

Kravdokumentasjon

Unity integrasjon med SensMax TAC-B

IDATT2900-022

Erik Borgeteien Hansen,

Oda Alida Fønstelien Hjelljord

Revisjonslogg

Dato	Versjon	Beskrivelse	Forfatter
29.03.2023	0.1	Begynne på kapittel 1 og 2	Erik
24.04.2023	0.2	Begynt på kapittel 4	Erik
08.05.2023	0.3	La til manglende historier, la til kapittel 3	Alida, Erik
08.05.2023	1.0	Gjennomgang av dokument	Alida, Erik

Innhold

1	Introduksjon	1
2	Brukerhistorier	2
3	Domenemodell	7
4	Prototyper	8

1 Introduksjon

Dette dokumentet er skrevet i sammenheng med bacheloroppgaven til Oda Alida Fønstelien Hjelljord og Erik Borgeteien Hansen, hvor det utvikles mellomvare som kobler sammen spillmotoren Unity og den mennesketellende radarsensoren TAC-B 3D-WP fra SensMax. Oppgaven er forespurt fra det Trondheimslokale selskapet Able Magic AS, som ønsker å dra nytte av sensoren i sine Unity-prosjekter, og som dermed spesifiserer de grunnleggende kravene.

Hensikten med dette dokumentet er å utdype rundt kravene til oppgaven, dermed blir innholdet i dokumentet punkter som fremmer dette målet; brukerhistorier som går i detalj på ulike ønskede funksjoner av kodebasen vår, en enkel domenemodell som gir et oversiktsbilde over den planlagte koden og litt om hvordan vi tenker å prototype oppgaven.

2 Brukerhistorier

Som en Unity-utvikler

Ønsker jeg å kunne enkelt sette opp en MQTT-server

Slik at jeg kan motta og videreformidle meldinger fra sensoren

- Jeg ønsker å gjøre minimalt arbeid for å sette opp en MQTT-server slik at jeg kan fokusere på å håndtere sensordata.
 - Serveren bør tillate klienter å koble seg til som sensorer og klienter jeg setter opp for å lytte til sensorene.
 - Denne serveren bør kunne kjøre mens mitt prosjekt kjører.
-

Som en Unity-utvikler

Ønsker jeg å etablere kontakt med sensoren

Slik at jeg kan få data fra sensoren inn i Unity

- Hvis jeg har satt opp et Unity prosjekt og installert pakken, kan jeg koble opp mot sensoren.
 - Jeg kan koble til sensoren ved å dra et ferdiglaget spillobjekt (en prefab) inn i Unity og konfigurere den via Unity sitt brukergrensesnitt.
 - Jeg vil ha en måte å oppdage om sensoren er koblet til eller ikke.
-

Som en Unity-utvikler

Ønsker jeg å få tak i data som radaren sender i et velegnet format

Slik at jeg kan bruke dataen i Unity

- Hvis jeg har etablert kontakt med sensoren, kan jeg motta data fra den.

- Jeg mottar data ved å lytte på hendelser som sensor-spillobjektet sender ut.
 - Hver gang en oppdatering (hendelse) blir sendt ut, vet jeg at det er kommet ny data fra sensoren.
 - Jeg kan enkelt bruke dataen fra sensoren vet at dataen blir deserialisert til et C#-objekt.
-

Som Unity-utvikler

Ønsker jeg å kunne feilsøke kontakt med sensoren

Slik at jeg kan enklere finne feilkilden i tilkoblingen

- Jeg vil ha advarsler og/eller error printet i konsollen dersom jeg ikke får kobla opp mot sensoren, eller andre problemer oppstår.
 - Jeg vil kunne slå av denne feilsøkingen dersom jeg ikke har behov for den sånn at jeg enklere kan feilsøke andre deler av mitt prosjekt.
-

