

# A0 møte 1, 08.02.24

## Sak 0 - Standardprotokoll

- Valg av ordstyrer [Egor](#)
- Valg av referent [William](#)
- Oppmøte/fravær [Egor](#), [Trym](#), [William](#), [Tarjei](#) / [Kevin](#) (Jobb)
- Godkjenning av innkalling [Ok](#)
- Godkjenning av saksliste [Ok](#)
- Regler for innlegg/replikk [Ok](#)

## Sak 1 - Forslag til ansvarsfordeling (Kan byttes om nødvendig)

- **Prosjektleder** (delegere arbeid, kommunisere med gruppeleder)
- **Systemarkitekt** (ansvar for struktur, modulær koding, grensesnitt, sjekk av kodekvalitet, dokumentasjon)
- **Utvikler** (Model - View, Controller (MVC), Main,)
- **Design og UX-ansvarlig** (Lyd, Bilde og grafikk)
- **Testansvarlig** (Fikse nødvendige tester og hjelper andre å kode)
- **Santas Helper** (Poteten i gruppen, hjelper de som trenger)

### Etter en åpen diskusjon, har vi fordelt rollene som følger:

- Prosjektleder: [Egor](#)
- Systemarkitekt: [Trym](#)
- Utvikler: [Alle](#)
- Design og UX: [Kevin](#)
- Testansvarlig: [William](#)
- Santas Helper: [Tarjei](#)

## Sak 2: Hva skal vi lage og hvordan?

- **Idee 1: Tower defence (med map creator hvis mulig)**
  - Bloons inspirert tower defence (kan være hva som helst, trenger ikke være "Bloons")
  - Forskjellige maps og design på "apene" og "ballongene"
  - Ulike "aper" som gjør forskjellige ting (basic-angriper, flamme-angriper (damage over time), is-angriper (slowness))
  - Ha med liv som, livene blir påvirket om ballongene klarer å nå enden.
  - Flere baner med 1, 2 eller 3 veier som ballonger kan komme fra
  - Map creator som spilleren kan lage sine egne maps
- **Idee 2: En platformer**
  - Mario/ super meat boy/ etc. insperert
  - Platformer med hinderløype, "power ups", fiender som kan skade deg (ta liv)
  - forskjellige baner som en spiller må komme gjennom for å gå til neste

### Sak 3: Kompetanse

#### Generelt:

- **Egor, Tarjei, William, Trym:**

Studerer: Datasikkerhet.

Alle har samme kompetanse med Java, ettersom at vi går i samme klasse.

Gjennomført emner som er relevant:

INF100 (Python)

INF101 (Java)

INF102 (Java med fokus på optimalisering, algoritmer og

datastrukturer)

INF142 (Datanett-Python)

INF113 (OS, C-kodespråk, assembly, minnehåndtering)

- **Kevin:**

Studerer: Datateknologi

Har samme kompetanse som de andre.

Gjennomført emner som er relevant:

INF100 (Python)

INF101 (Java)

INF102 (Java)

INF113 (OS/ C-kodespråk)

INF115 (Databaser)

#### Spesifikt (Sterkheter og svakheter):

- **Egor:**

Har kontroll på store deler av "MVC", men syns jeg sliter litt med tester.

Hvordan man implementerer de og hvordan selve testen skal teste et spesifikt ting i koden.

Syntes også at det er vanskelig å komme meg i gang med å kode, men å lese og forstår hvordan koden er satt opp og fungerer er jeg sterk på.

- **William:**

Føler meg mest komfortabel med både model og controller i "MVC", føler at jeg liker back-end-coding og syns dette er interessant og spennende.

Jeg liker også teste selve koden. Front-end derimot, liker jeg ikke så godt.

- **Tarjei:**

I model-view-controller så har jeg mest kontroll på "controls" og hvordan dette skal kommunisere med resten av koden.

Jeg sliter med kreativitet når det kommer til å komme på ulike ideer til hvordan komplekse problemer kan løses.

- **Trym**

Synes det å tenke utenfor boksen når det kommer til bruk av forskjellige datastrukturer er spennende, noe som åpner muligheten for en del optimalisering i prosjektet. Liker å teste sikkerhet i applikasjoner, dette kan komme godt med når vi skal teste spillet for bugs.

Synes det er vanskelig å komme i gang med store kode-prosjekter. Har lett for å hente inspirasjon fra kilder når jeg står fast, istedenfor å la hodet jobbe lenge.

- **Kevin**

Jeg er en stor fan av design, UI og generelt frontend-aspektet av programmer og systemer. Føler at jeg har et øye for slike ting, og klarer godt å formidle

programmets tematikk til brukeren på en fin måte som har “flow”. Kan slite litt i starten av et prosjekt, men når ideen for en viss løsning har truffet, er alt fine and dandy. Sliter litt med å skape tester.

#### **Sak 4: Hvordan skrive gode kommentarer og følger med på “Git”?**

Som systemarkitekt, fremmer Trym det å skrive god Java-dokumentasjon på metoder og klasser. Han mener det er nøkkelen til å la teamet skape et modulært prosjekt som fungerer bra som en helhet. Vi har derfor funnet noen ressurser vi kan se over, for å passe på kvaliteten på dokumentasjonen. Når dokumentasjonen er god, lar det oss bygge på hverandres arbeid.

- **Gode “JavaDoc”:**
  - ▶ **Java 1 Online, [4.14] Writing Classes: Javadoc Comments**
  - ▶ **How to use Javadocs in IntelliJ**
- **Git:**
  - ▶ **How to use GIT when working with a team?**
  - ▶ **Git for Professionals Tutorial - Tools & Concepts for Mastering Version Control ...**
  - ▶ **Git & GitHub Tutorial for Beginners #1 - Why Use Git?**