

VFU Projekt Rapport:

VFU-projektet består av två applikationer, den ena är en JavaFX-applikation, den andra är en Androidapplikation. I JavaFX finns fyra klasser:

DataLogger är en klass som består av en constructor och 8 get-metoder.

FXMLdocument består av en layout och har följande metoder:

FXMLDocumentController.java

- handleReadAction
- readFromFile
- setupZooming
- singlePlot
- autoPlot
- clearscreen.

JavaFX Application class autogenereras av JavaFX.

I FXMLController class finns en initialiseringsmetod som initialiserar både color picker, och mouse Moved eventhandler. Det finns en knapp som läser in textfilen från file explorer, samt en metod för zoomning.

I läsmetoden läses hela textfilen innehållet allt på en gång, som sen anpassas till positionen i DataLoggers constructor och sparar den till en arraylist. Sedan kan innehållet av textfilen hämtas med hjälp av en get-metod.

I setup zoomning-metoden, skapas först en ny anchorPanel, som har nya noder spara egenskap av den nod som används för senare manipulation . Sen skapar jag tre Mousehandler som hantera tryggar, drar och släpp event. I MousePressed metod först spara jag position av där man klicka. Sen skapar man en ruta (rectangle) som vara ihop med en motsvarande bredd och längd i förhållande till scenen. I Mouse Dragged method fortsätter man spara mouse positionen, och hittat alltid den störst värde mellan aktuella position och motsvarande scenens position. I mouseReleased metod skapade jag en zoom rutan med ny koordinator punkter i den referens koordinatsystem. sen lägga till zoom animerad KeyFrame. den hela setupZoomning method returnerar en anchor panelen. (Dessa metod är hänvisa till en open source projekt i github som heter [JFXUtilsExamples <version>4.11</version> av Jason Winnebeck](#)) sen filtrera man bort gamla mouseEvent som man vill applicera ny zoomning.

I singlePlot metod skapar man första åtta scatterchart series sen lägga man till ny scatterchart.data som hämta från getChannelsdata() observable arraylist iterator.next, tillsist, radera man den gamla node i getChannelData() observable arraylist. Sen manipulera varje kanelen t.ex. ändra färger osv...

I AutoPlot metod första rensa man grafen, och återställa scenen. Sen lägga man till de olika tidslinjer för valda kanaler. Sen starta man animering.

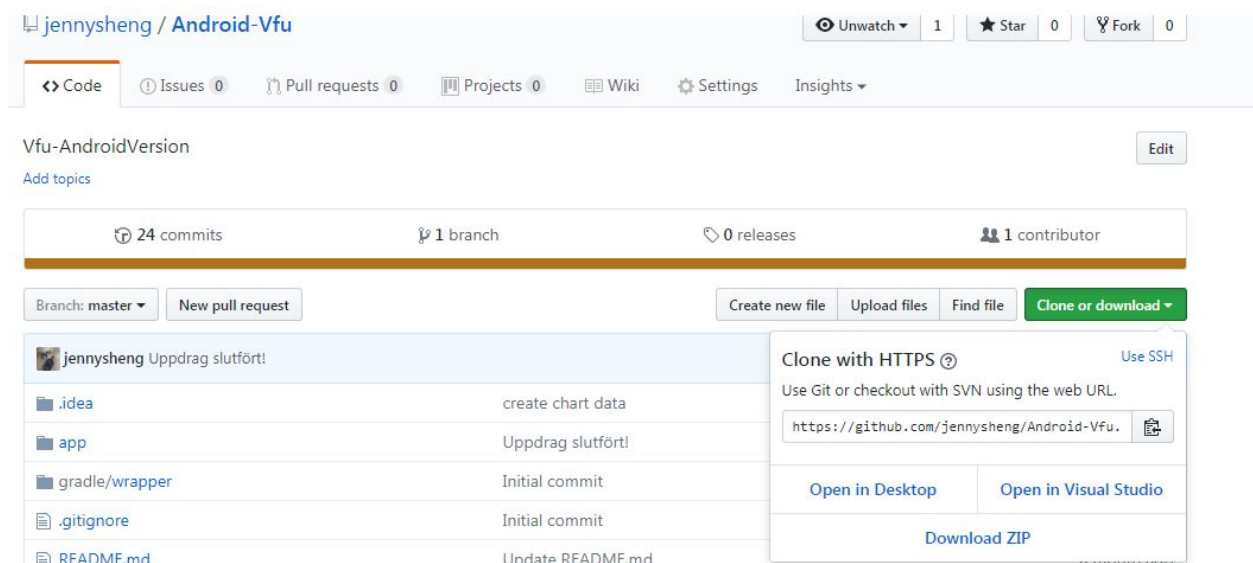
Sen rensar man scene igen.

I Android delen är man läser från SD card, och jag använda en graf bibliotek som heter `'com.github.PhilJay:MPAndroidChart:v2.1.6'`

Och det finns tre klasser: Datalogger och FileIO och Scatterchart.

Man använder bundle som passera värdena mellan klasserna. Sen skapar man Scatter DataSet som motsvarar series in java fxml. Sen lägga till den valda kanalen i scatter chart. Och Sökbart för att justera animeringshastighet. Zoomning Möjlighet finns i inbyggda bibliotek.

Installation av mina två projekt: första, går till varje projekts githubs sidan där står clone or download:



Kopiera den link och går till netbean eller Android: klick på git sen clone, sen kopiera den länk till popup fönster, så min projekt kommer att finnas på den lokal dator.

MVH

Jenny sheng

2017-10-09

