

VR Walk

- La folket vandre

Bachelor, Dataingeniør

Institutt for data- og realfag

Fakultet for ingeniør- og naturvitenskap

Oppdragsgiver:

Lars Peder Bovim

SimArena, HVL

Utarbeidet av D22:

Aleksander Hurtley

Fred Rune Breisnes

Thasajini Rasakunasingam

Jeg bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle kilder som er brukt i arbeidet er oppgitt, jfr. *Forskrift om studier og eksamen ved Høgskulen på Vestlandet, § 9-1.*

TITTELSIDE FOR HOVEDPROSJEKT

| | |
|--|---|
| <i>Rapportens tittel:</i> VR Walk | <i>Dato:</i> 02.06.2019 |
| <i>Forfatter(e):</i> Aleksander Hurtley Fred Rune Breisnes Thasajini Rasakunasingam | <i>Antall sider u/vedlegg:</i> 39 |
| | <i>Antall sider vedlegg:</i> 5 |
| <i>Studieretning:</i> Programutvikling | <i>Antall disketter/CD-er:</i> Ingen |
| <i>Kontaktperson ved studieretning:</i> Harald Soleim | <i>Gradering:</i> Ingen |
| <i>Merknader:</i> | |

| | |
|---|----------------------------------|
| <i>Oppdragsgiver:</i> Lars Peder Bovim ved SimArena, HVL | <i>Oppdragsgivers referanse:</i> |
| <i>Oppdragsgivers kontaktperson:</i> Lars Peder Bovim | <i>Telefon:</i> 55 58 57 50 |

| |
|---|
| <i>Sammendrag:</i> Bachelorprosjektet går ut på å bygge på og utvikle en kommunikasjonsmodul som gjør VR Walk programmet kompatibelt med både Rodby- og Woodway-tredemøller. Disse møllene dominerer markedet innen helsevesenet, som fører til at programvaren kan spres og bli brukt av flest mulige klinikker, uten at de må bruke ekstra ressurser for å unngå den eksklusive tilkoblingen til Rodby-møller slik som det er per dags dato. |
|---|

Stikkord:

| | | |
|---------|----------------------------|------------|
| Unity3D | Seriell Port kommunikasjon | Tredemølle |
|---------|----------------------------|------------|

Høgskulen på Vestlandet, Fakultet for ingeniør- og naturvitenskap

Postadresse: Postboks 7030, 5020 BERGEN

Tlf. 55 58 75 00

Fax 55 58 77 90

Besøksadresse: Inndalsveien 28, Bergen

E-post: post@hvl.no

Hjemmeside:

<http://www.hvl.no>

Forord

Denne rapporten dokumenterer bachelorprosjektet "VR Walk - La Folket Vandre". Bacheloroppgaven er utført av Aleksander Hurtley, Fred Rune Breisnes og Thasajini Rasakunasingam, ved Høgskulen på Vestlandet (HVL) våren 2019.

Vi ønsker å takke:

- Lars Peder Bovim, vår oppdragsgiver, for å gi oss muligheten for å arbeide med dette prosjektet, god veiledning, et godt knutepunkt til gode ressurser og god oppmuntring gjennom hele prosjektperioden.
- Harald Soleim ved Institutt for data- og realfag for god veiledning på rapporten, administrative oppgaver rundt prosjektet og stor interesse for gruppens velvære.
- Thomas Fiskeseth Larsen for teknisk hjelp angående Unity og serial port kommunikasjon med tredemøllene.
- SimArena og Institutt for data- og realfag for tilrettelegging av nødvendig utstyr.
- Tidligere bachelorstudenter Lauritz S. Valved og Bendik M. Bleikli for godt engasjement, hjelp og innsikt i tidligere arbeid med programmet.

Blogg

Bachelorgruppen har en blogg side der oppgaver og fremgang blir notert.

URL: <http://dat190vrwalk.simplesite.com/>

Innholdsfortegnelse

| | |
|---|-----------|
| Forord | 3 |
| Vi ønsker å takke: | 3 |
| Blogg | 3 |
| Ordliste | 6 |
| 1 Introduksjon | 8 |
| 1.1 Problemstilling | 8 |
| 1.1.1 Problemstilling | 8 |
| 1.1.2 Tilleggsmål | 8 |
| 1.2 Mål og motivasjon | 9 |
| 1.3 Begrensninger | 9 |
| 1.4 Ressurser | 10 |
| 1.4.1 Arbeidsplass | 10 |
| 1.4.2 Tilgjengelig utstyr | 10 |
| 1.4.3 Utviklingsmiljø | 11 |
| 1.4.4 Programvare | 11 |
| 1.5 Organisering av rapporten | 12 |
| 2 Prosjektbeskrivelse | 13 |
| 2.1 Om oppdragsgiver | 13 |
| 2.2 Tidligere arbeid | 13 |
| 2.3 Initiale kravspesifikasjoner | 14 |
| 2.4 Initialt løsningsforslag | 15 |
| 3 Prosjektdesign | 16 |
| 3.1 Mulig fremgangsmåter | 16 |
| 3.1.1 Generell kommunikasjonsmodul | 16 |
| 3.1.2 Avatar i skjermvisning | 16 |
| 3.1.3 Fast vinkel mellom skjermene | 16 |
| 3.1.4 Automatisk justering av stibredde | 17 |
| 3.2 Spesifikasjon | 17 |
| 3.3 Verktøy og programmeringsspråk | 17 |
| 3.4 Prosjektutvikling | 19 |
| 3.5 Evalueringsmetode | 19 |
| 4 Løsning | 20 |
| 4.1 Kompatibilitet med oppdaterte versjoner | 20 |
| 4.2 Tilknytning til Woodway tredemølle | 24 |

| | | |
|----------|--|-----------|
| 4.3 | Signal analyse | 26 |
| 4.4 | Utvikling av kommunikasjons script | 29 |
| 4.5 | Metode for å sette hastighet | 32 |
| 5 | Evaluering | 32 |
| 5.1 | Evalueringsmetode | 32 |
| 5.2 | Evalueringsresultat | 33 |
| 6 | Diskusjon | 34 |
| 6.1 | Endring i prioritering | 34 |
| 6.2 | Risikohåndtering ved endring | 34 |
| 6.3 | Fremgangsmåte | 34 |
| 6.3.1 | Utvikling av kommunikasjons-script | 34 |
| 6.3.2 | Videreutvikling av VR-Walk | 35 |
| 7 | Konklusjon og videre arbeid | 36 |
| 7.1 | Måloppnåelse | 36 |
| 7.2 | Kjente feil og begrensninger | 36 |
| 7.3 | Implikasjoner og videre arbeid | 37 |
| 8 | Litteratur/Referanser | 38 |
| 8.1 | Veiledninger/Litteratur | 38 |
| 8.2 | Verktøy | 39 |
| 8.3 | Bilder og figurer | 39 |
| 9 | Vedlegg | 40 |
| 9.1 | Risiko liste | 40 |
| 9.2 | Gantt diagram | 41 |
| 9.3 | Veiledning for bruk og oppsett av kommandomodul. | 42 |
| 9.3.1 | Bruk av Kommunikasjons-script | 42 |
| 9.3.2 | Bruk av signaler | 43 |

Ordliste

| Begrep | Forklaring |
|----------------------|---|
| VR / Virtual Reality | Virtuell virkelighet. Utstyr og system for å kunne se virtuelle 3D-omgivelser i første-person og utføre handlinger i disse. Gir en overbevisende illusjon av å befinne seg i disse omgivelsene. |
| HTC Vive | VR utstyret og løsningen som ble benyttet i dette prosjektet. Inkluderer VR-briller, kontrollere, basestasjoner, og ved dette prosjektet, Vive trackere. (HTC Corporation, 2018) |
| VR Walk | Programvare utviklet for bruk til fysioterapi. |
| C# | Programmeringsspråket som benyttes til å skrive Unity-scriptene for simulatoren. |
| Script | En form for asset. En fil som inneholder kode som kan leses i Unity 3D. Vanligvis skrevet i C#. Ofte assosiert med andre assets eller prefabs. |
| Namespace | Nøkkelord brukt for enklere organisering av et stort antall klasser i en C# applikasjon. |
| Command Prompt | (Ledetekst på norsk). En applikasjon , kommer med de fleste datamaskiner som bruker Windows-baserte operativsystem. |
| Byte array | Samling av data hvor hvert element kan ha en størrelse opp til 1 Byte i størrelse. |
| Unity3D | Programvare for utvikling av grafikk, spill, og andre visuelle medier. |
| Unity Hub | Eget program for å håndtere nye utgivelser, prosjekter, og annet som har med Unity å gjøre. |
| Woodway | En type tredemølle. Veldig mye brukt innen helsevesenet. |
| Rodby | En type tredemølle. Er den møllen som allerede har opprettet kontakt med VR Walk sin kontrollmodul. |
| Trackmaster | En styringsprotokoll for tredemøller sin kommunikasjon over rs-232 tilkobling (Seriell port). |
| Port-sniffer | Programvare som overvåker datastrømmen over en tilkobling. I dette prosjektet vil det referere til trafikken over |

| | |
|--------------|--|
| | usb-til-seriell port. |
| RS-232 | (Recommended Standard 232) er en standard for seriell overføring av binær data mellom forskjellige databehandlingsenheter. Har blitt erstattet med USB i nyere tid for personlig bruk, men spiller en viktig rolle i industrien. |
| Asset | En komponent brukt i Unity 3D. Kan blant annet være et script, prefab, 3D-modell, material, tekstur, shader. |
| Asset Store | En butikk for "assets" som er, mer eller mindre, klar til å tas i bruk i Unity3D. (Unity Technologies, 2018) |
| TextMesh Pro | Pakke i Unity3D, blir brukt til redigering av utseende på tekst. |

1 Introduksjon

Oppdragsgiveren for prosjektet er Lars Peder Bovim som er ansatt ved SimArena samt Høgskulen på Vestlandet. Virtual Reality Walk (VR Walk) er et program som hjelper oppdragsgiveren å vurdere hvordan personer blir påvirket av å bli oppslukt i et dataanimert virtuelt miljø under gange. Formålet er å vurdere virtuell realitet som et supplement til tradisjonell trening. Tidligere har VR Walk blitt utviklet av en bachelorgruppe ved Høgskulen på Vestlandet. Nå ønsker Lars Peder å videreutvikle programvaren samt gjøre den kompatibel med flere tredemøller. Under arbeidet med oppgaven har oppdragsgiveren kommet med et ønske om å legge til et tilleggsmål i problemstillingen. Tilleggsmålet tar for seg oppdatering av programvaren VR Walk slik at den er kompatibel med den nyeste versjonen av utviklingsprogrammet Unity3D.

