# GUI implementering

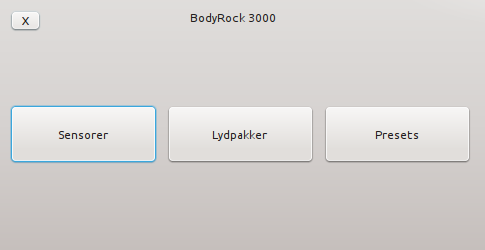
I dette afsnit kigges der nærmere på implementeringen af den grafiske brugerflade. Denne er opbygget i programmet QT. Det overordnede design af GUI’en er opbygget som et klassehieraki. Det vil sige hver klasse har sit eget vindue; oprettelse af ny sensorkonfiguration er for eksempel en klasse for sig selv der laver et nyt vindue for dette.

Derved oprettes og nedlægges der objekter af de forskellige klasser og tilsammen danner det rammerne for systemets grafiske brugerflade.

## Overordnet opbygning

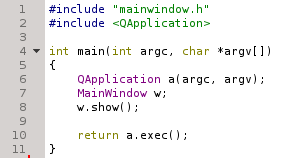
Den grafiske overflades funktion er at give brugeren et nemt anvendeligt værktøj til at håndtere og redigere indstillingerne for systemet. Derfor er der designet en overflade der sortere unødvendige ting fra når brugeren fortager et valg. Hovedmenuen består af 3 knapper der giver brugeren adgang til at konfigurere sensor konfigurationer preset og lydpakker.

### Hovedmenuen



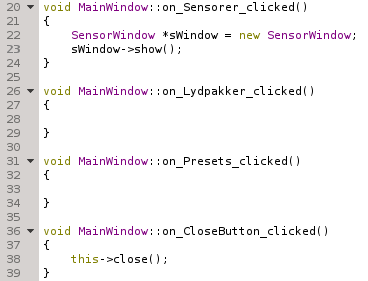
Figur 1 Hovedmenu

Dette billede fremkommer ved hjælp af QT’s UI funktion, i main oprettes der et objekt af klasse mainWindow hvis konstruktor laver et setup af et UI der designes i QT’s ui designer. Der kaldes hefter en show funktion som vist på billedet her under



Figur 2 Main i QT

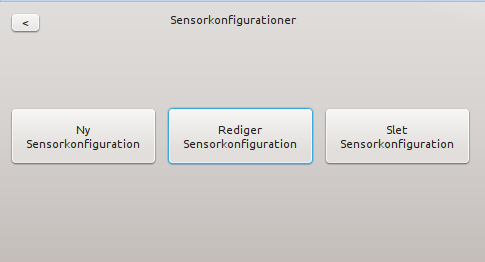
Selve designet for klassen mainWindow laves inde i sin egen klasse, her oprettes de forskellige knapper via QT’s design funktionaliteter og der kodes hvad funktion hver enkelt knap har.



Figur 3 mainWindow knappe funktionalitet

### sensorWindow

Som ses på billedet herover anvender vi det beskrevede klasse hierarki, der oprette instanser af et objekt ved klik på den valgte knappe. Klikker vi for eksempel på ”Sensorer” knappen i hovedmenuen kaldes eventet i linje 20-24 i figur XX her oprettes der en pointer der peger på en ny instans af SensorWindow. Den kalder herefter en show funktion, som er en del af QT’s funktionalitet, der herved viser det nye UI window.



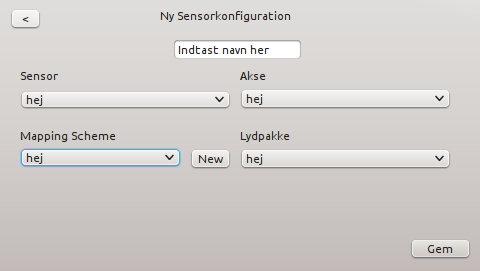
Figur 4 Den grafiske overflade til SensorWindow

Dette UI har lidt samme funktionalitet som mainWindow, det giver brugere tre muligheder for at vælge hvad de vil foretage under sensorkonfigurationer.

### NySensorKonf

Der tages udgang i oprettelse af en ny sensorkonfiguration, hvor der ved klik på denne knap sker et lignende event kald som i mainWindow.

Der oprettes herved det nye ui for NySensorKonf på samme måde, som der blev gjort ved SensorWindow. Det nyoprettede ui ser ud på følgende måde:



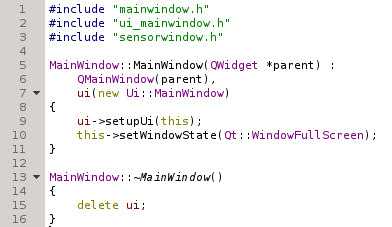
Figur 5 Den grafiske overflade til NySensorKonf

I dette ui har brugeren mulighed for at indstille de fire parametre der danner grundlag for en sensorkonfiguration

1. Senor: hvilken type sensor vi anvender
2. Mapping scheme: hvilken tone modulation der er tilhørende
3. Akse: valg af hvilken akse fra sensor der skal anvendes
4. Lydpakke: hvilken lydfil der skal spilles med

Efter parametrene er valgt gemmes de i en sensorKonf som medsendes gennem messageHandleren over til controller klassen og videre ned til Databanken.

Oprettelsen og nedlæggelsen af de enkelte windows sker i hver enkelt klasses konstrukter og destruktor

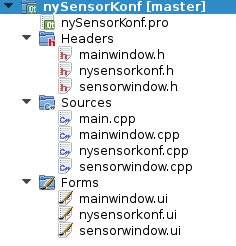


Figur 6 Konstruktor og Destruktor for mainWindow

Her anvendes QT’s funktionaliteter i forhold til oprettelse og nedlæggelse af UI elementer. I konstruktoren ses oprettelsen af vores UI der tilhøre mainWindow klassen. Der anvendes setupUI() , som laver en opsætning af vores designede widgets i QT design værktøjet.

## Andre design og implementerings overvejelser

Valget at implementere vore GUI som en form for layer programmering er valgt for overskuelighedens skyld. Dette er gjort ved at oprette hvert vindue i sin egen klasse og derved skabe muligheden for at vise og lukke vinduer ved hjælp af klassens konstrukter og destruktor der opretter og nedlægger instanser af klassen.



Figur 7 Oversigt over de implementerede klasser

I figur XX ses en oversigt hvor der er implementeret sensorkonfirgurations delen. Der er en .ui, en .h og en .cpp fil til hver klasse.

### MessageHandler

Herudover anvendes der en klasse messageHandler som står for kommunikationen mellem GUI programet og controller klassen. I denne klasse laves der funktioner der henter, gemmer eller sletter data nede i databanken.

Disse funktioner kaldes derved fra GUI klasserne, det vil sige i nySensorKonf konstruktoren laves der et kald til en funktion i messageHandleren der går ned og henter den information der skal til, at oprette en ny sensorkonfiguration (Sensor type (hardcoded), Akse (hardcoded), Mapping Scheme og Lydpakke). Med hardcoded menes der at disse er forhåndsbestemt og derfor ikke hentes i Datastorage.

Disse informationer hentes som lister, da der kan være flere Mapping Schemes og Lydpakker at vælge mellem, de indsættes derfor i en dropdown som vist i FIGUR XX. Her har brugeren muligheden for at vælge sine ønskede konfiguration og herefter gemme dem, hvor der igen laves et funktionskald til messageHandleren denne gang for at gemme de valgte konfigurationer i Datastorage.

Konfigurationerne sendes med som strings der bliver identificeret nede i Datastorage og herved vælger og gemmer de ønskede konfigurationer.