

## **Møte 8.april 2015**

**tilstede: Alle**

---

vi generaliserer - forenkler, ikke så naturtro prosess

lager spillet utvidbart - lett å forandre, legge til nye elementer i spillet

Ide til en abstrakt klasse rom. som arves av stue. Stue har et utseende og objekter man kan bruke som avis, skrivepult.

Finner substantiver - behov kan være en

spill er gøy fordi man har mange kortisiktige mål

vi deler inn i tid. En dag, en uke, en bok (10 uker - alt over vil være uten lønn)

man skal skrive 350 sider - dersom du får til dette, og har behovene dekket vil du ha skrevet en god bok.

Testing! Enhetstesting! Bruke Google Analytics

Siden har kvalitet - en side gir 1 - 2 eller 3 poeng.

Skal kan spille uendelig. Man kan tape men aldri vinne.

Man loggfører hvor mange sider man har skrevet, hvor mye kaffe man har drukket. Loggen kan sjekkes underveis. Loggen skal kunne bidra til statistikk.

Vi setter git ignore på Makefile eventuelt brancher. Setter

**Til neste gang** - tenkte litt på klassediagram, se på Qt og se på designpatterns.

## **Møte 10 april 2015**

**tilstede: alle (delvis - du har kanskje ikke fått med deg hele diskusjonen)**

---

Vi har diskutert design patterns, klassediagram, brukergrensesnitt flyt og generelle mål for prosjektet.

Vi ferdigstilte oppgaveteksten og konkretisert hva vi så på om minste mål, middelse mål og høye mål.

se [prosjektbeskrivelse.md](https://prosjektbeskrivelse.md)

Vi prøvde å definere ulike klasser og sammenhengen mellom disse. Vi diskuterte hva vi mener med tid og hvordan denne skulle implementeres og påvirke dataen i spillet.

Vi diskuterte klassene

- **Need** - mulig abstrakt super klasse, men hvert behov har muligens samme verdier. Derfor trenger vi kanskje bare en klasse need som blir deffinert (sleep, hunger, inspiration etc) der den blir instansiert. Need tar da i mot verdier i parameter som sier hva slags behov det er og hvor fort/sakte det synker per tidsenhet. Klassen Writer kan ha en beholder for Need objekter
- **Object** - Objekter tilhører Room og kan gi positiv og eller negativ innvirkning på forfatteren. Objektet vet hvor mye positiv og negativ innvirkning det har. Bruk av et objekt vil pause skrivingen. Object er mulig abstrakt superklasse for
  - Wine (+ intspirasjon - sleep (hangover))
  - Window (+ inspirasjon - inspirasjon eller ingen effekt)
  - Coffee (+ energi - blære - energi etterhvert)
  - Typewriter (+book progress)
  - Fridge (+ hunger)
  - Bed (+energi/sleep)
  - ....
- **Room** - inneholder objekter og har et utseende. Room kan være baseklassen til LivingRoom som har objektene typewriter, window mm. Room er en subklasse av House (*Ikke subklasse av house! - Hallbjørn*)
- **House** er en beholderklasse for Room og subklasse av World eventuelt definert i World - Eller noe slikt. (*Definert i World - Hallbjørn*)
- **World** kan fungere som et knutepunkt og inneholder et objekt av Writer, Forfatter, Time, Weather, House.
- **Brake** - abstrakt klasse som arves av objekter som kan pause skrivingen. Tanke- hele spillet skal jo kunne pauses av spilleren. Brake kan være en super klasse for Brake Need og Brake Text. - man pauser spillet ved å kalle breakeProgression() på alle objekter som har en progresjon.
- **Effekt** - abstrakt klasse som arves av objekter som kan gi en effekt på needs.
- **Boost** - en abstrakt klasse, som skal arves av objekter som gir boost til de ulike behovene er writer har
- **Statistikk** - 'husker' data
- **Filbehandler** - ansvarlig for å skrive og lese fra fil
- **Text** - abstrakt klasse som er base klasse for Book
- **Book** - har en tittel, sideantall og progresjon.
- **The Game** - har objekter av Filbehandler, GUI, World
- **Main** - starter spillet

andre mulige klasser

- **Events** - en event kan ha en positiv eller negativ innvirkning på behovet til forfatteren.  
Abstrakt base klasse/interface for
  - **Minigame**
  - **RandomEvent**

**Her vil det nok komme en del forandringer!**

Spillet har en event que - en beholder legger til ulike gjøremål i en kø. Hvert gjøremål tar en viss mengde tid, men kan avbrytes. Flere gjøremål kan legges til i køen. Dersom ingen gjøremål finnes, vil forfatteren skrive.