1.1 Problemstilling

1.1.1 Problemstilling

1. Hvordan kan man utvikle en kommunikasjonsplattform som kan kommunisere med flere tredemøller?
2. Hvordan kan man benytte Unity til å gjøre et VR basert program til kjøre på 1- eller 3-skjermsløsning?

1.1.2 Tilleggsmål

Oppdatere VR-Walk programvaren slik at den er kompatibel med nyeste versjon av utviklingsprogrammet Unity 3D.

1.2 Mål og motivasjon

VR Walk programvaren blir brukt i fysioterapi sammenheng, og er kompatibel med Rodby tredemøller. Grunnet at Woodway tredemøller er den mest dominerende type tredemølle i helsevesenet, er det ønskelig å gjøre VR Walk kompatibel med den type tredemølle. Videre vil muligheten for å benytte seg av skjermvisning, gjøre programvaren enda mer tilgjengelig til de som ikke har tilgang på VR-utstyr. En skjermvisning vil også eliminere ubehagelige opplevelser som enkelte brukere opplever ved bruk av VR-briller. Å gjøre VR Walk mer tilgjengelig for allerede eksisterende utstyr har stor økonomisk betydning for helsevesenet ettersom det ikke blir nødvendig å kjøpe inn nytt dyrt utstyr for å ta i bruk programvaren.

Hovedmål:

- Få opprettet kontakt med andre tredemøller gjennom utvikling av en ny kommunikasjonsmodul.
- Få programmet til å kjøre i skjermvisning.

Delmål:

- Samle inn nødvendig informasjon om kommunikasjonsprotokoller fra tredemølle leverandør.
- Kunne tolke og skrive kommunikasjonsmoduler.
- Tilegne seg grunnleggende kunnskaper i Unity3D og andre verktøy som ble nødvendige for utvikling i prosjektet.

1.3 Begrensninger

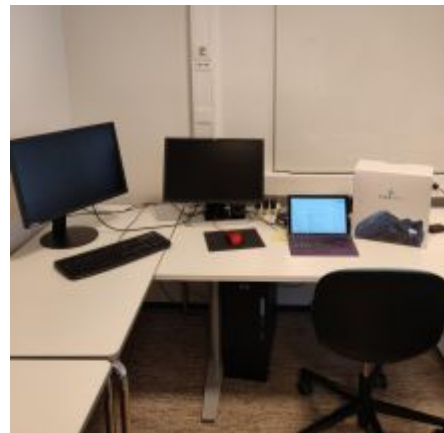
Gruppen har tilgang til to ulike tredemøller, Rodby RL2550E x 1000 og Woodway, for testing og sammenligning av moduler, hvor Rodby bruker Trackmaster-protokollen og Woodway bruker enkelte elementer av den. Tredemøllene som blir brukt til testing og avlesing er lokalisert på rehabiliteringslaboratoriet, som er ofte brukt av andre, og dermed ikke garantert tilgang til enhver tid.

1.4 Ressurser

Oppdragsgiver, Lars Peder Bovim, og intern veileder, Harald Soleim, har vært en stor støtte og gitt god veiledning på forskjellige områder. I tillegg har kontakt med den forhenværende gruppen vært en uvurderlig ressurs da det kom til å få oversikt over VR Walk sin oppbygging, og tidligere masterstudent i programutvikling Thomas Fiskeseth Larsen, som lagde modulen som blir brukt av VR Walk.

1.4.1 Arbeidsplass

Det ble reservert plass på Master-laben (E441) som kunne benyttes som arbeidsstasjon for gruppen ut prosjektets levetid. Ved reservasjon hadde gruppen tilgang til Fysio-laben (F327).



Figur 1.4.1: Arbeidsplass

1.4.2 Tilgjengelig utstyr

På fysio-laben stod det to tredemøller, hvor den ene ble brukt med programvaren fra det forrige prosjektet. Møllen som kunne kommunisere med VR Walk programmet var en Rodby RL2550E x 1000. VR Walk sin kommunikasjonsmodul var ikke kompatibel med styringsprotokollene som ble brukt i Woodway møllen.



Figur 1.4.2: VR Walk oppsett

Arbeidsstasjonen på Master-laben bestod av en kraftig stasjonær datamaskin, et par skjermer, datamus, og tastatur, alt var på utlån fra datafag ved HVL, for arbeid med prosjektet. Det var også ett sett med HTC Vive Pro VR-briller med sensorer som ble brukt som hjelpemiddel for utvikling i Unity 3D. Gruppemedlemmene hadde også personlige datamaskiner som ble benyttet.

1.4.3 Utviklingsmiljø

- IDE:
Visual Studio 2017 med MS DreamSpark Educational lisens.
- Engine:
Unity 3D Pro med Education lisens.
- Kodedeling:
GitHub Desktop med Educational konto for privat lagringsområde



Figur 1.4.3: HTC Vive utstyr

1.4.4 Programvare

Modulen som ble utviklet i det forrige prosjektet var en viktig ressurs som ble brukt til sammenligning av kommandoer og responser. For å analysere datastrømmen mellom tjener og klient (tredemølle/kontrollpanel) ble det brukt en pakke-sniffer for seriell port overgang. Gruppen gikk gjennom flere forskjellige programmer for å finne en som tilfredsstilte de to kravene de hadde stilt seg, kostnaden for programvaren måtte være lavest mulig og være lett å bruke. Gruppen endte opp med å ta i bruk Device Monitoring Studio.

1.5 Organisering av rapporten

- Kapittel 1** Presentasjon av mål og motivasjon til oppgaven, samt kontekst og relevans for oppdragsgiver. Gruppens begrensninger i forhold til å utføre prosjektet og oppnå målene. Presentasjon av de tilgjengelige ressursene for prosjektgruppen.
- Kapittel 2** Presentasjon av bakgrunnen til problemstillingen, bakgrunnen til arbeidsgiver, og tidligere arbeid.
- Kapittel 3** Presentasjon av mulige fremgangsmåter for å løse problemstillingen, spesifikasjoner fra oppdragsgiver, valg av verktøy og programmeringsspråk av oppdragsgiver.
- Kapittel 4** Presentasjon av arbeidet gjort i prosjektet.
- Kapittel 5** Presentasjon av evaluering, hvilken metoder benyttet og resultatet.
- Kapittel 6** Presentasjon av diskusjon rundt framgangsmåte og resultat.
- Kapittel 7** Presentasjon av konklusjon og videre arbeid.
- Kapittel 9** Presentasjon av risikoliste og Gantt diagram.

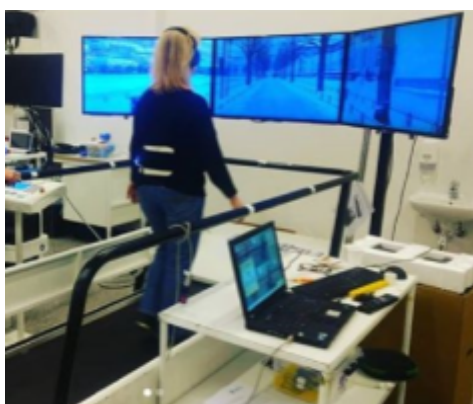
2 Prosjektbeskrivelse

2.1 Om oppdragsgiver

SimArena er en samlebetegnelse for 25 testlaboratorier. De har laboratorier for klinisk sykepleie, radiografi, fysioterapi og ergoterapi. Disse brukes til ferdighetstrening, veiledning, refleksjon, simulering og forskning. Rehabiliteringslab F327 er et tverrfaglig laboratorium som benyttes i forskning-, utvikling- og i undervisningssammenheng. Laben har i hovedsak vært benyttet innen fysioterapifag, men nå benytter også Ergoterapi, Radiografi og Data- og Realfag laboratoriet og tilhørende utstyr (HVL, 2018).

Lars Peder jobber som prosjektutvikler og høgskolelærer ved HVL og tar samtidig en masterutdannelse på UiB. I masterprosjektet ønsker han å forsøke VR-briller som en ekstra oppgave ved tredemøll trening. Under testene skal han undersøke endringer i gangmønstre under forskjellige oppgaver. Etter å ha utført målingene skal han rapportere de ulike scorene for å se gjennomførbarhet for den aktuelle gruppen (unge friske) og diskutere om dette er forsvarlig å prøve ut på en mer sårbar gruppe (eldre eller personer med funksjonsnedsettelse).