Som Unity-utvikler

Ønsker jeg at pakken skal ta lite unødvendige resurser

Slik at jeg kan bruke disse resursene i mitt prosjekt

- Jeg ønsker at denne pakken bruker minimalt med resurser for at mitt prosjekt skal ha mest mulig resurser tilgjengelig.
 - Denne pakken er modulær slik at jeg som bruker kan velge hvilke funksjoner jeg vil implementere.
 - Denne pakkens nødvendige funksjoner tar lite resurser.
-

Som en Unity-utvikler

Ønsker jeg å se mennesker i radarens deteksjonsfelt, representert i Unity
Slik at jeg kan se at posisjonsdata blir behandlet riktig

- Hvis jeg har satt opp prosjektet med pakken og etablert kontakt med sensoren, kan jeg se 3D-objekter i spillverden der hvor sensoren oppdager mennesker i den virkelige verden.
 - Jeg kan bruke eksempelkode til å raskt verifisere at jeg får inn data fra sensoren.
 - Jeg kan dra eksempelkoden fra pakken oppå sensoren for å oppnå ønsket funksjonalitet.
-

Som en Unity-utvikler

Ønsker jeg å få estimert hastighet på oppdagede mennesker
Slik at jeg kan bruke hastigheten i spillverden

- Hvis jeg har satt opp prosjektet riktig og etablert kontakt med sensoren, kan jeg få tak i hastigheten til oppdagede objekter.
 - Jeg kan få tak i hastigheten ved å lese dette fra objektet som sensoren sender ut med hendelser.
-

Som en Unity-utvikler

Ønsker jeg å få tak i entitetene sensoren ser
Slik at jeg kan bruke disse entitetene i spillverden

- Dersom jeg har et prosjekt satt opp med pakken, skal jeg klare å få tak i alt sensoren ser som en liste.
 - Jeg kan få tak i listen med å lytte på hendelser fra sensoren, og lese fra gitte objekter.
-

Som en Unity-utvikler

Ønsker jeg å kunne koble til flere sensorer i samme prosjekt

Slik at jeg kan bruke mer enn en sensor i prosjektet mitt

- Jeg vil ha muligheten til å koble opp mot flere sensorer slik at de kan brukes i ulike deler av mitt prosjekt.
 - Jeg kan konfigurere sensor-spillobjektet med en unik ID (for eksempel serienummeret), slik at jeg lytter til en spesifikk sensorer.
 - Jeg ønsker at de forskjellige sensorene kan plasseres på forskjellige steder i spillverden.
 - Objekter oppdaget av en sensor får relativ posisjon til den sensoren i spillverden.
-

Som en Unity-utvikler

Ønsker jeg å kunne loggføre statistikk om bruken av mitt prosjekt

Slik at jeg bedre kan forstå engasjementet rundt mitt prosjekt

- Hvis jeg vet hvor lenge brukere er engasjert med mitt prosjekt og hvor lenge det går mellom brukere, kan jeg implementere funksjonalitet for å øke interessen.
 - Sensor-objektet har data om aktiv/inaktiv tid som jeg kan benytte.
 - Det finnes eksempelkode som jeg kan benytte eller som jeg kan bruke som utgangspunkt for min egen kode.
-

Som en Unity-utvikler

Ønsker jeg at koden i denne pakken skal være godt dokumentert

Slik at jeg enkelt kan finne frem til de funksjonene jeg trenger

- Metoder og variabler har docstringer som forklarer funksjonen og retur verdiene sine.

- Det er en README som gir en gjennomgang av installasjon og oppsett av pakken.
- README gjør det tydelig hvilke funksjoner pakken tilbyr.

