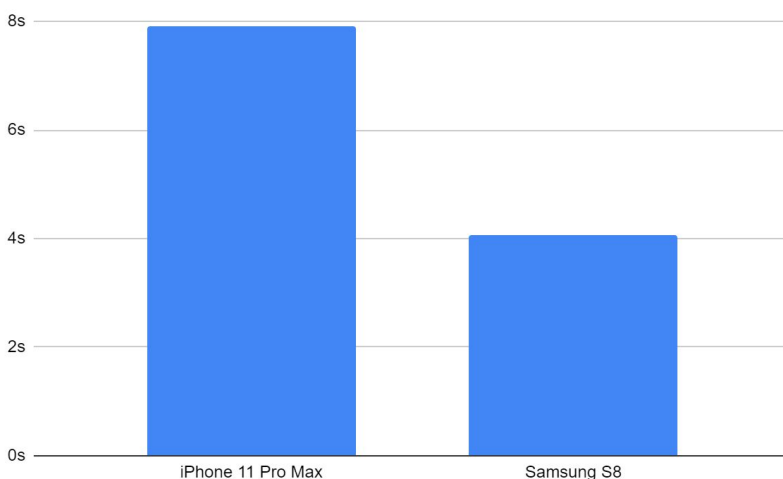
Figur 5.2.2, Gjennomsnittlig kalibreringstid med forskjellige lysforhold for miljø 2

Utenom miljøer med blanke overflater, er det liten forskjell i kalibreringstid mellom sterkt og svakt lys (Figur 5.2.3). Dette tyder på at lysforholdene i forsøkene har hatt liten til ingen effekt på resultatene.



Figur 5.2.3, Gjennomsnittlig kalibreringstid for ulike lysforhold

Kalibreringstiden for de to ulike mobiltelefonene (iPhone 11 Pro Max og Samsung S8) som ble brukt i forsøket, viser til en forskjell på 4 sekunder. Det kunne vært interessant å undersøke om dette skyldes miljø eller ulike egenskaper ved mobiltelefonene. Gruppemedlemmene utførte forsøkene i separate miljøer, hvilket har konsekvenser på verdien av dataene. Begrenset mengde data gjør det ikke mulig å avgjøre dette.