2.2 Tidligere arbeid



Figur 2.2.1: Videovisning

SimArena har utviklet et prosjekt hvor brukere går på en tredemølle og følger med på en video som spilles av på skjermer foran tredemøllen. Videoen som spilles av på skjermene simulerer en tur gjennom et bymiljø der brukeren blir stilt spørsmål om, for eksempel, hvilken farge personen som nettopp passerte hadde på jakken. Tidligere bachelorstudenter, Lauritz S.



Figur 2.2.2: Testing av VR Walk

Valved og Bendik M. Bleikli, har i samarbeid med SimArena videreutviklet prosjektet slik at det nå benytter VR briller og gir brukeren en mer interaktiv opplevelse i en virtuell verden. Spillet er bygget opp i flere sekvenser, der hver sekvens har ulike oppgaver. Sekvensene settes sammen i en sekvensbygger før man starter spillet. Som en forberedelse til VR Walk prosjektet, ble det utviklet en kommunikasjonsmodul som gjør det mulig å sende enkle kommandoer som start, sett hastighet og stopp til Rodby tredemøllen.

2.3 Initielle kravspesifikasjoner

- Det skal utvikles en kommunikasjonsmodul som kan kommunisere med andre tredemøller. Fokuser på tredemøller som bruker Trackmaster protokollen først for å få en liten gruppe og videreutvikle en mer generalisert modul derfra.
- Det skal utvikles en sømløs overgang fra VR-briller til monitorer, hvor sensorer fortsatt skal kunne lese bevegelsene til bruker og lese det av til korresponderende bevegelser på skjermen.
- Alt av utvikling skulle skje ved å benytte C# som programmeringsspråk, slik at det kan lettere integreres inn i VR Walk sin eksisterende kode.
- Den nye kommunikasjonsmodulen skulle kunne utføre de samme kommandoene som VR Walk kunne med Rodby-møllen.
 - Sett hastighet.
 - Sett vinkel på båndet.
 - Endre hastighet positivt eller negativt med en spesifikk mengde.
 - Endre akselerasjon.
 - Nødstopp.

2.4 Initialt løsningsforslag

I første omgang ville det blitt utviklet et kommunikasjons-script som kunne kommunisere med Woodway tredemøllen uavhengig av VR Walk. Etter først opprettet kontakt ville gruppen forsøke å kartlegge hvilke kommandoer og input-verdier som var nødvendige for å utføre forskjellige funksjoner, med tanke på kravspesifikasjonene som ble gitt av oppdragsgiver og VR Walk

3 Prosjektdesign

3.1 Mulig fremgangsmåter

3.1.1 Generell kommunikasjonsmodul

Utforme kommunikasjonsmodulen slik at den klarer å identifisere hvilken tredemølle som er koblet til, og deretter benytter seg av riktige kommandoer til den tilhørende tredemøllen.

3.1.2 Avatar i skjermvisning

Om brukeren skal benytte seg av skjermvisning, vil spillet vises i tredjeperson der en avatar imiterer bevegelsene til brukeren. På denne måten vil brukeren ha noe å henvende seg til i forhold til posisjonering på stien.

3.1.3 Fast vinkel mellom skjermene

Om brukeren skal benytte seg av skjermvisning, er det viktig at skjermene som brukeren skal henvende seg til er i samme posisjon og vinkel hver gang for å unngå visuelle avvik. Siden det er tre skjermer, vil det være en vinkel og et mellomrom der skjermene møtes. En mulig løsning vil være å benytte seg av 3d-printeren på HVL for å lage et feste som kan plasseres på eller mellom skjermene slik at mellomrommet og vinkelen blir den samme hver gang.

3.1.4 Automatisk justering av stibredde

Siden tredemøllene har ulike bredder må stibredden i programmet justeres ut fra hvilken tredemølle brukeren benytter seg av. En mulig løsning på dette kan være å justere bredden på stien etter at kommunikasjonsmodulen har identifisert hvilken tredemølle som er koblet til.

3.2 Spesifikasjon

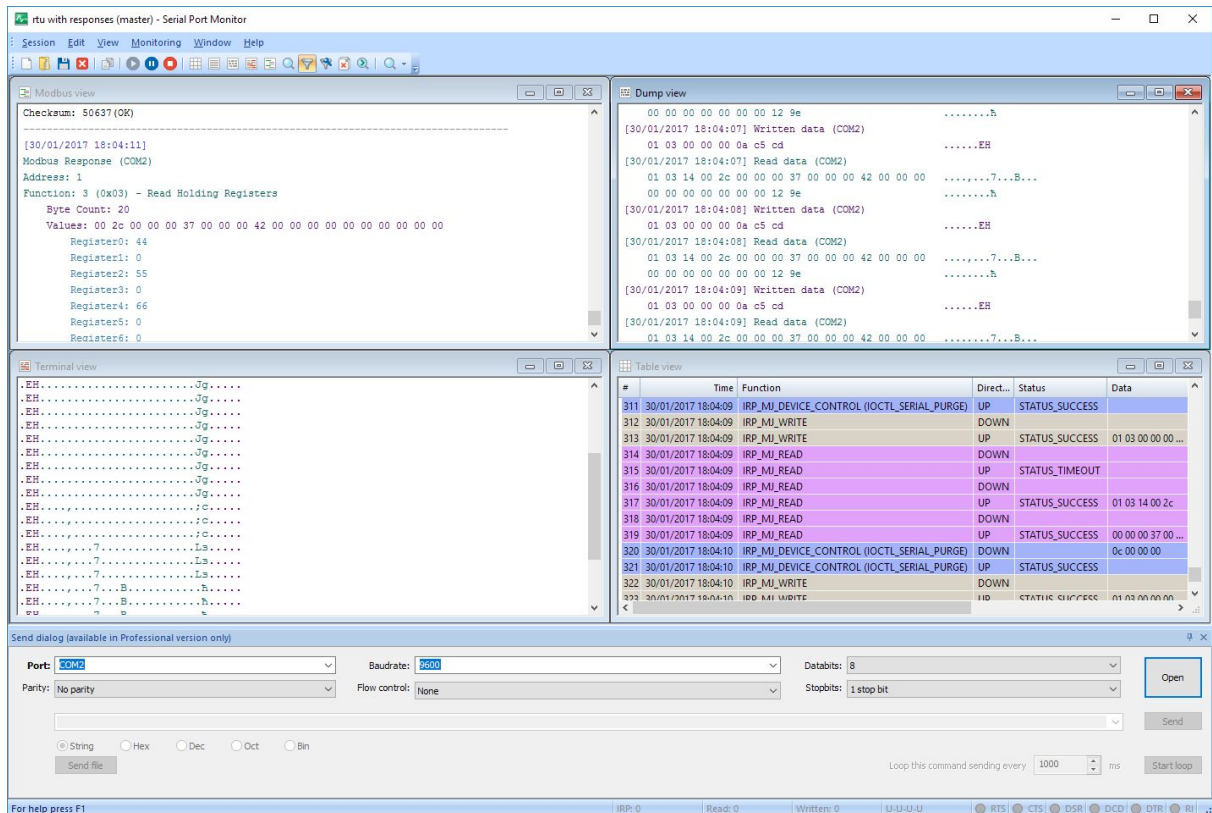
Kommunikasjonsmodulene skulle bli utviklet under trackmaster modulen. Når det gjelder sporing og VR utstyr vil HTC Vive benyttes, siden det kommuniserer bra med Unity 3D og vil gi en god tilnærming av posisjon til en avatar/bruker på tredemøllen.

3.3 Verktøy og programmeringsspråk

Programmet er utviklet i spillmotoren Unity 3D og Visual Studio IDE. For å ta backup, kodedeling, og versjonskontroll av programmet, har gruppen benyttet seg av GitHub Desktop i tillegg til private lagringsområder. Videre deling av bilder og filer ble gjort gjennom Google Drive og Google Docs.

Kommunikasjonsmodulen til tredemøllene er skrevet i programmeringsspråket C# grunnet at det er godt støttet i Unity 3D. Til visning av tekst i simulatoren ble TextMeshPro (Digital Native Studios, Unity Technologies, 2018) benyttet. Denne viste seg å være mer fleksibel, brukervennlig og egnet til VR-bruk enn Unity sin innebygde løsning for tekstvisning. Videre er SteamVR software benyttet for implementering av VR funksjoner.

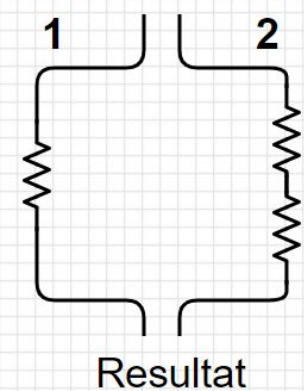
For lytting og analysering av pakketrafikken mellom klient (arbeidsstasjon) og tjener (tredemølle) ble sniffer-programmet, Device Monitoring Studio, valgt. Verktøyets funksjoner og oversikt tilfredsstilte gruppens behov, og påvirket ikke prosjektet økonomisk.



Figur 3.3.1: Skjermbilde fra pakkesniffer programvare

3.4 Prosjektutvikling

Problemstillinger



Utviklingen startet med å fokusere på de to hovedmålene, men etter at gruppen møtte på uforventede problemer med å få VR Walk programmet til å kjøre på den nyeste versjonen av Unity 3D, la oppdragsgiveren til et tilleggsmål om å ordne opp i problemet. Tilleggsmålet gikk ut på å ordne opp i alle feilmeldinger samt oppdatere programvaren slik at den ble kompatibel med nyeste versjon av Unity3D. Etter tilleggsmålet var utført, tok oppdragsgiveren en avgjørelse om at gruppen skulle

prioritere å opprette kontakt med Woodway tredemøllen fremfor videre utvikling av VR Walk programmet. Ved å benytte en serial port sniffer og Woodway Treadmill Control Software, har signalene blitt tolket, skrevet ned i en oversikt og implementert som byte arrays i et C# script slik at de benyttes som en asset i VR Walk.

