

## Dokumentasjonsfil

- Valg av løsninger/elementer der flere har vært til vurdering.
  - Vi startet å lage WPF programmet med XAML sin innebygde button\_click funksjon. Men dette er ikke MVVM, så det gikk vi bort ifra, og satt opp løsningen i forhold til hvordan MVVM skal være
  - Knappene skulle egentlig blitt grået ut om programmet ikke har en godkjent fil, men commands bestemte seg for å ikke samarbeide. Det vi gjorde isteden var å la en boolverdi bestemme om man kan de/serialisere. dette var en grei løsning siden infosystemet til enhver tid forteller hvordan statusen er.
  - Vi var en stund litt usikre på hva vi skulle de/serialisere og hvordan formatet skulle bli. Til slutt endte vi opp med å lage en klasse som inneholdt alt vi ville de/serialisere, og så lot vi den de/serialisere seg selv.
- Spesielle utfordringer.
  - Å få de jærskla knappene til å bli grået ut ved hjelp av commands... de gadd jo søren meg ikke for fem flate øre! Satt i mange timer med det D:<
  - å finne en god måte å samarbeide med Unity for å få ut alle variablene vi trengte og tilbake igjen
  - å lage en dynamisk løsning.. mye av størrelser og designvariabler er hardkodet.. Vi liker egentlig ikke å hardkode, men følte det var nødvendig for store deler av oppgava, og man endte opp med et ryddigere resultat (i hvert fall front-end).
  - Å hente ut variabler fra Unity, da disse lå i Javascript-filer.
  - Å ha jobb parallellt med innlevering.
- Elementer du/dere er spesielt fornøyd med.
  - Designet ble oversiktlig og lett å bruke.
  - Vi føler applikasjonen ble mer levende med et live infosystem som viser info og ikoner om statusen til programmet.
  - Det ble et ganske kort program som gjorde alt det skulle.
- Elementer dere ville hatt med som dere ikke rakk (og hvorfor).
  - En mer dynamisk løsning både for WPF og Unity delen av løsningen,. Så man kunne fått ut annen/mer info uten å måtte endre så mye av koden, og slik at Stats Editoren enklere kunne brukes med andre Unity-prosjekter. Dette rakk vi ikke å få til fordi vi ikke har drevet så mye på å finne ut hvordan Unity var bygd opp, og hvordan den strukturerte objekter og hvor variabler ble satt..
  - Bedre struktur/arkitektur i Unity-filene, dele opp ordentlig mellom GUI og logikk.
- Formatbeskrivelse av JSON fil.
  - Json fila vår består av 2 lister med objekter. den ene lista er elementer med navn(special enemies og spiller), og en liste med elementer uten navn. Under hvert element ligger stats!