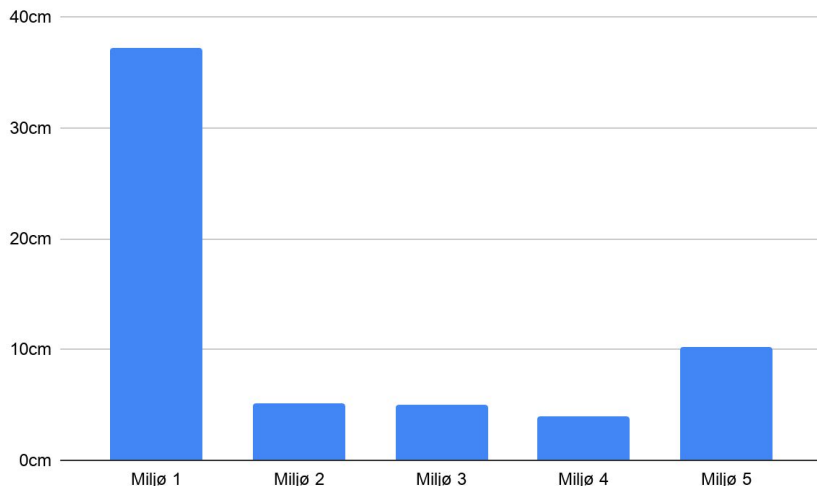


Figur 5.2.4, Gjennomsnittlig kalibreringstid for ulike mobiltelefoner

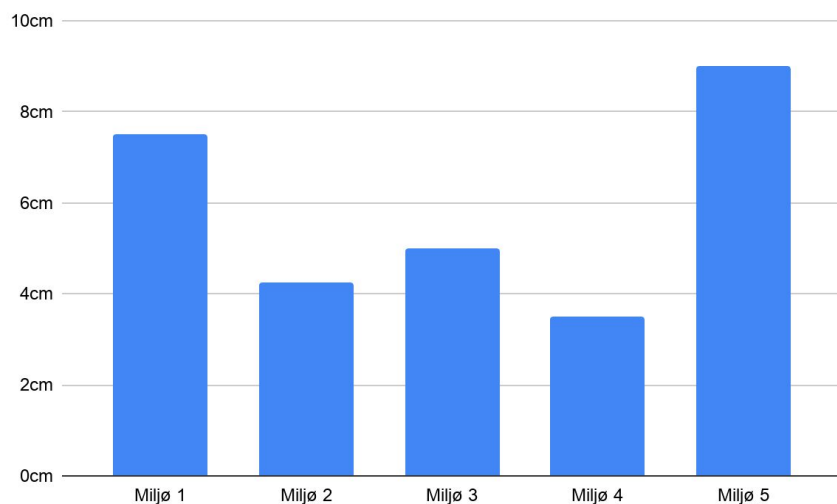
Nøyaktighet

Resultatene fra forsøkene indikerer at miljøet er en viktig faktor også for nøyaktighet (Figur 5.2.5). Spørsmålet er i hvilken grad miljøet påvirker nøyaktighet. Gjennomsnittlig avstand fra

original posisjon viser til større ulikheter mellom miljøene, enn ved bruk av median. Dette kan skyldes avvikende målinger, særlig i miljø 1 (Figur 5.2.5). Gjennomsnittet av alle miljøene er 12.6 cm og median er 5.9 cm. Disse tallene tyder på høy nøyaktighet.

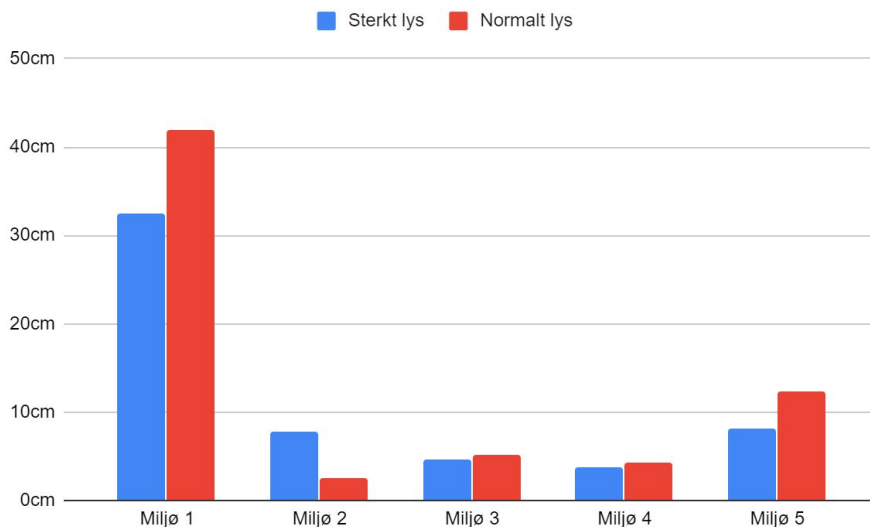


Figur 5.2.5, Gjennomsnittlig avstand fra original posisjon



Figur 5.2.6, Median avstand fra original posisjon

Den gjennomsnittlige avstanden ved de ulike lysforholdene indikerer at forskjellen er liten (Figur 5.2.7). Det ser dermed ut til at endringen i lysforhold har minimal eller ingen effekt på avstanden.