3.5 Evalueringsmetode

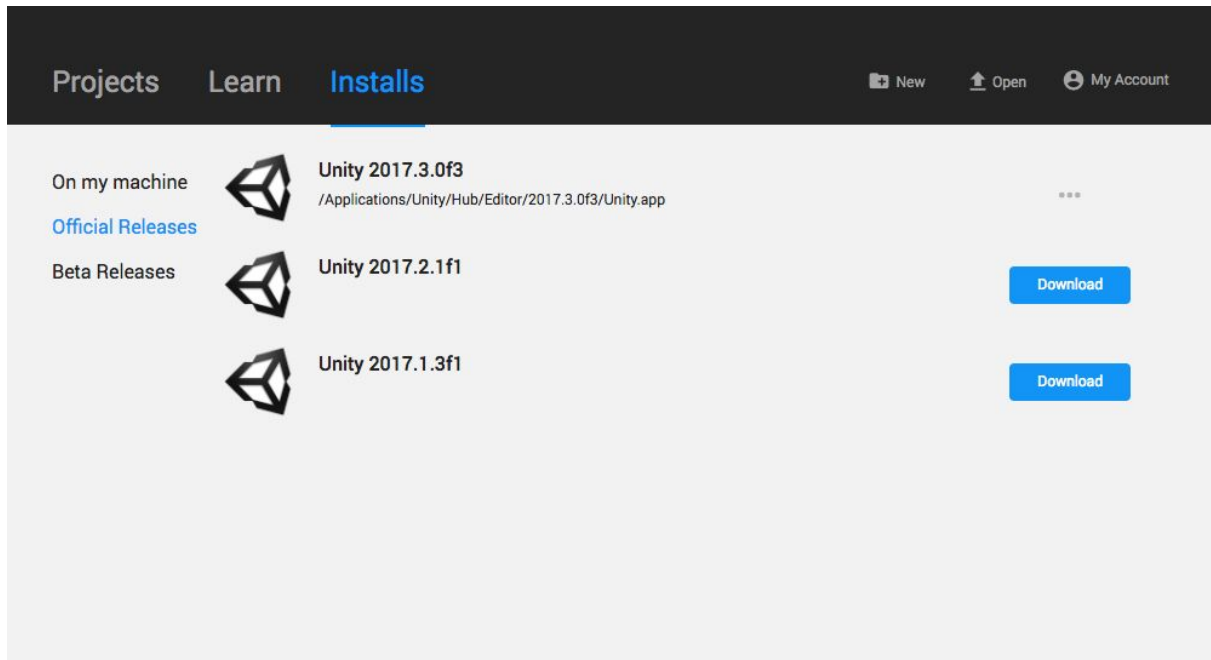
Evalueringen av arbeidet som har blitt utført vil bli foretatt mot slutten av prosjektet. Resultatene gruppen har kommet frem til vil være funksjonelle og dermed bli evaluert ut i fra funksjonalitet.

4 Løsning

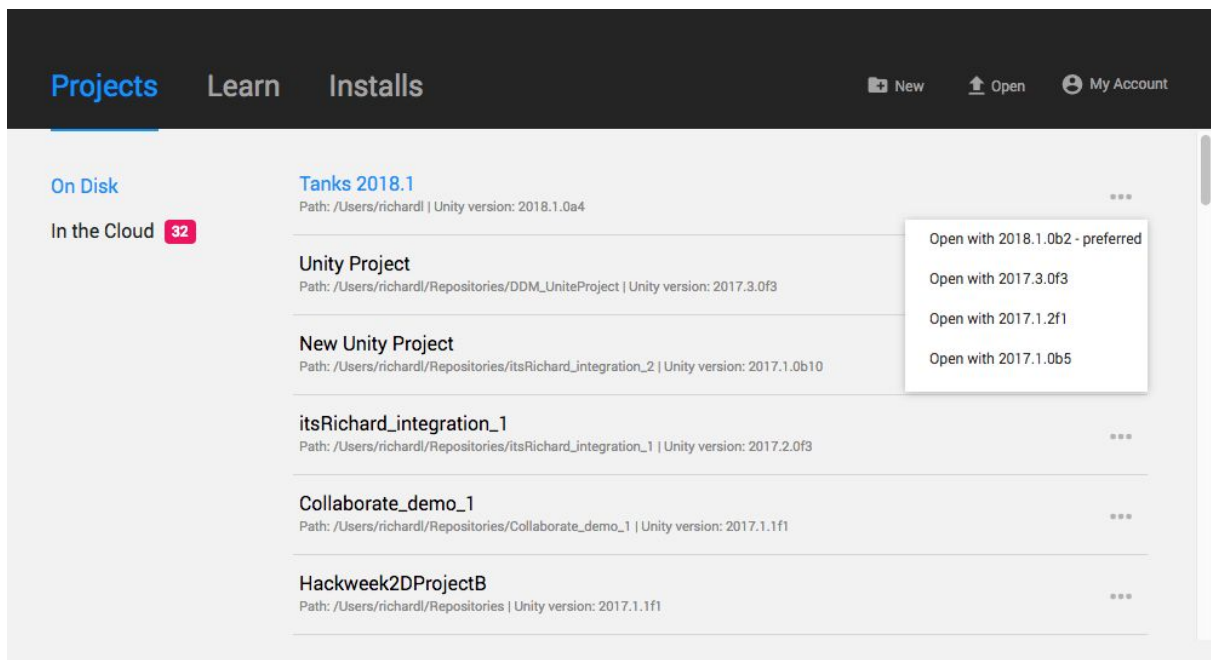
4.1 Kompatibilitet med oppdaterte versjoner

Prosjektet startet ved at vi ble presentert med program VR Walk, som ble utviklet i en utdatert versjon av editoren, Unity3D, som gruppen skulle benytte. Med tanke på at dette er et prosjekt som kommer til å videreutvikles over tid fikk gruppen en tilleggsoppgave om å sørge for at VR Walk vil kunne utvikles på nye versjoner av Unity3D editoren og fremdeles fungere som det har gjort før. For å gjøre det enklere å bevege seg frem og tilbake mellom nye og eldre versjoner, valgte vi å laste ned Unity Hub.

Unity Hub er et eget program som strømlinjeformer måten en finner, laster ned, håndterer Unity-prosjekter og installerer nye versjoner. Den lar en også oppdatere og holde de nye versjonene av editoren ved like på en enklere måte. Figur 4.1.1 er et utklipp fra Unity Hub og viser forskjellige oppdateringer til versjon Unity 2017 som kan lastes ned eller allerede er på maskinen. En kan åpne Unity-prosjekter i den versjonen en foretrekker av de som ligger lagret i Hub-en, eller sette preferanse for en spesifikk versjon. Figur 4.1.2 representerer oversikten over Unity-prosjektene som en har, om de er lagret på disk eller skylagring, og en liten meny hvor en kan bestemme hvilken versjon av Unity en ønsker å åpne prosjektet. Programmet automatiserer veldig mye av det som ellers måtte blitt gjort manuelt, f.eks. å flytte rundt på filer og mapper i filutforskeren.



Figur 4.1.1: Oversikt over Unity3D versjoner i Unity Hub.



Figur 4.1.2: Oversikt over Unity3D-prosjekter i Unity Hub.

Det å oppgradere fra versjon 2017.4.23f1 til 2018.3f presenterte seg som et stort problem i begynnelsen. For det første visste ingen av oss noe om hva som var nytt med (2018.3) og hvordan disse oppdateringene ville påvirke overgangen. Å flytte prosjektet direkte over fra versjon til versjon førte til en stor mengde feilmeldinger på grunn av et tillegg den forrige gruppen hadde benyttet seg av, som nå hadde blitt en del av Unity sitt standard programvare. Nettsiden til Unity inneholder den informasjonen en måtte trenge over hvilke nye funksjoner som er lagt til, (<https://unity3d.com/>).

Et annet problem som kom opp på grunn av den nye 'Package Manager' funksjonen, som ble lagt til i Unity 2018.3, gjorde at assets fra 'Asset Store', mer spesifikt PostProcessing, fikk problemer. Feilmeldingen som kom opp var,

```
Assets/Vendor/PostProcessing/Editor/PropertyDrawers/MinDrawer.cs(6,34): error CS0104:
'MinAttribute' is an ambiguous reference between 'UnityEngine.PostProcessing.MinAttribute'
and 'UnityEngine.MinAttribute'
```

Unity sin 'Asset Store' har en svakhet ved at det ikke går an å oppdatere eller sette versjoner som foreldet, som også er grunnlaget for at Package Manager ble introdusert i utgangspunktet. Både Unity og denne pakken legger symboler til felles globale namespace, og som i dette tilfellet kan det derfor forekomme «kollisjoner», her i referansene. Hva gruppen kom frem til var tre enkle måter å ordne feilen på. Første var å endre typereferansen til variabelen direkte i koden (figur 4.1.3, linje 11),

```
1  using UnityEngine;
2  using UnityEngine.PostProcessing;
3
4  namespace UnityEditor.PostProcessing
5  {
6      [CustomPropertyDrawer(typeof(UnityEngine.PostProcessing.MinAttribute))]
7      sealed class MinDrawer : PropertyDrawer
8      {
9          public override void OnGUI(Rect position, SerializedProperty property, GUIContent label)
10         {
11             UnityEngine.PostProcessing.MinAttribute attribute = (UnityEngine.PostProcessing.MinAttribute)base.attribute;
12         }
13     }
14 }
```

Figur 4.1.3

```

1  using UnityEngine;
2  using UnityEngine.PostProcessing;
3
4  namespace UnityEditor.PostProcessing
5  {
6      [CustomPropertyDrawer(typeof(MinAttribute))]
7      sealed class MinDrawer : PropertyDrawer
8      {
9          public override void OnGUI(Rect position, SerializedProperty property, GUIContent label)
10         {
11             MinAttribute attribute = (MinAttribute)base.attribute;
12         }
13     }
14 }

```

Figur 4.1.4

En annen metode en kunne benytte for å ordne problemet var å legge til en enkelt linje på toppen av koden i figur 4.1.5,

```

1  using MinAttribute = UnityEngine.PostProcessing.MinAttribute;
2  using UnityEngine;
3  using UnityEngine.PostProcessing;
4
5  namespace UnityEditor.PostProcessing
6  {
7      [CustomPropertyDrawer(typeof(MinAttribute))]
8      sealed class MinDrawer : PropertyDrawer
9      {
10         public override void OnGUI(Rect position, SerializedProperty property, GUIContent label)
11         {
12             MinAttribute attribute = (MinAttribute)base.attribute;
13         }
14     }
15 }

```

Figur 4.1.5

Løsningene ble funnet i en tråd på Github, hvor andre hadde støtt på samme problemet.