Hva skjer dersom et behov går til bunnen? Vil forfatteren automatisk gå på do/spise/sove eller dø?

Hvordan skal behovene oppdateres? Vi fant eksempelkode [her](#):  
der subjektet( forfatter i vårt tilfelle) og et behov, blir sendt med som objekt i parameteret til metoder. En metode har ansvar for et behov og har en if else setning som bestemmer hva subjektet gjør. Deretter bestemmer metoden hva utregningen blir.

en annen interessant idé fra artikkelen :

Energy could be based on hunger: The less hungry you are, the faster your energy regenerates.

Qt hadde en tid funskjon som vi skal se nærmere på.

Vi diskuterte at brukergrensesnittet kan bestå av en tegning som kan ha mange tomrom. Disse tomrommene skal bli fylt med illustrasjoner som skal endres ofte. F.eks en vinflaske som kan byttes ut med en mer fancy vin og eller bli tom etterhvert som forfatteren drikker. Dersom spilleren holder musepekeren over et klikkbart objekt skal noe skje for å vise at objektet er klikkbart. Musepeker kan endre utseende og eller en kontur rundt objektet kan vises. Her vil vi kanskje ikke gjøre det for lett for spilleren.

Progresjonsmeterene er avhengige av verdiene til need objektene, men også relative til viduet de befinner seg i - og kanskje også relative til hverandre?

Justeringer må gjøres slik at spillet skal kunne sees på ulike skjermer. Dirty flag design pattern er laget slik at objekter kan være plassert i forhold til hverandre. Dette er muligens noe vi kan se på når vi kommer hit og se om det er mulig å få objektene til å skalere i forhold til hverandre. **MEN KEEP IT SIMPLE STUPID**

Skisser av brukergrensesnittet, klassediagram og flytdiagram ligger under skisser i CPP mappen

Til neste gang: på tirsdag er det prøve i CPP og det er ellers mye å gjøre. Vi prøver å sette oss mer inn i ting dersom vi har kapasitet, men møtes på tirsdag etter prøven og diskuterer veien videre og deler opp noen arbeidsoppgaver dersom mulig.

**GOD HELG!**

## **14. APRIL**

---

Aller er slitne etter cpp prøve.

Mål for dagen prøve

å deligere oppgaver

når er neste møte og hva skal vi ha gjort til da

Stine ser på filestream og prøver å lage dette som some en singleton

Alle -setter seg inn i Qt- hvordan fungerer det?

Klassene føles fortsatt litt abstrakt - kanskje på tide å prøve å sette sammen.

Lær mer om design patterns - hva er kanskje hensiktsmessig å bruke

Neste møte torsdag (ikke forelesning jippi!) jobbe litt sammen denne dagen  
oppsummering på torsdag på slutten av dagen.

16. april.2015

---

Siden sist Hallbjørn har jobbet med klassediagram og laget en revidert utgave. Vi snakket om hvordan man kan jobbe med objekter som gir boost til forfatteren. Et objekt kan legges inn i en boost que, men vi er ikke sikre på hvordan behovene blir oppdatert. Dette skjer jo uavhengig av hva han gjør for øyeblikket. Eksempelvis så vil effekten av å drikke kaffe vare lenger enn handlingen å drikke kaffe.

Kristoffer har laget en prototype av spillet med den klokke og behov som stiger/synker og som kan påvirkes av å trykke på knapper som spis og sov. Fabelaktig!

Stine har satt seg in i qt

Vi diskuterte også:

Hvordan regner man ut kvalitet på boken: antall poeng og antall ord i en kvalitetsvariabel.

En avgjør hvor bra ordet er og hvor mange ord du skriver. Dette gjøres per minutt.

Kvaliteten representeres visuelt - ikke numerisk. - Man får statistikk hver dag.

Vi trenger en klasse dag eller finne en måte å representere tiden i deler.

Et objekt kan være klikkbart og kan være en boost. Skal den arve begge deler eller skal det være arv? Dersom forfatteren skal kunne bli påvirket av været så er det nok best at et objekt ikke trenger å være klikkbart for å kunne gi en boost. Vi har ingen objekter i tankene som skal kunne være klikkbare å ikke ha en effekt på forfatteren, men for utvidbarhet legger vi kanskje til dette? Men skal i så fall være i en klikkbar klasse dersom det ikke gis noe boost?

Real time vær? Fra når skal været være? Været - være fra de siste 30 dagene eller real time vær? Effekten vil kanskje være større dersom man blir påvirket av det faktiske været?