Figur 5.2.7, Gjennomsnittlig avstand fra original posisjon med lysforhold

Avstand fra ankerpunkt

Under evalueringen har det vært mangel på større lokaler til testing. Tilgjengelighet på større lokaler har vært ønskelig for å teste avstand fra ankerpunkt. Det har derfor kun blitt utført forsøk i ett større miljø. I et større miljø er det utfordrende å kontrollere lysforhold, og denne faktoren har dermed blitt utelukket som en kontrollert variabel. Resultatene fra miljøet som er brukt for testing, viser at en bruker gjennomsnittlig kan bevege seg 11.1 meter vekk fra ankerpunktene før navigeringen feilet. Median avstand var 12.0 meter. Disse tallene er viktig med hensyn til anvendelse av løsningen i et nytt bygg. Det gir altså et estimat på hvor nærme ankerpunktene må være hverandre.

Oppsummering

COVID-19 har gjort det utfordrende å gjennomføre ønskelig antall testrunder, samtidig som det ga lite tilgang til testmiljøer. Dette har satt begrensninger på verdien og mengden av data som ble samlet. Ervervet data fra forsøkene som er gjennomført, viser til at miljøet har størst effekt på nøyaktighet og kalibreringstid. Lysforhold har vist til liten eller ingen effekt på disse. Resultatene som ble målt er følgende:

Kalibreringstid

Gjennomsnitt: 5.6s
Median: 4.0s

Nøyaktighet

Gjennomsnitt: 12.6cm

Median: 5.0cm

Avstand fra ankerpunkt:

Gjennomsnitt: 11.1m

Median: 12.0m

6. DISKUSJON

6.1 Bruk av Unity

Gruppen besluttet tidlig å ta i bruk Unity som utviklingsplattform. Kildekoden til applikasjonen er skrevet i C#. Dette er det foretrukne programmeringsspråket på utviklingsplattformen Unity. C# er et objektorientert språk som har store likheter med programmeringsspråket Java. Gruppemedlemmene har hatt undervisning i objektorientert programmering og Java gjennom studiet. Gruppemedlemmene hadde også tidligere erfaring med C#, og selve språket skapte derfor ingen komplikasjoner. Unity som utviklingsplattform var nytt for samtlige av gruppens medlemmer. Som konsekvens av dette gikk det mye tid under oppstartsfasen på å bli kjent med verktøyet.

6.2 Microsoft Hololens

I utgangspunktet var det tenkt at applikasjonen kunne kjøres på Microsoft Hololens, som er maskinvare dedikert til AR i form av briller. Gruppen fikk tilgang Microsoft Hololens via Bouvet sine kontorer, men den fysiske tilgangen har vært fraværende grunnet COVID-19. Dette, i kombinasjon med at denne type maskinvare er lite utbredt, gjorde at gruppen nedprioriterte Microsoft Hololens. I videre arbeid vil det være mulig å gjøre løsningen kompatibel med Hololens eller fremtidige AR-briller fra Microsoft med relativt små endringer.

Dersom gruppen hadde valgt å utvikle applikasjonen eksklusivt for Microsoft Hololens, ville en kunne anvendt Hololens sine mer avanserte funksjoner. Etersom Microsoft Hololens er laget spesifikt for å kunne tilby en god brukeropplevelse i AR, har den muligheter mobiltelefoner ikke har. Den har dybdesensorer som kunne vært tatt i bruk for å øke nøyaktigheten til løsningen. En finner også innebygd gjenkjenning av håndbevegelser, noe som kunne vært tatt i bruk for å lage hele brukergrensesnittet i AR.

6.3 Manglende testmiljø

Gruppen hadde avtalt å utvikle løsningen på Bouvet sine kontorer. Disse planene ble brått endret av COVID-19, som brøt ut tidlig i oppstartsfasen. Dermed var gruppen nødt til å fortsette arbeidet fra forskjellige lokasjoner. Dette gikk spesielt ut over gruppens mulighet til å teste løsningen i tiltenkt miljø. Det er dermed mulig at brukeren møter uventede utfordringer når løsningen skal testes på store områder, som den er ment å fungere i. Det er mulig at feilmarginer som er ubetydelige under testing av løsningen på små områder, blir utslagsgivende

på større områder, slik som kontorlandskap. Uten begrensningen av COVID-19, ville gruppen kunne teste løsningen mer grundig og fått resultater basert på mer data.