4.2 Tilknytning til Woodway tredemølle



Figur 4.2.1: Woodway tilkoblingspanel



Figur 4.2.2: Seriell port kabel

Alt av kommunikasjon som går gjennom grensesnittet fra tredemøllen (figur 4.2.1) til datamaskinen, går gjennom en RS-232 9-pin seriell port. Den første utfordringen bachelorgruppen møtte på under utviklingen av et kommunikasjons-script, var å koble seg til tredemøllen. For å koble datamaskinen til tredemøllen benyttes en seriell port kabel (figur 4.2.2) med en seriell port til usb overgang på datamaskinenden. Seriell porten tillater kommunikasjon begge veier, som vil si at enheter kan sende og motta data gjennom tildelte pinner i tilkoblingen. Seriell porten tar en byte med data, som inneholder 8 bits, og overfører én bit om gangen (derav navnet seriell). Før hver byte som blir overført, sendes det en start bit etterfulgt av en byte med data og en stop bit som signaliserer slutten av overføringen. Når man sender data til en seriell port, går den gjennom en kontroll chip (Universal Asynchronous Receiver/Transmitter) som har en innebygget buffer. Denne bufferen tillater å cache data som kommer inn fra system bus når data som går ut blir prosessert. De 9-pinnene i kabelen har følgende oppgaver:

1. **Data Carrier Detect** - Beslutter om det er en aktiv tilkobling.
2. **Receive Data** - Datamaskinen mottar data.
3. **Transmit Data** - Datamaskinen sender data.
4. **Data Terminal Ready** - Datamaskinen forteller at den er klar til å starte en kommunikasjon.
5. **Signal Ground** - Pinnen er jordet.
6. **Data Set Ready** - Mottar klar til å starte kommunikasjon signal.
7. **Request To Send** - Datamaskin spør om tillatelse til å sende informasjon.
8. **Clear To Send** - Mottar klarsignal til å starte overføring.
9. **Ring Indicator** - Signal for at ring er oppdaget.

Etter mye søking av dokumenter om maskinvareinformasjon til forskjellige tredemøller, både kommersielle og innen medisin, kom gruppen frem til en konklusjon om hva som gjorde at Woodway-møllen ikke responderte. Den mest usikre innstillingen var hvilken baud rate som skulle bli benyttet. Baud rate er en innstilling som angir hvilken hastighet informasjon overføres. Rodby tredemøllen opererer med en baudrate på 4800 som vil si at den er kapabel til å overføre 4800 bits per sekund. Port-verdiene som blir brukt for Woodway tredemøllen er som følger:

- BaudRate: 9600
- Databits: 8
- Parity: no parity
- Stop Bit: 1 stop bit

Ved bruken av den oppgitte innstillingen og uten en handshake prosedyre, vil tilkoblingen bli vellykket.

4.3 Signal analyse

For å muliggjøre en kommunikasjon mellom VR-Walk og Woodway tredemøller, har det vært nødvendig å identifisere hvilke signaler som sendes fra tredemøllen til pc-en, og hvilke kommandoer som blir akseptert av Woodway tredemøllen. Gruppen har benyttet seg av Device Monitoring Studio serial port sniffer for å overvåke all datatrafikk som går gjennom seriell porten ved testing.

Etter møtet med Thomas Fiskeseth Larsen fikk vi tilsendt over en e-post, den 22.februar 2019, med de dokumentene som han tok utgangspunkt i for å utvikle kommunikasjonsmodulen som ble brukt for Rodby-møllen. Det viste seg at Rodby-møllen var kompatibel med Trackmaster-protokollen, som hadde noen likheter med Woodway sin kontrollprotokoll, noe vi ved et tilfelle kom over under testing. I begynnelsen av testingen undersøkte gruppen hva for responser som ble sendt tilbake over seriell-porten etter å ha sendt en kommando gjennom VR Walk programmet, og om svaret var noe som var forventet eller ikke. Gruppen oppdaget fort at selv om noen av responsene stemte over med dokumentasjonen som de hadde fått utgitt, kom det ingen fysisk reaksjon fra møllen. En annen ting som ble lagt merke til var at de to tredemøllene kommuniserte med forskjellig baudrate. Etter å ha endret port-innstillingene i koden til VR Walk, slik at det stemte over med spesifikasjonene til Woodway-tredemøllen, ble det sendt andre tilbakemeldinger enn hva det hadde vært før. Dette var fortsatt i det stadiet hvor gruppen ikke hadde funnet ut hvordan og hvilke data som måtte sendes over tilkoblingen for å kommunisere. Så fort gruppen hadde gjennomgått hvilke kommandoer/tilbakemeldinger som ble sendt via VR Walk, gikk de over til å bruke Woodway sin programvare for kontroll av tredemøllen (Woodway, 2014), og forsøkte å kartlegge og sammenligne hvilke meldinger som kom opp. Dette ble gjort i et tidsrom overkant av to uker, uten å få noen betydelige resultat. Vi hadde satt oss fast uten nok informasjon om hvilken protokoll Woodway-møllen kjørte med, og måtte søke om råd fra leverandøren.

Oppdragsgiver sørget for kontakt med leverandør og via dialog over e-post ble vi tilsendt en del utvalgte PDF 'er.. PDF 'ene inneholdt informasjon om protokoller, lik Trackmaster-protokollen som ble gitt av den tidligere gruppen, men listen over kommandoer var forskjellig for hver protokoll. Dette skapte mye forvirring, men førte også til at det ble gjort et valg som førte til at gruppen gikk vekk fra å bruke den forrige gruppens modul til å kommunisere og utviklet en helt egen for å kunne lettere bestemme informasjonen i hele datapakken som ble sendt. Etter en del testing og kartlegging av hvilke datapakker som ofte dukket opp under analysering, og konsultasjon med Thomas Fiskeseth Larsen, fikk vi endelig opprettet kontakt med møllen og laget til et kontroller-script med noen enkle metoder for testing.

| Kommando | Hex | Parameter (hex) | Oversatt hex |
|--|-----|-----------------|--------------|
| Sett timeout | AE | 30 30 | 0.0 |
| Protokoll bytte til kontroller | A2 | 30 30 | |
| Slå på tredemøllen | A6 | 31 | 1 / true |
| Protokoll bytte til standard protokoll | A5 | - | Null |
| Gjør møllen mottakelig for ordre | A9 | - | Null |
| Sett helling | A4 | | |
| Sett akselerasjon | A5 | | |
| Sett fart | A3 | | |
| Stop møllen/Ikke mottakelig for ordre | A8 | - | Null |
| Protokoll bytte til kontroller | A2 | 30 30 | |
| Skru av tredemøllen | A6 | 30 | 0 / false |

(Tabell for kommunikasjon med Woodway-tredemølle. Tabellen viser en oversikt over kommandoer og de korresponderende hex-verdiene som brukes, samt eksempel på parameter og hva de betyr.)

Det kan være nyttig å ha en ASCII tabell ved siden av når en skal skrive inn verdiene som skal inn i kommandoene om en ikke kan konverteringen fra Hexadecimal til standard karakterer.

Kommandoene kan slås sammen for å gjøre flere ting samtidig. Eksempelet under er en hel datapakke/byte array som blir sendt som en kommando:

```
"A5 30 31 32 30 0D A3 30 35 34 30 0D A4 30 30 35 37 30 0D"
```

Sett akselerasjon Sett hastighet Sett stigning

I eksempelet slår vi sammen kommandoene for å sette akselerasjonen til 1.2m/s^2 , sette farten på møllen til 0.54 m/s (540 mm/s), og hellingen til 5.7% stigning. En viktig detalj å få med seg er at fart blir lest som **mm/s** av maskinen og en blir derfor nødt å gjøre konverteringen til km/t utenom for å få ønsket hastighet. En annen ting er å ende alle kommandoer med stopbit-et **0D** (understreknet i eksempelet). En mer detaljert gjennomgang kan finnes i vedlegg 9.3.

I scriptet ble det valgt å slå sammen de fire første kommandoene i en og samme datapakke for å ha en renere kode og raskere implementering (se Figur 4.6, første kommando). Dette var også anbefalt i dokumentet *PPS/Loko treadmill rs-232 summary*, som gruppen fikk den 6.mai fra kontakten i Woodway.