Inndeling av oppgaver - prøver å løse problemer individuelt og setter oss videre inn i qt.

20 APRIL

---

Siden sist:

Hallbjørn har gjort masse bra jobb med klassediagram og lastet opp versjon 2 til github og har jobbet med boka

Kristoffer har utvidet simulasjonen og har fått parametre multiplayer og booster

Stine har sett på filbehandling

Fordeler med å lagre i XML - man kan hente ut de data man trenger enkelt. Enkelt å bruke, men vanskelig å implementere?

Struktur - vi lagrer ord ikke sider- Det gir mer håndterlige tall.

En vektor holder poengene per side antall der indeks er antall poeng. Hvert poeng/indeksposisjon har et antall sider. Dersom indeks 2 har 40 sider er det 40 sider som har kvalitet på 2.

En vektor har dagens ord og når det har gått en time/dag opptateres respektive vektorer (tre vektorer tilsammen)

Hvor krevende er det å skrive til statistikk - kanskje hver dag.

Stine: Filbehandling, xml og statistikk

Kristoffer: algoritmer og boostere og enkel GUI

Hallbjørn: book , (er det hensiktsmessig å også lage et hierarki av tekster?), se mer på qt.

21 APRIL

---

Mange interessante diskusjoner. Vi har diskutert hvordan verdier skal endres og hvordan de skal fungere. F.eks. skal statistikk lagre en XML med dager som inneholder array med antall sider og kvalitet på sidene.

Vi ser mer for oss hvordan ting skal fungere og begynner å se at det er en del spillmekanikker som blir inkludert bare for å prøve, f.eks. økonomi.

Vi opplever at spillet kan være kjedelig. Så derfor skal vi ha med minigame! Les “typing of the dead”, “fang fluer”,

Hallbjørn: Grafikk (legge inn bilder), GUI-elementer og se på minigames. Poupes.

Stine: Fremstilling av statistikk og XML-import.

Kristoffer: Fortsette med gameplaymekanikker. Make it play!

Designvalg:

250 ord per side. 1-2mnd på en bok.

Quicktime events/poups, jehovas vitner, mor ringer, filmselskap vil kjøpe manus+++

22 APRIL

---

Til idag:

Kristoffer har jobbet med flyt og booster multiplier/values og generell gameplayflyt.

Til imorgen:

Lage ny bok og alle variablene skal være med. Skille ut en del kode fra mainview inn i egne klasser så ikke mainview er så bloated. Det bør helst bare være klokke og basic gui der.

22 APRIL

---

Hallbjørn har gjort følgende til i dag:

- Funnet ut hvordan lage et popUp vindu.
- Legge til et skalerbart bakgrunnsbilde til vinduet (QWidget) ++

Hallbjørn fortsetter å jobbe med:

- PopUp-vindu:
  - Sette ulike innstillinger ... (fixed og/eller responsiv)
- Forsøke å lage en “quick time event” ved et popUp-vindu som ber deg gjøre et valg. Avhengig av valg (Ja/Nei) skal den trigge boostere i spillet. (legges inn i køen til writeren. )
- Generalisere pop-up: Gjenbrukbar kode metode som tar imot verdier og setter dem i pop-up vinduet.
- En pool for quick time events, som man kan randomisere.
- Ideer:
  - Man bør flagge hvilke eventer som er kjørt så man ikke får dem på nytt ofte.... Men kanskje ikke nødvendig om det er ting som kan skje på nytt og på nytt Eks: “man har sovet dårlig- starter dagen med halv energi”

- Alle “boostere/events” som sendes med til popUp-vinduet må ha en `execute()`, som kalles ved trykk på de ulike knappene. På den måten behøver ikke vinduet vite av noe som helst. Hver booster eller event (Det vil si det objektet hvis man skal kjøre `execute()` på) kan gjøre forskjellige ting: legge en booster i køen til writer, sette en multiplier(environmental boost) eller starte en ny popup, eller minigame.
- Om man vil ha noe til å skje umiddelbart før man skal fortsette med andre boostere man har lagt til i køen kan man gjøre følgende triks: `køen.push_front( ... )`, i motsetning til alle andre som man legger inn på vanlig måte hvor man gjør `køen.push_back( ... )`.
- Om man vil ha en latenstid før man kan gå igang med en ny booster kan man legge til en ny booster som ikke gir noe boost (alt er 0) forrest i køen: `køen.push_front( ... )`. Eksempel på når dette kan være aktuelt: PopUp : “Noen barn har veltet de fulle søppelkassene dine og du må bruke en time på å rydde opp etter dem” OK ... En time går uten at du får skrevet, drikket kaffe eller noen andre ting, bare behovene som “tikker” ned i vanlig hastighet.