6.4 Evaluering av løsningen

Det er gjort flere typer evalueringer av løsningen. Som en kan se i figur 5.2.1 og 5.2.7, er resultatene av testene for kalibreringstid og nøyaktighet nært optimalt. Det er også viktig å trekke frem at gruppen ikke hadde mulighet til å utføre evaluering av løsningen på Bouvet sine kontorer. Det er dermed mulig at evalueringen ikke reflekterer hvordan løsningen fungerer når den brukes på tiltenkt lokasjon.

All testing foregikk i liten skala med hensyn til antall brukere. Det er uvisst hvordan løsningen fungerer når antall brukere økes. Løsningen bruker for eksempel skytjenesten Azure for å anvende spatial anchors. Om dette er skalerbart er usikkert.

Brukertesting er gjort vanskeligere av COVID-19 situasjonen. Dette er en løsning som skal brukes av andre enn gruppen selv. Det er dermed viktig at applikasjonen er enkel og intuitiv i bruk.

7. KONKLUSJON OG VIDERE ARBEID

7.1 Kryssplattform

Det ble laget en kryssplattform mobilapplikasjon, hvilket var et viktig mål for gruppen. Ettersom markedet av smarttelefoner i Norge er splittet mellom iOS og Android var det viktig å kunne tilby applikasjonen på begge plattformene. Applikasjonen til begge plattformene baseres på samme kodebase, hvilket har tillatt rask fremgang i utviklingen til begge plattformene. Utviklingsverktøyet Unity, som gruppen tidlig bestemte seg for å ta i bruk, har mulighet til å lage kryssplattform applikasjoner. Det initielle oppsettet av løsningen i Unity har vært krevende.

7.2 Nøyaktighet

Løsningen skal bistå brukere i å navigere innendørs, og nøyaktighet er viktig for at brukeropplevelsen skal være god. Ved bruk av teknologien spatial anchors oppnådde løsningen svært høy nøyaktighet. 9.2.2 viser til en gjennomsnittlig nøyaktighet på 12.6cm. Dette er tilstrekkelig for å innfri gruppens initielle krav om høy nøyaktighet.

7.3 Brukervennlighet

Når brukeren starter applikasjonen, må en 3D-modell velges som navigasjonsgrunnlag. For øyeblikket må dette gjøres manuelt av brukeren, men ved videreutvikling er det mulig å bruke GPS for å automatisk avgjøre hvilken 3D-modell brukeren befinner seg i. Brukeropplevelsen kunne dermed ha vært forbedret.

Det ble lagt frem et mål om at oppsett av løsningen skulle være enkelt og brukervennlig. Dette målet er kun delvis oppnådd. For at innendørs navigering skal være mulig, må det være tilgjengelig en 3D-modell av bygget. Ideelt kunne løsningen tilbudt funksjonalitet som tillater brukere å lage en modell direkte i løsningen. Slik funksjonalitet er svært ambisiøs, og omfanget er større enn dette prosjektet omfavner. Det ble derfor tidlig besluttet at løsningen må stille krav til at en 3D-modell allerede må eksistere. Flere nye bygninger har slike 3D-modeller knyttet til seg som kan anvendes for navigasjon. En prioritet for gruppen ble dermed å støtte flere format for 3D-modeller. Ettersom Unity er laget for behandling av 3D-modeller, støttes allerede de mest populære formater.