4.4 Utvikling av kommunikasjons script

Gjennom programmeringsspråket C# har det blitt utviklet et script som er lagt opp slik at det kan bli benyttet som en asset i Unity programmet VR Walk. Videre gir scriptet muligheten til å sende ønskede kommandoer over til tredemøllen. På denne måten kan man kartlegge kommandoer og responser slik at tredemøllen har en ønskelig oppførsel. For å teste kommandoene som har blitt identifisert, ble signalene sendt direkte til tredemøllen gjennom Command Prompt. Scriptet opererer uavhengig av VR Walk og Unity, for å minimere sjansen for en utilsiktet endring av signalet.

```
using System;
using System.IO.Ports;
using System.Text;
using System.Threading;

0 references
public class KommunikasjonsScript
{
    static bool _continue;
    static SerialPort _serialPort;

    0 references
    public static void Main()
    {
        string kommando;
        StringComparer stringComparer = StringComparer.OrdinalIgnoreCase;
        Thread readThread = new Thread(Read);

        // Lager et nytt port-objekt med default settings
        _serialPort = new SerialPort();

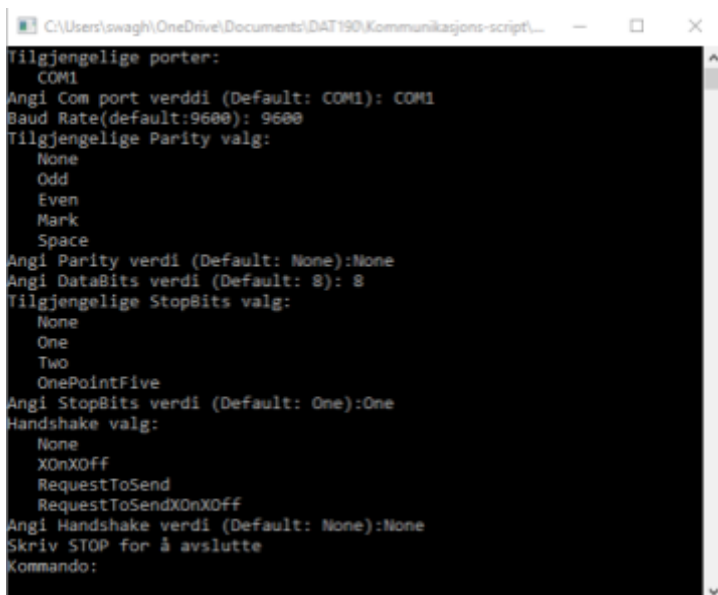
        // Setter seriell port verdier
        _serialPort.PortName = SetPortName(_serialPort.PortName);
        _serialPort.BaudRate = SetPortBaudRate(_serialPort.BaudRate);
        _serialPort.Parity = SetPortParity(_serialPort.Parity);
        _serialPort.DataBits = SetPortDataBits(_serialPort.DataBits);
        _serialPort.StopBits = SetPortStopBits(_serialPort.StopBits);
        _serialPort.Handshake = SetPortHandshake(_serialPort.Handshake);
    }
}
```

Figur 4.4.1: Bruk av System.IO.Ports

representerer seriell port ressurser og gir brukeren et rammeverk for å koble seg til og kommunisere gjennom porten. Ved bruk av SerialPort klassen, kan man opprette et seriell port-objekt, gi objektet ønskede verdier, åpne seriell porten for kommunikasjon og sende data.

Scriptet ble utviklet i programmeringsspråket C# gjennom utviklingsverktøyet Visual Studio. For å at brukeren skal kunne kommunisere gjennom seriell porten, benytter scriptet seg av System.IO.Ports namespace. System.IO.Ports inneholder klasser for kontrollering av seriell porten. Den viktigste klassen, SerialPort,

Når man kjører scriptet vil det åpne et Command Prompt vindu, der brukeren blir bedt om å angi port-verdiene som vist i figur 4.4. Når ønskede innstillinger er angitt, kan brukeren skrive inn kommandoer. Når en bruker skriver inn en kommando, vil



```
C:\Users\swagh\OneDrive\Documents\DAT190\Kommunikasjons-script\...
Tilgjengelige porter:
COM1
Angi Com port verdi (Default: COM1): COM1
Baud Rate(default:9600): 9600
Tilgjengelige Parity valg:
None
Odd
Even
Mark
Space
Angi Parity verdi (Default: None):None
Angi DataBits verdi (Default: 8): 8
Tilgjengelige StopBits valg:
None
One
Two
OnePointFive
Angi StopBits verdi (Default: One):One
Handshake valg:
None
XOnXOff
RequestToSend
RequestToSendXOnXOff
Angi Handshake verdi (Default: None):None
Skriv STOP for å avslutte
Kommando:
```

Figur 4.4.2: Seriell port innstillinger i Command Prompt

scriptet se om kommandoen er lik en av flere forhåndsdefinerte kommandoer. Om kommandoen er lik, vil den tilhørende byte arrayen bli sendt gjennom seriell porten. Mellom hver byte som blir sendt, vil tråden bli bedt om å vente i 40 millisekunder for å unngå at for mye informasjon blir sendt samtidig (Figur 4.6). Kommandoer som er lagt til er:

- stop: setter farten til 0 m/s (stopped båndet).
- off: skruer av tredemøllen.
- walk: setter båndhastigheten til 0.52 m/s (1.9 km/t).
- jog: setter båndhastigheten til 1.4 m/s (5 km/t).
- run: setter båndhastigheten til 2.77 m/s (10 km/t).

- on: skruer på møllen og setter den i standard protokoll.

```

while (_continue)
{
    Console.WriteLine("Kommando: ");
    kommando = Console.ReadLine();

    if (stringComparer.Equals("off", kommando))
    {
        foreach (byte[] c in off)
        {
            _serialPort.Write(c, 0, c.Length);
            _serialPort.BaseStream.Flush();
            Thread.Sleep(40);
        }

        _continue = false;
    }
}

```

Figur 4.4.3: Eksempel på kode som sjekker hvilken kommando du kaller på

```

// Diverse byte arrays som gir ønskede responser
// Skruer på tredemøllen og sette den i standard protokoll
byte[][] on =
{
    new byte[] { 0xAE, 0x30, 0x30, 0x0D, 0xA2, 0x30, 0x30, 0x0D, 0xA6, 0x31, 0x0D, 0xA5, 0x0D }
};

// Slår av tredemøllen
byte[][] off =
{
    new byte[] { 0xA2, 0x30, 0x30, 0x0D },
    new byte[] { 0xA6, 0x30, 0x0D }
};

// Stopper båndet
byte[][] stop =
{
    new byte[] { 0xA9, 0x0D },
    new byte[] { 0xA8, 0x0D }
};

// Setter akselerasjon til 0.01m/s^2 og hastighet til 0.52m/s
byte[][] walk =
{
    new byte[] { 0xA9, 0x0D },
    new byte[] { 0xA5, 0x30, 0x30, 0x31, 0x30, 0x0D },
    new byte[] { 0xA3, 0x30, 0x35, 0x32, 0x30, 0x0D }
};

// Setter hastighet til 1.4 m/s
byte[][] jog =
{
    new byte[] { 0xA9, 0x0D },
    new byte[] { 0xA3, 0x31, 0x34, 0x30, 0x30, 0x0D }
};

// Setter hastighet til 2.77m/s
byte[][] run =
{
    new byte[] { 0xA9, 0x0D },
    new byte[] { 0xA3, 0x32, 0x37, 0x37, 0x30, 0x0D }
};

```

Figur 4.4.4: Byte arrays

4.5 Metode for å sette hastighet

I C# scriptet har det blitt laget en metode for å sette hastighet. Når man kaller på metoden, sender man med en numerisk parameter. Denne parameteren vil være den ønskede hastigheten en vil sette båndet på tredemøllen til, målt i km/t.

Woodway tredemøllen kan kun sette hastighet i form av mm/s, noe som fører til at metoden først må konvertere km/t til mm/s. Etter farten er i riktig format, kan metoden opprette en byte som videre vil bli sendt til tredemøllen. For å opprette byte-et i form av en hastighet kommando, må det starte med 0xA3 etterfulgt av fire bit som skal inneholde fartsdata. Eksempelvis om man skal sette hastigheten til 2.77mm/s:

- 0xA3, 0x32, 0x37, 0x37, 0x30, 0x0D

Her ser vi A3 kommandoen etterfulgt av 32, 37, 37 og 30 som blir tolket som 2,770mm/s og avsluttes med en stopbit 0D.

5 Evaluering

5.1 Evalueringsmetode



Figur 5.1.1: Testing av kontroll-script

I løpet av prosjektperioden, har målene endret seg en del. Ved et samarbeid mellom bachelorgruppen og oppdragsgiver, har prioriteringslisten over målene flyttet seg vekk fra det visuelle aspektet av oppgaven. Fokuset har blitt lagt på tilretteleggelse og vedlikehold for videreutvikling, og funksjonell kommunikasjon. Det har vært regelmessige møter med oppdragsgiver hvor gruppen utfører funksjonelle tester på fysio-lab. Evalueringsmetoden bestod av funksjonell testing der kommandoer ble sendt

til tredemøllen og responsen ble vurdert. Store deler av prosjektets levetid, hadde gruppen problemer med å få kontakt med tredemøllen og det var ikke før slutfasen at gruppen kunne presentere funksjonelle resultater til oppdragsgiver og veileder.