## 23.APRIL

---

Pop ups - hvordan få til at flere views / popups snakker med hverdander - kristoffer og hallbjørn jobber med dette.

hvordan få visuelle aspekter til å skallere - stine jobber med dette

Stine har

laget png bilder som skal vise de ulike objektene i spillet, forfatter etc.

Laget en egen QLabel klasse som kan ta imot klikk.

Laget en egen Qgroupbox hvor paint event metoden blir overskrevet for å kunne skalere bakgrunnen (fra hallbjørn sin kode )

Skal fortsette å jobbe med dette samt lage visuelle fremstillinger av statistikk - fint om dette har en visuell forbindele med spillet pixelart i ‘study’ vinduet. Disse skal hentes fra xml-en

Xml-en og filbehandler klassen skal gjøres om til singleton

## 28.APRIL

---

Vi fikk problemer med at boost biblioteket ikke var inkludert i mac sitt bibliotek. fordi---

- selv om det var det på linux bestemte vi oss for å prøve å finne en løsning. Man kunne laste ned boost på mac og legge til bibliotekene og includpath i .pro filen slik :

```
macx{

    INCLUDEPATH += "/usr/local/Cellar/boost/1.57.0/include"
    LIBPATH += "/usr/local/Cellar/boost/1.57.0/lib"
}

windows{
    INCLUDEPATH += "c:\\local\\boost_1_58_0"
    LIBPATH+= "C:\\local\\boost_1_58_0\\libs"
}

LIBS += -lboost_random \
        -lboost_chrono \
        -lboost_system
```

Men man var da avhengi av at de som skulle kompilere koden hadde biblioteket installert på et sted som samsvarte, la til sin path eller i værste fall - måtte laste ned biblioteket.

Vi bestemte oss isteden for å bruke bcp

[http://www.boost.org/doc/libs/1\\_43\\_0/tools/bcp/doc/html/index.html](http://www.boost.org/doc/libs/1_43_0/tools/bcp/doc/html/index.html)

et verktøy man bruker mot boost biblioteket for å hente ut den koden man trenger og la denne til i vår egen prosjektfil. Prosjektet kan nå kompileres mer uavhengig av platform.

Vi har jobbet med MVVM for prosjektet. Qt er svært godt egnet for denne fremgangsmåten. fordi-----

Bilde med forfatteren er nå skallebart! Viewet med forfatteren kalt studyview kan fylle en gitt øvre prosent av vinduet. Flere individuelle objekter kan klikkes på og alle skallerer i forhold til vinduet. Det oppnås ved å ha flere objekter som arver QLabel. QLabel er i utgangspunktet ikke 'klikkbar', men dette overskrives i klassen ClickLabel. ClickLabel er sub klasse av ScaleLabel. ScaleLabel tar i mot posisjon og størrelse av foreldervinduet og tegner seg selv deretter for hver gang det skjer en paint event. option label er subklasse av ClickLabel og inneholder tekst som lar brukeren gjøre valg.

Flbehandler klassene er nå endelig singletons



## todos()

Man skal kunne trykke på forfatteren og skrive ny bok dersom det ikke finnes

Boka må oppdateres kontinuerlig - hvor langt har man skrevet

Statistikk og visualisering venter vi med. (...det venter vi veldig lenge med)

Lag copy constructor og move constructor i forfatter og bok - og bostere

## Lage ny bok (og gi den tittel) (stine og kristoffer)

Se på ordgenereringsalgoritmen. Den virker litt humørsyk nå (Kristoffer lager ticks for hver time og dag, hallbjørn gjør at boka oppdateres kontinuerlig. Kristoffer ser mer på algoritmen.)

## Realtimeoppdatering av antall ord (se over)

## Legge inn design (stine og kristoffer)

End book, hva nå (hallbjørn)

## Progressbar for bokskriving (kristoffer)

End game (når skjer det), etter 100 dager. Man oppsummerer det som er skrevet og forhåpentligvis statistikk. (Hallbjørn)

## Se på dagteling/lagring (Hallbjørn)

## Endre status på spill (Nå står det idle, bør stå "skriver bok", "spiser sandwich", "jager katter" hører hjemme i statusview)

Gi straff for lave needs

## Definere hvilke needs vi skal ha og hvordan de skal påvirke boksskrivingen

- Hunger ()

- Sleep

- Inspiration

Primærbehov og skrivekvalitet. Primærbehov avgjør hastighet og inspirasjon hvor bra boken blir. (Kristoffer)

Lage flere "randomevents" som gjør at man mister f.eks. sider

## Fordele filene i mapper

## Endre knappene for speed

## Husleie (kanskje som event, etter 7 dager så trekkes \$100 fra konto med en popup?)

Saving/loading/vær/døgn (stine)

Ha en innboks for meldinger? (kommer nok litt seint)

Få events inn i en liste som snakker med en list view

Se på boosters. Hvilke typer skal vi ha og hvordan skal de påvirke (event, environment og singleboosters?)

