

Introduktion

Grundlæggende Programmering - Projekt

Generelt om projektet

- Jeres program SKAL:
 - Have en GUI
 - Benytte en Database
 - Have en serie af unit tests som beviser programmets funktion
 - Leveres som en JAR-fil som vi (undervisere & censorer) kan køre
- I skal:
 - arbejde i grupper af 3, som i selv danner.

Ekstra Materialer

- 5 instruktionsvideoer er uploadet til LearnIt
 1. Installationsguide
 2. Introduktion til IntelliJ IDE
 3. JavaFX Tutorial - GUI design
 4. Testing med JUnit & Mockito
 5. Introduktion til versionsstyring med GIT

... Og 1 tekst, om testing!

- Lidt om Unit testing
- Lidt om test-drevent design & udvikling
- Introduktion til Mocking

videoerne mv. er vejledende

- I må fortsat vælge teknologier selv, f.eks.:
 - Swing til GUI
 - BlueJ/Eclipse/Netbeans som værktøj
 - SVN/Mercurial/CVS til versionsstyring
 - MS ACCESS / MSSQL til Database
 - TestNG til unit testing
 - EasyMock til Mocking (hvis overhovedet nødvendigt?)

Målet

1. At i bliver bedre programmører
 - a. Jeres første ikke-trivielle program
 - b. I kommer til at fange fisk selv (?)
2. At i undgår fejltagelserne fra min årgang
3. At jeg bliver glad når jeg læser jeres rapporter..!

Opfordring mht. det tekniske

- Observation omkring dårlige programmører:
 1. De har ingen tests eller måder at verificere deres kodes adfærd

War Story: Modern AI for Games

2. De har ingen versionsstyring

War story: Førsteårsprojekt

Hvad skal vi .. ?

Grundlæggende Programmering - Projekt

Intended Learning Outcomes

- mit.itu.dk - Kursusbasen
- ILO's - dette er grundlaget for din karakter!
- Assessment Form - hvordan i bliver bedømt!
 - ... With following oral exam supplemented by hand-in
 - Shared responsibility for the report

IL0's omfattende opgaven

1. Analysere Problemformuleringen
2. Foreslå programdesign
3. Programmere systemet
4. Præsentere .. mundtligt og skriftligt, systemets formål, opbygning & virkemåde
5. Kunne afprøve (teste) systemet

ILO Opsummering

- Programmering
 - Vigtigt, fagets "håndværk"
 - Refleksion, design mv. er dog samlet set vigtigere..!
- Brugergrænseflade
 - Bør være simpel, ikke det primære fokus
- Ansvar
 - Gruppen hæfter kollektivt, hjælp og inddrag hinanden !

ILO Opsummering, part 2

- Tests
 - Separat ILO
 - Beviser at i har reflekteret over designet
 - Beviser at programmet overhovedet fungerer
- Analyse & Programdesign
 - Læs problemformuleringen gentagne gange, lær den udenad!
 - Bevis at i har tænkt - inddrag figurer, arbejdsblade mv.

Rapporten

- Præsentation og afgrænsning af krav til systemet
 - Udledt af de arbejdsopgaver, som opgaveformuleringen giver
- Begrundede designbeslutninger
 - Figurer, diskussion af designvalg
- Refleksion over tests: opfylder systemet kravene ?
- Refleksion over process: har vi nået projektets lærsmål?
- Konklusion: Hvad nåede vi? Hvad mangler vi?