Som en Unity-utvikler

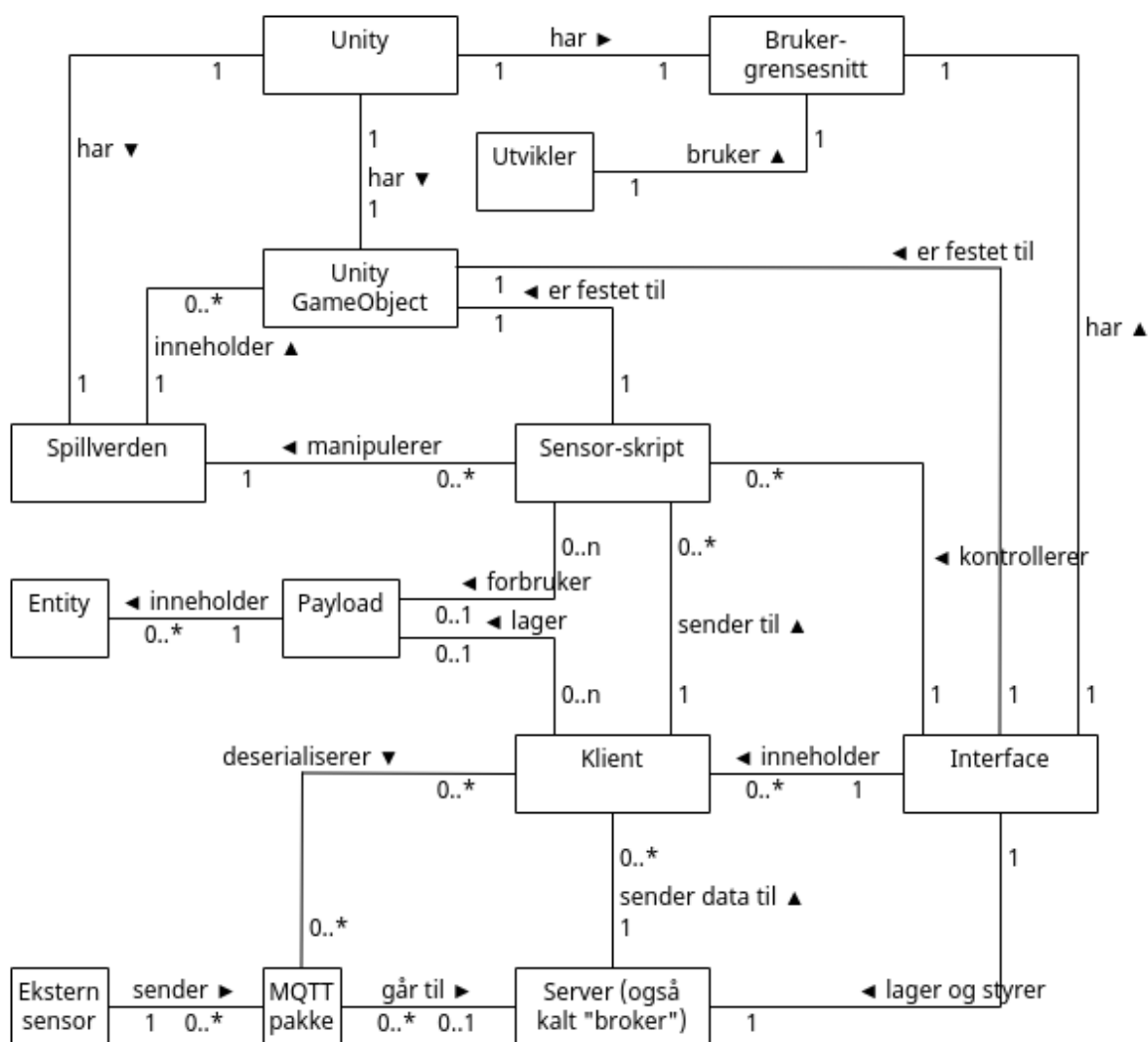
Ønsker jeg å kunne laste ned pakken gjennom Unity Asset Store