Dag telling er veldig, veldig ducked up

Meny, appikon og "finish"

Needs kan gå under 0 (inntil -10, og maks 110)

Rydder i kode! slette ut-kommentert kode og kommentere der det trengs. (alle)

Hendene til forfatteren må kunne stoppes / endre tempo

skriv rapport

Sjekk etter minnelekasjer!

Programmet stopper ikke når det lukkes før du har startet spillet. Fordi closeevent er på mainview?

Skal det som vises av antall sider, kvalitet++ formateres og beregnes hos writer?

Skal vi rename boostercontroller til needscontroller? Da kan man rettferdiggjøre sjekk på needs++ og sende signaler fra boostercontrol til mainviewcontroller (eller rett til popupfactory)

popupklassen arves av mange klasser - implementere dette

Sette min str på vinduet! (stine)

29.april

Forslag til flere events:

- cinders burning - om en stund så vil / noen av arkene dine ta fyr!
  - treger å gi et signal - når lyset slutter å brenne. booster har en event inne i seg som trigges for å få til dette. Tanken her er at når man har tatt et valg så skal dette ha konsekvenser senere
- moren din ringer - vil du ta telefonen
  - Ja - hun snakker i tre timer - du mister tid! (sult, søvn og inspirasjon)
  - Nei, dårlig samvittighet
- vasken drypper. vil du prøve å reparere den eller la den være?
- pizzabudet leverte feil, vil du betale pizzabudet og få mat? \$5

Hva med hendelser som trigges av at du f.eks. spiser / sover - mareritt

Number of pages oppdateres ikke i book.

Flytte skrive boklabel til den under writer

Oppdatere gamestatus riktig

Sove på sofaen funksjon

Legg til get-metoder for om hunger, sleep eller inspiration er lav/none

Antall sider \* kvalitet = \$\$\$ (beregning for end book)

Endringslogg (fra Kristoffer)

- Har snudd verdiene for needs og booster. Så nå er +0.1 på hunger, ikke -0.1 på hunger når den øker. Så nå står det på booster +0.1 når du spiser, ikke -0.1.
- Writing blir satt på true når man begynner å skrive.
- is\_writing setter writing=true.
- Man kan ikke lage mer enn en bok om gangen.
- Man mister inspiration når man skriver
- En SVÆRT enkel måte for å beregne fart på bokskriving er laget.  
 $10*((hunger+sleep)/200)$

Dette fordi sleep + hunger i utgangspunktet er lik 200. Og siden sleep og hunger kan være inntil 110 så kan man faktisk få en bonus!

- Quality blir beregnet like lett,  $10*(inspiration/100)$
- Status oppdateres
- Booster sin tostr viser kun verdier med verdi. så om hunger er lik 0, så vises den ikke
- Statuslinja er blitt lengre!
- sliders er blitt progressbars
- Animasjon på writer kun når det skrives

### **Ting vi ikke implementerer, men burde ha gjort:**

- Måneder og år.
- Flere typer quick time events og flere typer eventfactories.
- Writer og weather burde vært i environment-listen. Men den er enn så lenge tom. Det skjer nok ikke, blir heller litt tungvinte løsninger på det.
- Environmentalboosters, rommet blir gradvis skitnere.
- Vi burde hatt en bedre plan på UX-opplevelsen med mer feedback til brukeren, men det ble nedprioritert. Dette kunne blitt løst med et ekstra datafelt i de ulike eventene++++.

### **Begrunnelse på valg**

- Datatyper. Siden det ikke er så mange objekter så har vi ikke vært så nøye på valg av int, short+++ mye fordi vi ikke vet hvordan de skal brukes senere og vi vil unngå en Y2K fadese hvor x antall ingeniører må bruke tid på å rette opp vår gjerrighet på bytes.

- Ticks, et sekund per endring. Gir lav framerate og er ikke så smooth. men er veldig enkelt å gjøre beregninger utfra. 1 sekund = 1 minutt.
-