7.4 Fremtidsplaner

7.4.1 Nettbasert løsning

I fremtiden er det ønskelig at løsningen skal ta i bruk nettjenester for en del av funksjonaliteten som for øyeblikket gjøres lokalt. Alle tilgjengelige modeller skal ligge på en server som applikasjonen kommuniserer direkte med. Løsningen vil dermed kunne laste ned kun de ønskede 3D-modellene, og ikke alle tilgjengelige i databasen. Slik funksjonalitet vil redusere størrelsen på løsningen. Dette gir en bedre brukeropplevelse grunnet kortere installasjonstid og redusert bruk av lagringsplass på mobiltelefonen.

Fra en utviklers perspektiv vil dette også gjøre det enklere å oppdatere databasen som inneholder alle tilgjengelige 3D-modeller. Når 3D-modellene er koblet fra løsningen, vil ikke brukere påvirkes av endringer som blir gjort på modeller de ikke bruker. Ved å gjøre 3D-modellene til en egen tjeneste, blir sikkerheten også bedre. En kan lage systemer rundt hvilke brukere som skal ha tilgang til hvilke modeller. Slik systemet er i øyeblikket, følger alle modeller med enhver installasjon av applikasjonen. Som følge av dette vil samtlige brukere ha tilgang til alle modeller. På sikt kan det ikke være slik.

En annen del av løsningen som i fremtiden bør være nettbasert, er lagrede interessepunkter. Lagrede interessepunkter bør lagres på samme måte som 3D-modeller i en skytjeneste. Dette vil gjøre det enklere for brukere å lage sine egne interessepunkter og dele dem med andre. Eksempelvis dersom en bruker befinner seg på et annet sted enn normalt over lenger tid. Da vil brukeren åpne løsningen og oppdatere sin egen posisjon. Den oppdaterte informasjonen vil umiddelbart gjøres tilgjengelig for andre brukere som ønsker å navigere til denne posisjonen.

7.4.2 Andre tjenester

Det var originalt planlagt at løsningen skulle inneholde flere tjenester enn kun navigasjon. En ser dermed for seg at løsningen i fremtiden kan utvides til å bli en plattform som inneholder flere nyttige verktøy og tjenester sentrert rundt kontorlandskap.

I tidlige faser av prosjektet ble det jobbet med en modul som skulle visualisere inn klimaet på et kontor ved hjelp av AR. Her ble det bygget en fysisk modul med blant annet sensorer for temperatur, fuktighet og CO₂. Denne modulen lastet opp informasjonen på en server. På denne serveren ble dataen eksponert som via et API (Application Programming Interface). Ved å ta i bruk dette APIet kunne applikasjonen på mobiltelefoner hente ut oppdatert informasjon om inn klimaet lokalt på et kontor. Arbeidet på denne modulen ble avsluttet før selve modulen var ferdigstilt. Det ble besluttet at gruppens begrensede tid og ressurser burde fokuseres på ett



enkelt område. Derfor valgte gruppen å fokusere på navigasjon. Hadde ikke gruppen gjort denne begrensningen, ville kvaliteten på løsningen vært lavere.

8. REFERANSER

Berryman, D.R., 2012. Augmented Reality: A Review. *Medical Reference Services Quarterly*, 31(2), pp.212–218.

Developer.apple.com. 2020. *Apple Device Compatibility*. [online] Tilgjengelig på:
<https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/DeviceCompatibilityMatrix/DeviceCompatibilityMatrix.html#/apple_ref/doc/uid/TP40013599-CH17-SW10> [Hentet 13 Mai 2020].

Developer.apple.com. 2020. *Rawfeaturepoints - Arframe*. [online] Tilgjengelig på:
<<https://developer.apple.com/documentation/arkit/arframe/2887449-rawfeaturepoints>
> [Hentet 14 Mai 2020].

Docs.microsoft.com. 2020. *Azure Spatial Anchors Overview*. [online] Tilgjengelig på:
<<https://docs.microsoft.com/nb-no/azure/spatial-anchors/overview>> [Hentet 13 Mai 2020].