5.2 Evalueringresultat

Oppdragsgiveren sine ønsker var at det skulle være mulig å gi tredemøllen kommandoer som start/på, set hastighet, heving av båndet og stopp/av. Når det gjelder sending av kommandoer med riktig respons, har gruppen oppnådd de ønskelige resultatene. Videre var målet å veve inn scriptet som en asset i Unity programmet VR Walk, men grunnet trangt med tid, har scriptet ikke blitt implementert videre. Derimot er scriptet lagt opp for at det enkelt skal kunne benyttes som en asset gjennom Unity. Resultatene har blitt demonstrert for oppdragsgiver gjennom funksjonell testing ved fysio-lab.

Bachelorgruppen har oppdatert programvaren VR Walk til å være kompatibel med nyeste versjon av Unity3D. Dette var en uforventet oppgave som oppstod på grunn av utviklingen av VR Walk programmet tok sted i en utdatert Unity versjon. Arbeidet ble utviklet på en egen datamaskin på master-lab og senere oppdatert på arbeidsstasjonen på fysio-lab slik at masterstudentene Bendik og Lauritz fikk tilgang til den oppdaterte versjonen.

6 Diskusjon

6.1 Endring i prioritering

Underveis i prosjektet la oppdragsgiver til en ny problemstilling, å oppdatere programmet slik at det er kompatibelt med Unity sine nye versjoner, fra og med versjon 2017.4.23f1 til 2019.1.1f1 og eventuelle nye utgivelser. Dette er noe som ble utelatt av den forrige gruppen grunnet tidspress av å få et ferdig produkt som kunne fremvises. For å gjøre det enklere å holde Unity oppdatert bruker vi programmet Unity Hub.

6.2 Risikohåndtering ved endring

Etter en tid, i de senere stadiene av utviklingsperioden, oppdaget vi at oppgaven som foretok seg kommunikasjon mellom programvare og tredemøller kom til å bli en veldig utfordrende oppgave som vi kom til å bruke lang tid på for å muligens komme til en løsning. Som resultat av dette har vi, etter drøfting og forklaring av situasjonen, og godkjenning samt enighet fra oppdragsgiver, kuttet ut problemstillingen hvor vi skulle flytte VR aspektet til skjermvisning med sensor avlesning.

6.3 Fremgangsmåte

6.3.1 Utvikling av kommunikasjons-script

For at scriptet skulle kunne brukes videre, var det mest gunstig at det ble utviklet i programmeringsspråket C#. Før utviklingen startet, måtte gruppen tilegne seg kunnskap om programmeringsspråket grunnet at ingen av gruppemedlemmene hadde noe særlig erfaring fra språket C#. Den første fremgangsmåten var å manipulere det allerede eksisterende scriptet som fungerer med Rodby tredemøllen slik at det kunne sende kommandoer til Woodway i tillegg. Denne fremgangsmåten

skapte problemer da det eksisterende scriptet var dypt integrert i VR-Walk Unity programmet. Det viste seg å være vanskelig å endre på scriptet da en endring forårsaket flere feilmeldinger. Videre hadde ikke gruppen vært med å utvikle programmet og hadde dermed vanskeligheter for å se hvordan klassene hengte sammen. Gruppen tok da en avgjørelse om å utvikle et script som opererer uavhengig av VR-Walk og Unity. Målet med dette scriptet var først å opprette kontakt med tredemøllen, noe som hadde vist seg å være vanskelig, og kartlegge hvilke kommandoer som kan benyttes for å styre tredemøllen. Scriptet ble utviklet for å være så enkelt som mulig og for at det kan videre bli implementert som en asset i VR Walk.

6.3.2 Videreutvikling av VR-Walk

Gruppen fikk tilgang til en datamaskin med nyeste versjon av Unity. Da utviklingen skulle starte viste det seg at VR-utstyr måtte være tilkoblet for å kunne kjøre programmet. Gruppen tilegnet seg VR-utstyr og ble da møtt med en lang liste feilmeldinger som oppstod på grunn av at programmet ble utviklet i en eldre versjon av Unity. Gruppen hadde ingen tidligere erfaring med Unity og brukte derfor mye tid på å luke ut alle feilmeldingene. Når oppdateringsprosessen var fullført, kom oppdragsgiver med et ønske om at gruppen skulle fokusere på utviklingen av et kommunikasjons-script.

7 Konklusjon og videre arbeid

7.1 Måloppnåelse

I løpet av prosjektperioden har det blitt gjort forarbeid og tilretteleggelse for videre utvikling av VR-Walk prosjektet sin neste iterasjon. Kravet om å gjøre det gamle programmet kompatibelt med Unity3D sine nyeste oppdateringer er oppfylt. Videre er det kartlagt hvilke data og verdier som er nødvendig for å kommunisere med Woodway-tredemøllen. Gruppen har laget en veiledning som viser hvordan resultatene kan bli benyttet i videre utvikling. Problemstillingen om flytting fra VR-headset til monitor kom ikke lenger enn til idémyldringsfasen hvor vi gjorde oss noen tanker om forskjellige løsninger som kan bli brukt, står mer detaljert om dette i 3.1.2 og 3.1.3.

7.2 Kjente feil og begrensninger

Scriptet som har blitt utviklet av gruppen kan kommunisere feilfritt med Woodway-tredemøllen, men har ikke blitt videre implementert i VR Walk programvaren. Per nå har scriptet de samme funksjonene som Woodway Treadmill Control Software, styringsprogrammet som kommer med tredemøllen, men i et format som kan bearbeides til å ha verdi for videre arbeid.

Det er ikke laget til en løsning for inkrementell økning av hastighet. Dette skyldes for liten tid til å eksperimentere med konvertering av informasjonen som blir sendt mellom VR Walk programmet og tredemøllen. Videre er dokumentasjonen, og kommentarene i koden er kun norsk-språklig, men koden er på engelsk.

7.3 Implikasjoner og videre arbeid

Det tok lang tid for gruppen å lete etter informasjon og komme frem til en mulig løsning på kommunikasjonsproblemet. Det at gruppen ikke søkte etter assistanse fra produsenten før midten av utviklingsfasen gjorde at mye unødvendig tid ble brukt på research av problem som hadde lite relevans til sluttresultatet. Mens produsenten av tredemøllen som ble brukt av VR Walk, Rodby, er veldig åpne og imøtekomende til endringer utenfra på møllene sine, er Woodway det motsatte.

Arbeidet som har blitt gjort av gruppen, vil bli brukt videre av den forrige bachelor-gruppen som originalt utviklet VR Walk. Deres masteroppgave går ut på å videreutvikle VR Walk ved å legge til flere funksjoner og gjøre det mer motiverende for brukerne. Videre må det skrives kode i VR Walk programmet som benytter seg av sett hastighet metoden i C# scriptet når en Woodway tredemølle skal tas i bruk.

8 Litteratur/Referanser

8.1 Veiledninger/Litteratur

Microsoft (2018). *System.IO.Ports Namespace*. Tilgjengelig fra:

<<https://docs.microsoft.com/en-us/dotnet/api/system.io.ports?view=netframework-4.8>
> [Lest 9 May 2019].

Unity Technologies (2018) *Unity 2018.3*. Tilgjengelig fra:

<<https://unity3d.com/unity/whats-new/unity-2018.3.0/>> [Lest 26.mars 2019]

Stephan_B/Unity Technologies (2018) *Migrating from Legacy / Asset Store to Package Manager Releases*. Last edited: Fredag 3.mai 2019 klokken 09:46.

Tilgjengelig fra:

<<https://forum.unity.com/threads/migrating-from-legacy-asset-store-to-package-manager-releases.511748/>> [Lest 26.mars 2019].

rhys-vdw (2018) *'MinAttribute' is an ambiguous reference between*

'UnityEngine.PostProcessing.MinAttribute' and 'UnityEngine.MinAttribute' #725.

Last edited: lørdag 28.april 2019. Tilgjengelig fra:

<<https://github.com/Unity-Technologies/PostProcessing/issues/725>> [Lest 28.mars 2019].

Microsoft (2018) *SerialPort Class*. Tilgjengelig fra:

<<https://docs.microsoft.com/en-us/dotnet/api/system.io.ports.serialport?view=netframework-4.7.2>> [Lest 22.mars 2019]

Woodway (2014) *Woodway Treadmill Control Software*. Tilgjengelig fra:

<[http://puresports.hu/uploads/files/WOODWAY/Medical/PPS_Software_ENGLISH\(1\).pdf](http://puresports.hu/uploads/files/WOODWAY/Medical/PPS_Software_ENGLISH(1).pdf)> [19.mai 2019].

