Uge 1:

Lær en masse teori og praksis vi har brug for, for at lave projektet:

- Læs artiklen om Scotty og få litteratur fra Clemens om malleable software generelt
- Crash kursus i Linux, C og pointer programmering

Uge 2:

- Få sat os ind i GTK biblioteket, og lav en basic applikation som vi kommer til at teste vores værktøjer mod resten af projektet
- Få sat os ind i GObject biblioteket som GTK bruger Et åbenbart ret cursed library der simulerer objekt-orienteret programmering i C (ikke c++, men bare vanilla C). Hvis tid og hvis muligt, se om det er muligt at lave et objekt i GObject der kalder kode fra et andet programmeringssprog (og kode fra et andet programmeringsprog der kalder metode på et objekt i GObject). Åbenbart trods hvor forfærdeligt GObject er, så er det godt netop fordi det kan virke som et "bindeled" mellem forskellige sprog.

Uge 3:

I Scotty bliver der introduceret 5 abstraktioner til at ændre et program mens det kører:

- Window and Widget hooks: Brugeren kan lave kode der kan reagerer på events som "vindue åbner, widget bliver skabt / destrueret, widget opdaterer sin grafik" ect
- Event funnels: Hvor der i programmets underliggende kode bliver lavet events der kalder bestemte funktioner, kan brugeren erstatte de events så de kalder ens egen funktioner i stedet
- Glass Sheets: Brugeren kan lave et vindue/widget der kører *ovenpå* et eksisterende vindue/widget. Gennem det kunne de f.eks tilføje undertekster til en media-afspiller der ikke understøttede det.
- Object proxies: Indtil videre har vores arbejde med f.eks event funnels rettet sig mod *klasser generelt*, ikke enkelte instantieringer af de klasser. Men tit er det netop enkelte instantiering af klasserne man vil ændre, f.eks en enkelt knap inde i programmet. Object proxies arbejder videre på vores tidligere arbejde og tillader brugere at ændre i en enkelt instantiering af et objekt. Rent lavpraktisk bliver det gjort ved at lave en ny klasse der perfekt matcher instantiering af objektet, og så ændre på den klasse.

- Code inspection: Forestil dig debug-view fra google chrome, men inde i applikation til din computer. I Scotty skriver de intet om at direkte inspektere koden, men i stedet mere visuelle værktøjer som:
  - Hierarchy browser (udforsk hierarkiet / træet af widgets og vinduer)
  - Object inspector (lidt ligesom du kan "inspect" elementer i google chrome)
  - Widget picker ("maps a click in the interface to the code object backing the clicked widget")
  - Interactive interpreter: En lille konsol inde i programmet der har adgang til vindue hiarkiet og globale variabler

Jeg tænker lidt de her grovt kan deles op i tre fundamentale problemer der skal løses:

- Hvordan overskriver vi / får fat i "hooks" fra applikationen?
- Hvordan læser vi kontekst/indhold fra applikationen? F.eks. læse teksten på et bestemt label, bredden af en bestemt widget, ect. Den her kontekst får man højest sandsynlig nok brug får til at skrive sin egen kode.
- Hvordan viser vi nyt grafisk indhold på skærmen? (Får vi både brug for til Glass Sheets og potentielt Code Inspection)

I denne uge kunne vi potentielt dele os op og begynde at undersøge én af dem her hver. Her i processen er vi også nødt til at finde ud af et meget vigtigt spørgsmål: Hvad præcis "er" vores applikation overhovedet?

- Er det en ny version af GTK der automatisk vil understøtte vores features så længe programmet bruger denne version? (det ville være simplest at udvikle, men potentielt begrænsende for brugeren. Potentielt kunne det dog virke lige meget hvilket sprog de brugte! Og det ville nok kun kræve at GTK blev rekompileret og relinket med programmet, ikke at hele applikationen bliver rekompileret)
- Er det en patch af programmets source code der bygger et nyt program med vores features?
- Er det et program/process der kan gøre alt det her uden nogensinde direkte at modificere

Når du downloader et open-source program med din packet manager eller med deres installer, så downloader den for det meste ikke source koden. Vi er også nødt til at begynde at finde ud af, hvad rollen af source koden skal være. Måske skal vi antage at programmet ligger på github, og så skal brugeren bare give et link til github repository, og vores program finder ud af resten?

Uge 4:

Mød og begynd at find ud af, hvordan vi har tænkt os at gøre det her i grove træk, og hvor lang tid det kommer til at tage. Syntes vi det virker så overkommeligt, at vi også kan gøre det i QT, eller skal vi holde os til GTK?

Gå til værks med at implementere Window og Widget Hooks, og Event Funnels

Uge 5:

To be continued.