Docs.microsoft.com. 2020. *Spatial Anchors - Mixed Reality*. [online] Tilgjengelig på:
<<https://docs.microsoft.com/en-us/windows/mixed-reality/spatial-anchors>> [Hentet 10 Mai 2020].

Durrant-Whyte, H. and Bailey, T., 2006. Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, 13(2), pp.99-110.

Furht, B., 2011. *Handbook Of Augmented Reality*. New York, NY: Springer Science+Business Media, LLC, p.8.

GitHub. 2020. *Unity-Technologies*. [online] Tilgjengelig på:

<<https://github.com/Unity-Technologies/NavMeshComponents>> [Hentet 14 Mai 2020].

Google Developers. 2020. *Arcore Supported Devices*. [online] Tilgjengelig på:

<https://developers.google.com/ar/discover/supported-devices#android_play> [Hentet 13 Mai 2020].

Google Developers. 2020. *Street View For Mobile*. [online] Tilgjengelig på:

<https://developers.google.com/streetview/android#street_view_in_the_maps_apis_for_android_and_ios> [Hentet 10 Mai 2020].

Inman, R., 2019. *Take Off To Your Next Destination With Google Maps*. [online] Google.

Tilgjengelig på:

[https://www.blog.google/products/maps/take-your-next-destination-google-maps/.](https://www.blog.google/products/maps/take-your-next-destination-google-maps/)>

[Hentet 8 Mai 2020].

Rehman, U. & Cao, S., 2016. Augmented-Reality-Based Indoor Navigation: A Comparative Analysis of Handheld Devices Versus Google Glass. *IEEE Transactions on Human-Machine Systems*, pp.1–12.