8.2 Verktøy

Unity Technologies (2018). Unity 3D. Besøkt 15.05.2019

Hentet fra: <https://unity3d.com/>

Visual Studio IDE. Visual Studio. Besøkt 15.05.2019

Hentet fra: <https://visualstudio.microsoft.com/>

HHD Software. Device Monitoring Studio. besøkt 02.06.2019

Hentet fra: <https://www.hhdsoftware.com/>

8.3 Bilder og figurer

Figur 2.1: *Treskjerm visning av video som beveger seg i bymiljø. Bildet av SimArena, hentet fra: https://www.instagram.com/p/BeV1mlcFT_3/?taken-by=hvlrehablab*

Figur 4.1.1 *hentet fra:*

<https://blogs.unity3d.com/2018/01/24/streamline-your-workflow-introducing-unity-hub-beta/>

Figur 4.1.2 *hentet fra:*

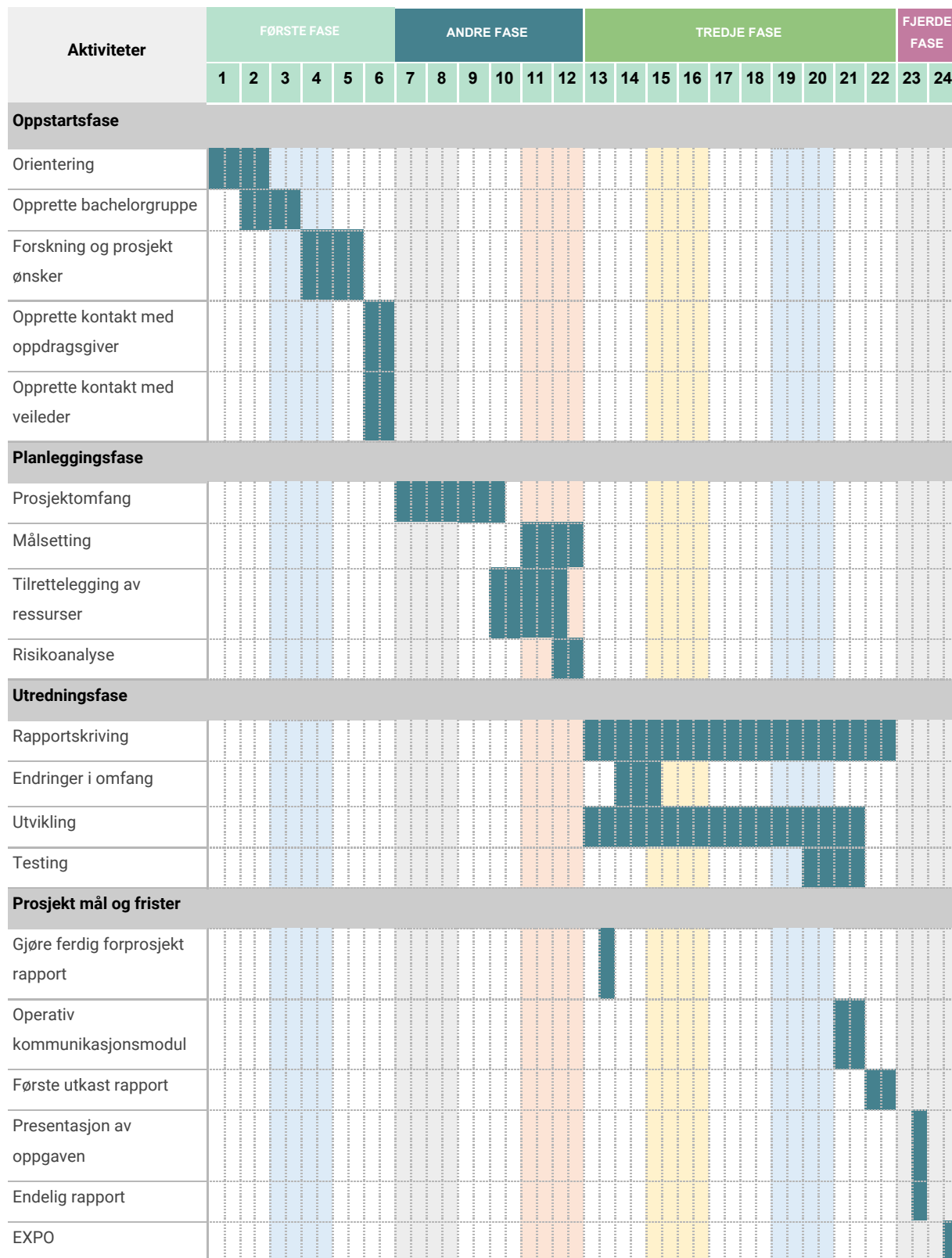
<https://blogs.unity3d.com/2018/01/24/streamline-your-workflow-introducing-unity-hub-beta/>

9 Vedlegg

9.1 Risiko liste

| Risiko | Alvorlighetsgrad | Sannsynlighet | Mulig løsning |
|---|------------------|---------------|---|
| Utstyr blir ødelagt. | Høy | Lav | Alltid ha oversikt over hvor utstyr er og bruke det riktig. Ha kontinuerlig backup av programvare.. |
| Langtidssykdom. | Middels | Lav | Oversikt over ting som skal gjøres og forventet tidsbruk |
| Endringer i kravspesifikasjon. | Middels | Middels | God dialog med oppdragsgiver, og programmere løsninger som er enkle å tilpasse. Å kunne si nei til endringer som fører til stort tap av tid. |
| Misforståelser i forhold til kravspesifikasjon. | Høy | Middels | Tett dialog med oppdragsgiver. |
| Ikke oppnå ønsket produkt pga. dårlig prioritering av tidsbruk. | Middels | Middels | Være bevisst på hva målet for sluttresultatet er og prioritere deretter. Holde milepælene. |
| For sjelden tilgang på nødvendig utstyr. | Middels | Lav | Reservere lab-rom med utstyr i god tid. Avtale faste arbeidstider. Benytte tiden på oppgaver som ikke krever tilgang på utstyret. Dersom absolutt nødvendig, benytte sen ettermiddag. |

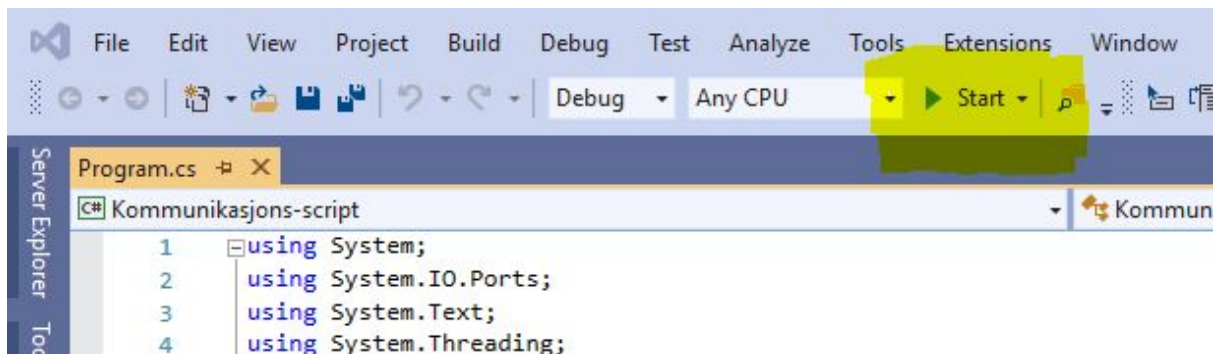
9.2 Gantt diagram



9.3 Veiledning for bruk og oppsett av kommandomodul.

9.3.1 Bruk av Kommunikasjons-script

- Åpne filen Kommunikasjons-script.sln i Visual Studio Solution.
- Trykk start knappen for å kjøre scriptet.



- Et CMD vindu vil åpne seg der du nå kan sette ønskede innstillinger. For Woodway tredemøllen er alt default settings bortsett fra COM port nummeret. De tilgjengelige port valgene vil komme opp i starten, og brukeren må skrive inn portnummeret der tredemøllen er koblet inn. Om tredemøllen hadde vært koblet til COM1 må brukeren skrive inn COM1.

```
Tilgjengelige porter:  
COM1  
Angi Com port verddi (Default: COM1): COM1
```

- For de resterende valgene er det bare å trykke enter for å velge default settings.
- Etter innstillingene er satt, kan brukeren skrive inn forhåndsdefinerte kommandoer. De som er satt nå er: on, off, stop, walk, jog og run. Hva kommandoene gjør er spesifisert i scriptet.

9.3.2 Bruk av signaler

Om man ønsker å sende kommandoer til Woodway tredemøllen over seriell porten, må man sende det i byte format. Løsningen vår var å lage lister av bytes (byte arrays) som vi sendte over en om gangen ved bruk av en forEach-loop. Under er en oversikt over de nødvendige kommandoene. Hver byte som sendes over slutter med 0D.

Skurer på tredemølle displayet og setter den i standard protokoll slik at den er mottakelig for fjernstyring:

0xAE, 0x30, 0x30, 0x0D, 0xA2, 0x30, 0x30, 0x0D, 0xA6, 0x31, 0x0d, 0xA5, 0x0d

Hele denne kommandoen kan sendes som én melding.

Når man skal sende over en kommando må man først begynne med å sende over A9, altså:

0xA9, 0x0D

For å sette hastighet benytter man A3 etterfulgt av hastigheten man ønsker å sette og avslutter med et stopbit. Husk å først sende over **0xA9, 0x0D** etterfulgt av ønsket kommando. Eksempelvis om man skal sette hastigheten til 2.77m/s:

0xA3, 0x32, 0x37, 0x37, 0x30, 0x0D

Her ser vi **A3** kommandoen etterfulgt av **32, 37, 37 og 30** som blir tolket som 2,770m/s og avsluttes med en stopbit **0D**.

For å stoppe tredemøllen sendes A8. Eksempelvis:

0xA9, 0x0D, 0xA8, 0x0D

For å skru av tredemølle display:

0xA2, 0x30, 0x30, 0x0D, 0xA6, 0x30, 0x0D

Kommandoliste:

| Kommando | Hex | Parameter (hex) | Oversatt hex |
|--|-----|-----------------|--------------|
| Sett timeout | AE | 30 30 | 0.0 |
| Protokoll bytte til kontroller | A2 | 30 30 | |
| Slå på tredemøllen | A6 | 31 | 1 / true |
| Protokoll bytte til standard protokoll | A5 | - | Null |
| Gjør møllen mottakelig for ordre | A9 | - | Null |
| Sett helling | A4 | | |
| Sett akselerasjon | A5 | | |
| Sett fart | A3 | | |
| Stop møllen/Ikke mottakelig for ordre | A8 | - | Null |
| Protokoll bytte til kontroller | A2 | 30 30 | |
| Skru av tredemøllen | A6 | 30 | 0 / false |