Slik at jeg kan finne og installere pakken på en måte jeg er vant til

- Pakken blir lagt ut på Unity Asset Store i tillegg til GitHub
-

3 Domenemodell

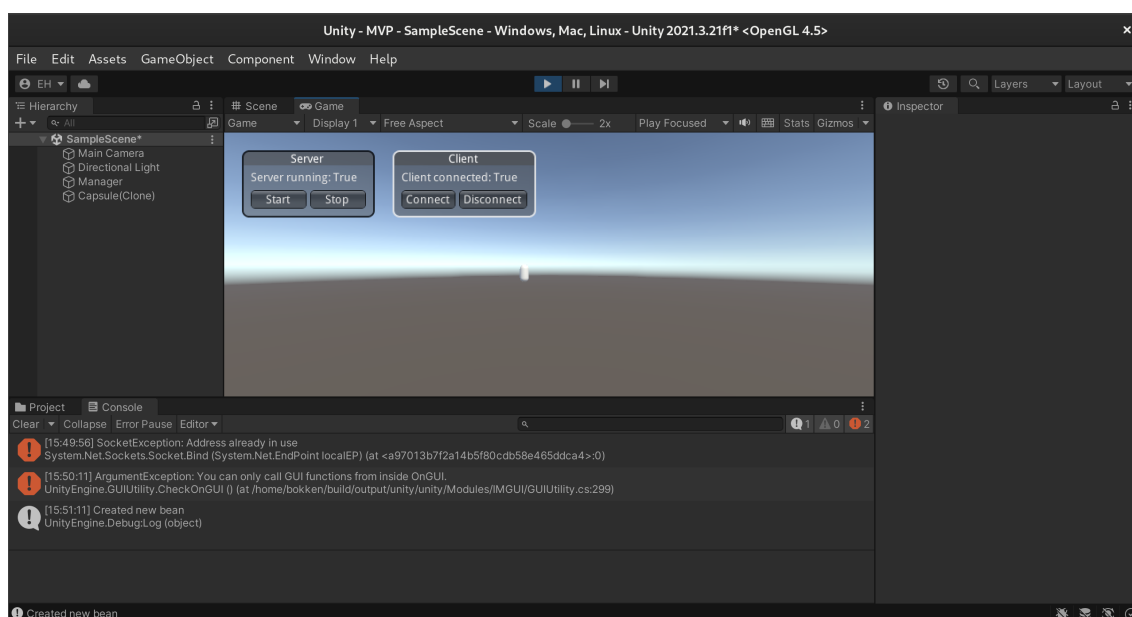
For å dekke brukenes behov må pakken kunne sette opp en MQTT-server og klient som kan motta og håndtere sensordata. Det trengs også datatyper for å deserialisere sensordata slik at brukeren enkelt kan benytte seg av data. Denne domenemodellen ble foreslått tidlig i prosjektperioden:



Figur 1: Domenemodell for en Unity SensMax TAC-B integrasjon

4 Prototyper

Med tanke på oppgavens fokus på brukervennlighet, samt Unity sin visuelle arbeidsflyt, blir det tydelig at et godt grensesnitt er en vesentlig del av prosjektet. Derfor har det vært svært nyttig å lage en prototyp i form av en MVP (Minimum Viable Produkt) tidlig i prosjektets løp. MVP-en har også vært hjelpsom i kommunikasjon med oppdragsgiver, og har også vært grunnsteinen i videreutviklingen av prosjektet.



Figur 2: Blide av MVP med instansiert objekt som representerer en person oppdaget av sensoren

MVP-en var ganske enkel, og holdt nødvendigvis til i Unity. Den besto i en MQTT server og en MQTT-klient på et spillobjekt i Unity, som representerer sensoren i spillverden. Den hadde som vist over et grafisk grensesnitt som lot brukeren starte og stoppe server og klient, og se om de kjørte som forventet. MVP-en instansierte objekter i spillverden som samsvarte med personer oppdaget av sensoren. Dette var for å vise at pakker fra sensoren ble mottatt og kunne benyttes i spillverdene.

Under utviklingen av MVP-en lærte gruppen mye om hvordan Unity fungerer, hvordan oppdragsgiver vil bruke pakken vi skal utvikle og hvordan vi skal strukturere resten av utviklingen. Etter fremvisning av MVP-en ble det etterspurt mange funksjoner som ikke hadde kommet frem tidligere i prosjektet.