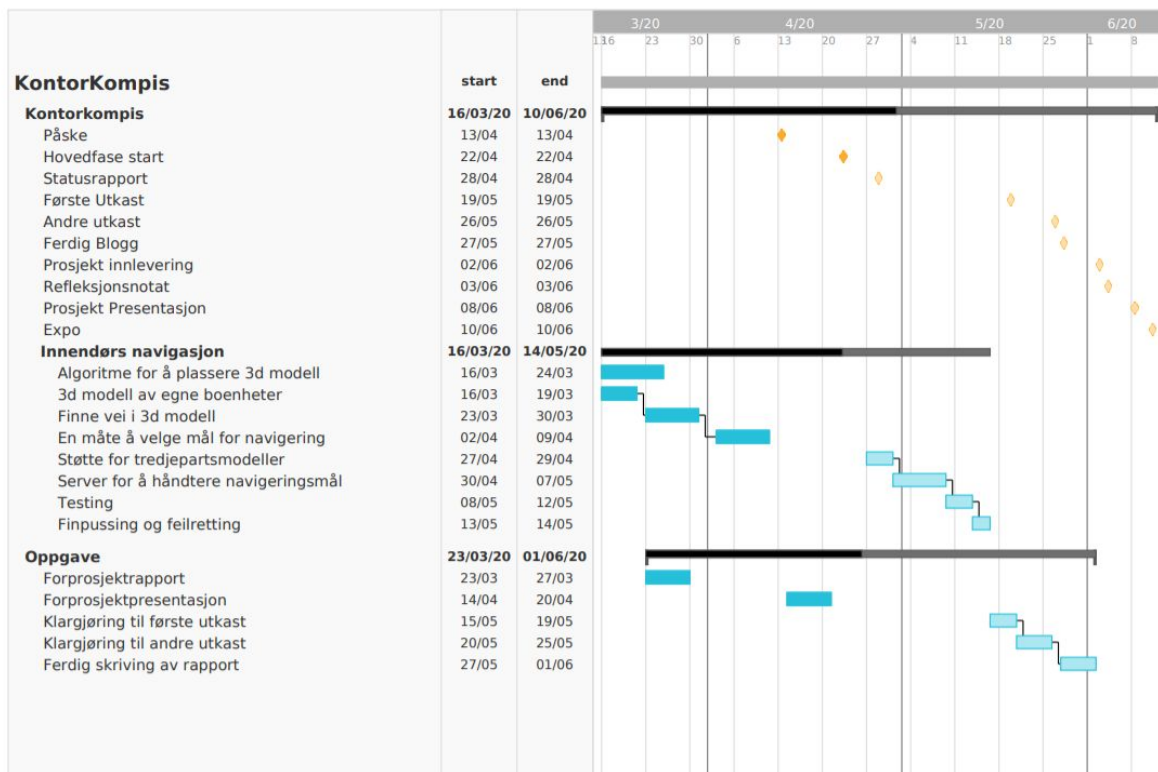


Scrum.org. 2020. *What Is Scrum?*. [online] Tilgjengelig på:

<<https://www.scrum.org/resources/what-is-scrum>> [Hentet 1 Juni 2020].

9. APPENDIX

9.1 Gantt diagram



9.2 Evalueringsresultat

9.2.1 Kalibreringstid

Miljø	Telefon	Tid (s)	Lysforhold
Miljø 1	iPhone 11 Pro Max		11 Sterkt
Miljø 1	iPhone 11 Pro Max		10 Sterkt
Miljø 1	iPhone 11 Pro Max		4 Sterkt
Miljø 1	iPhone 11 Pro Max		3 Sterkt
Miljø 1	iPhone 11 Pro Max		4 Sterkt
Miljø 1	iPhone 11 Pro Max		4 Normalt
Miljø 1	iPhone 11 Pro Max		5 Normalt
Miljø 1	iPhone 11 Pro Max		6 Normalt



Miljø 1	iPhone 11 Pro Max	5	Normalt
Miljø 1	iPhone 11 Pro Max	5	Normalt
Miljø 2	iPhone 11 Pro Max	4	Sterkt
Miljø 2	iPhone 11 Pro Max	22	Sterkt
Miljø 2	iPhone 11 Pro Max	3	Sterkt
Miljø 2	iPhone 11 Pro Max	13	Sterkt
Miljø 2	iPhone 11 Pro Max	17	Sterkt
Miljø 2	iPhone 11 Pro Max	11	Normalt
Miljø 2	iPhone 11 Pro Max	5	Normalt
Miljø 2	iPhone 11 Pro Max	11	Normalt
Miljø 2	iPhone 11 Pro Max	5	Normalt
Miljø 2	iPhone 11 Pro Max	10	Normalt
Miljø 3	Samsung S8	4	Sterkt
Miljø 3	Samsung S8	4	Sterkt
Miljø 3	Samsung S8	4	Sterkt
Miljø 3	Samsung S8	4	Sterkt
Miljø 3	Samsung S8	5	Sterkt
Miljø 3	Samsung S8	4	Normalt
Miljø 3	Samsung S8	4	Normalt
Miljø 3	Samsung S8	8	Normalt
Miljø 3	Samsung S8	3	Normalt
Miljø 3	Samsung S8	6	Normalt
Miljø 4	Samsung S8	3	Sterkt
Miljø 4	Samsung S8	3	Sterkt
Miljø 4	Samsung S8	4	Sterkt
Miljø 4	Samsung S8	2	Sterkt
Miljø 4	Samsung S8	3	Sterkt
Miljø 4	Samsung S8	6	Normalt
Miljø 4	Samsung S8	3	Normalt
Miljø 4	Samsung S8	3	Normalt
Miljø 4	Samsung S8	3	Normalt
Miljø 4	Samsung S8	1	Normalt
Miljø 5	Samsung S8	6	Sterkt
Miljø 5	Samsung S8	3	Sterkt
Miljø 5	Samsung S8	2	Sterkt