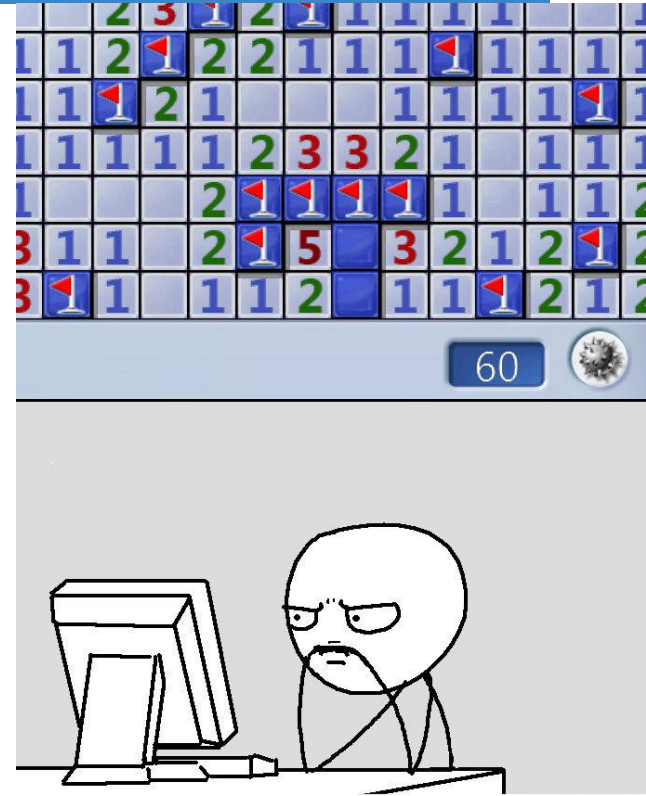
Iterativt Design

- Scenarie: Svenskerne sejler forbi KBH uden at betale handelsafgift.
- Opgave: Sænk de svenske både
- Forslag: Byg en kanon, sænk dem!
..Men hvordan ?

<https://www.youtube.com/watch?v=XP400ArkP4s> (Richard Buckland, Extreme Programming)

Problemløsning

- Du har ikke altid nok information eller viden til at løse hele problemet med det samme
- Løs et simplere problem - det giver måske indsigt til, hvordan det komplekse problem skal løses.

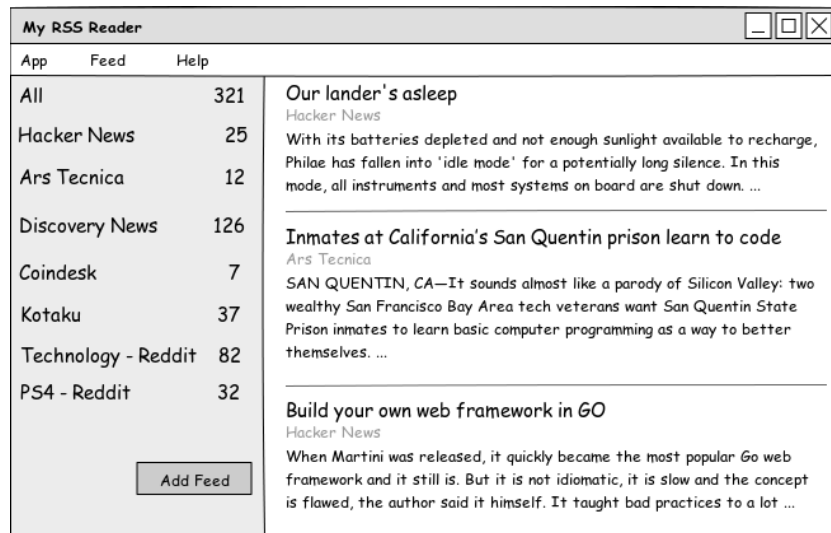


Databasedesign

- I er ikke eksperter herpå – kravene slækkes
 - PHEW!
- ... Men i er forventet at:
 - Dokumentere & Illustrere jeres design
 - Reflektere over designet:
 - Hvordan understøtter databasens design og data arbejdsopgaverne ?

GUI Design, 1

- Iterativt design... !
- Mocking/Sketching
 - Kan lave mange hurtige forslag
 - Kan diskutere og iterere på forslagene - videreføre de gode elementer.



Gui Design, 2

- Overskuelighed
 - Gruppér relateret data på skærmen,
 - Adskild ikke-relateret data
- Optimeret til opgaven
 - Hver tab/vindue optimeret til én type opgave.
 - Være optimeret til de hyppigste opgaver
- Lad data & arbejdsopgaver drive GUI-udvikling

Opskriften på en katastrofe

- Kod derudad
- Ingen tests før til sidst
 - Et par nemme test skrives, DONE!, NEXT!
- Febrilsk rapportskrivning, fra 0 til 15 sider på 3 dage
- Ingen arbejdsblade, ingen figurer
- Til eksamen: "Jamen, det var ikke mig der lavede det der"

Opskriften på et godt resultat

- Læs opgaven igennem, diskutér i gruppen
- Planlæg, hvilke delopgaver der løses først
 - Med nogle dages mellemrum: skal vi justere i planen ?
- Skriv tanker om design ned i arbejdsblade
- Lav figurer, prototyper mv. før i implementerer noget
- Skriv tests, mens i løser delopgaven
- Sæt rigeligt med tid af til rapporten. Skriv lidt undervejs!

Held og Lykke!