

# Mötesanteckningar



GÖTEBORGS  
UNIVERSITET

**CHALMERS**

Gustaf Ringius

May 12, 2014

Dagordning:<sup>1</sup>

## 1. Gustaf har:

- Fixat så ljudet endast kräver en tråd för att spela upp två ljud samtidigt.
- Kört findbugs och ändrat i konfigurationen i pom så den printar ut buggar med låg prio utan att faila byggandet. Har fixat 30 av buggarna den skrev ut varav de flesta var endast variabler som hade för stora scope, glömda exception och variabler som aldrig användes. `fileHandler` var enda klassen som hade högprio- buggar och dessa har lösts och klassen har gjorts om. Gjorde även ett försök att tillfredsställa vissa småsaker som findbugs klagade på. Detta var dock inte speciellt enkelt då den inte specificerar vilka konkreta rader det handlar om eller vad man möjligtvis skulle kunna byta ut med. T ex så klagar den väldigt mycket på att vi använder `Math.Random` och att man bör använda `nextInt` istället. När man då skapar en `Random` för att använda `nextInt` då den behöver en instans så klagar den på att instanser av klassen `Random` är osäkra objekt som den inte vill godkänna. Ytterligare ett steg togs med ett försök att använda `SecureRandom` men denna vill den inte heller acceptera och där drogs en linje med att “enough is enough”. Den klagar väldigt ofta på scope på variabler samt att den vill att alla `private final` variabler som får sitt värde innan konstruktorn och inte ändras ska vara `static` för att undvika att varje instans skapar egna kopior av dessa. Den ser dock inte att vi använder oss väldigt ofta av singletons vilket gör en sådan varning möjligtvis redundant. Det man sparar med att sätta dessa variabler `static` är möjligtvis minne men eftersom våra klasser redan är singletons så finns det inget direkt minne att spara då det enbart kommer finnas en instans av den variablen anyway, vilket även Felix påpekade. Gustaf gjorde några ändringar för att se hur många av de 120 klagomål som findbugs skriver ut för GUIt som är just av den tidigare nämnda karaktär och det är minst 40 sådana varningar men eftersom Felix och Gustaf ansåg att findbugs printout inte är en biblisk skrift så reverterades dessa ändringar av Gustaf ifrån repot.
- Uppdaterad SDD med algoritmlösningar och lite annat smått samt uppdaterat RAD så den är bättre synkad med det vi genomfört. Bör ses över av de andra parterna för att dubbelkolla stavning etc då Gustaf endast tar det översta. Behöver även er syn på vilka designmönster som ni använt er av. Gustaf har bara listat de mönster han känner att han själv använt.

---

<sup>1</sup>Närvarande: Gustaf Ringius, Andreas Löfman, Robert Wennergren

- Fixat brus i ljudspelningen som Felix påpekat
  - Implementerat ljudhanteringen i GUIt. GUIt kallar alltså på soundController och ändrar vad som spelas. EN fundering dock om det är okej att hela tiden kalla på getInstance. Behöver diskutera om det är effektivare att möjligtvis spara en referens konkret i klassen. Negativa punkter angående detta är att det finns ju endast en instans av SoundController vilket gör att den inte behöver ha reda på en viss specifik instans. getInstance kanske inte behövs användas alls då det möjligtvis kan vara så att man kan klara sig med att göra allt i klassen statiskt så man endast gör statiska anrop och inte gör någon instans alls av klassen. Funderingar?
2. Andreas har:
    - Implementerat algoritmen för randomGenerering av banorna som fungerar som den ska.
    - Implementerat under mötet med närvarande skalning i genereringen.
  3. Robert har:
    - Har försökt lösa deadlocks som blir i små banor med många noder. Har även jobbat med gult ljus för att kunna använda bilar risktagande så att den enskilda bilen även påverkar trafiken och flödet av bilar.
  4. Findbugs är otroligt gnällig men vi har inga högprioritetsproblem i projektet. Gustaf har provat att byta ut Math.Random som den klagar på men då klagar den på att varken SecureRandom eller Random är säkra att instasiera så borde gå bra att ha vår implementation med tanke på att inget annat duger heller.
  5. Shortest Path
    - Vår implementation av sökalgoritmen är väldigt TUNG för att hitta den kortaste vägen då den bygger på en pseudoversion av Dijkstra's vilket innebär att den är rekursiv och kollar alla vägar, inkl de som inte leder till slutmålet. Detta görs för alla bilar när de åker hemifrån och med många korsningar/vägval och många bilar så blir denna sökning så tung så den låser nästan alla resurser på även den starkaste dator i gruppen som är Gustafs med 12 tillgängliga trådar, 32 GiB minne etc. Så vi beräknar alltså INTE shortest path i varje nod då det skulle bli alldeles för tungt för våra system. Möjligtvis i framtiden med en optimerad sökfunktion.
    - Möjligförbättring är att eliminera all kopiering av vår specialdatastruktur PATH. Möjligt att man skulle kunna skriva algoritmen i ett annat språk som hanterar rekursionen bättre och där man kan slippa kopieringen. Haskell? C?
  6. Vi kommer lägga vårt projektet för rättning i Master branchen och sedan fortsätta utveckla i develop-branchen. Så vi måste merge:a in develop i Master innan rättningen och sedan har joachim något att undersöka där.
  7. Redovisningspunkter som vi vill/kan demonstrera.
    - Generera banor.
    - Söka den kortaste vägen
    - Sätta alla ljus till gula och se hur de individuella entiteterna kan påverka trafikens flöde. Demonstrerar hur bilar har olika personligheter och att vissa tar risker vid gult ljus när andra stannar vid dessa och blokerar.
  8. Funderingar till möjligheten att ladda en karta. Vad behöver sparas och i vilket format? kanske enbart behöver spara matrisen i en enkel text-fil? Ska man spara det i ASCII eller ska man försöka göra något binärt format? Prioritet?

9. Styra trafikljusen. Detta är en viktig funktion för själva presentationen.
10. Måste se till så Robert får lagt in mer kodrader eftersom han ligger i riskzonen.
11. Nästa möte
  - Gustaf: Fortsätta med dokumentation och tester samt findbugs.
  - Andreas: Ordna viktningen i grafen (möjligen kolla om man kan måla/färga pathen dynamiskt medan den söker för att det är kult)
  - Robert: Rullande ljus och möjligt tester. Stänga av en väg: Låta noden se till att veta vilka vägar som är avstängda och sedan räkna med getPath på bilar som kommer till den avstängda vägen för att tänka om pathen bilen vill köra. Bilar som är på en väg när den stängs av får skickas vidare ut men inga nya får komma in på vägen i framtiden.
12. Handledarfrågor
  - Generering av banorna. Shortest path blir för tung att beräkna med för stora banor.
  - Findbugs tankar nu när vi vet vad den kollar och felen som den skriver ut. Kanske även visa några utav dessa för att få ett hum om “varningarna” är relevanta att lägga tid på.