Miljø 5	Samsung S8	5	Sterkt
Miljø 5	Samsung S8	6	Sterkt
Miljø 5	Samsung S8	5	Normalt
Miljø 5	Samsung S8	4	Normalt
Miljø 5	Samsung S8	3	Normalt
Miljø 5	Samsung S8	7	Normalt
Miljø 5	Samsung S8	4	Normalt

9.2.2 Nøyaktighet

Miljø	Telefon	Differanse (cm)	Lysforhold
Miljø 1	iPhone 11 Pro Max	45.00	Sterkt
Miljø 1	iPhone 11 Pro Max	2.50	Sterkt
Miljø 1	iPhone 11 Pro Max	10.00	Sterkt
Miljø 1	iPhone 11 Pro Max	4.50	Sterkt
Miljø 1	iPhone 11 Pro Max	100.00	Sterkt
Miljø 1	iPhone 11 Pro Max	100.00	Normalt
Miljø 1	iPhone 11 Pro Max	5.00	Normalt
Miljø 1	iPhone 11 Pro Max	100.00	Normalt
Miljø 1	iPhone 11 Pro Max	5.00	Normalt
Miljø 1	iPhone 11 Pro Max	0.00	Normalt
Miljø 2	iPhone 11 Pro Max	3.00	Sterkt
Miljø 2	iPhone 11 Pro Max	19.00	Sterkt
Miljø 2	iPhone 11 Pro Max	8.00	Sterkt
Miljø 2	iPhone 11 Pro Max	8.50	Sterkt
Miljø 2	iPhone 11 Pro Max	0.00	Sterkt
Miljø 2	iPhone 11 Pro Max	5.50	Normalt
Miljø 2	iPhone 11 Pro Max	7.50	Normalt
Miljø 2	iPhone 11 Pro Max	0.00	Normalt
Miljø 2	iPhone 11 Pro Max	0.00	Normalt
Miljø 2	iPhone 11 Pro Max	0.00	Normalt
Miljø 3	Samsung S8	5.00	Sterkt
Miljø 3	Samsung S8	9.00	Sterkt
Miljø 3	Samsung S8	2.50	Sterkt
Miljø 3	Samsung S8	5.00	Sterkt



Miljø 3	Samsung S8	2.00	Sterkt
Miljø 3	Samsung S8	7.00	Normalt
Miljø 3	Samsung S8	1.00	Normalt
Miljø 3	Samsung S8	11.00	Normalt
Miljø 3	Samsung S8	6.00	Normalt
Miljø 3	Samsung S8	1.00	Normalt
Miljø 4	Samsung S8	5.00	Sterkt
Miljø 4	Samsung S8	4.00	Sterkt
Miljø 4	Samsung S8	5.00	Sterkt
Miljø 4	Samsung S8	3.00	Sterkt
Miljø 4	Samsung S8	2.00	Sterkt
Miljø 4	Samsung S8	3.00	Normalt
Miljø 4	Samsung S8	5.00	Normalt
Miljø 4	Samsung S8	7.00	Normalt
Miljø 4	Samsung S8	3.00	Normalt
Miljø 4	Samsung S8	3.00	Normalt
Miljø 5	Samsung S8	3.00	Sterkt
Miljø 5	Samsung S8	12.00	Sterkt
Miljø 5	Samsung S8	5.50	Sterkt
Miljø 5	Samsung S8	13.00	Sterkt
Miljø 5	Samsung S8	7.00	Sterkt
Miljø 5	Samsung S8	4.00	Normalt
Miljø 5	Samsung S8	6.50	Normalt
Miljø 5	Samsung S8	22.00	Normalt
Miljø 5	Samsung S8	18.00	Normalt
Miljø 5	Samsung S8	11.00	Normalt

9.2.3 Distanse fra ankerpunkt

Miljø	Telefon	Avstand (m)
Miljø 6	Samsung S8	7.9
Miljø 6	Samsung S8	13.1
Miljø 6	Samsung S8	16
Miljø 6	Samsung S8	12
Miljø 6	Samsung S8	6.5