

Innlevering 2 - PG4600 Mobil Utvikling

av Eivind Vegsundvåg

Innledning

Løsningen leveres som et modulært Gradle-prosjekt. Skjermbilder finnes i mappen “screenshots” i roten av modulene.

Del 1 - Kim's Game

Innledning

Applikasjonen er laget i JavaFXMobile. Dette er et rammeverk av JavaFXPorts, som sikter på å gjøre det like lett å skrive en applikasjon for skrivebord og mobile enheter. Det er også slik at rammeverket gjør at man kan bruke nøyaktig samme applikasjon på forskjellige enheter, inklusivt iOS, selv om jeg ikke har testet å bygge for denne type enhet.

Applikasjonen installeres på en tilkople Android-enhet ved å kalle på Gradle med målet “androidInstall”. For å kjøre som Java-applikasjon startes klassen KimGame.

Fordeler

1. Applikasjonen følger prinsippet “Write Once Run Anywhere”. Den fungerer like godt på mobil, som på PC.
2. JavaFX leverer et hendelses-drevet rammeverk som gjør det enkelt å bestemme livssyklus for applikasjonen.

Ulemper

1. Som punkt 1 i fordeler: bortsett fra at den fungerer like dårlig på alle enheter. Porting av grensesnitt mot forskjellige enheter har hindret meg i å levere optimal opplevelse på noen som helst av dem. Dette er muligens løselig, men jeg har ikke funnet ut hvordan.

Problemer

Applikasjonen kan ofte vise grafiske artefakter, og disse fikses ved å endre orientasjonen til enheten. Dette er problematisk ettersom applikasjonen egentlig er ubrukelig i Landscape mode.

Komponenter som JavaFX sin ListView er dessverre ikke mulig å sammenligne med Android sin ListView, og støtter ikke adaptere. Dette er skuffende ettersom ChoiceBox og lignende har [StringConverters](#).

Konklusjon

JavaFX løser veldig mange av problemene jeg oppfatter i Android. Dessverre er det slik at platformen til JavaFXMobile gjør alle de samme forutsetningene som man kunne forventet på en skrivebordsapplikasjon. Dette medfører at man i JavaFXMobile bruker mer tid på tilpasning til mobil enn på utvikling av applikasjonens funksjoner. Løsningene jeg har implementert for dette er heller ikke gode, og applikasjonen ser derfor forferdelig ut. Det er

derfor ikke helt utenkelig at tiden man sparer på delt kodebase uansett må brukes på å skalere applikasjonen riktig. Jeg håper derfor at JavaFXPorts porter UI-komponentene til å skalere som man kunne forventet på en mobil enhet.

Del 2 - Unesco Browser

Innledning

Jeg har valgt å lage en kartapplikasjon som laster ned og viser frem alle UNESCO World Heritage sites på et kart. Applikasjonen skal være enkel, men samtidig fremstå som komplett.

Forutsetninger

Applikasjonen krever at brukeren kjører installasjon av Android med Google Mobile Services. Applikasjonen krever også at enheten den kjører på er API level 22(Android 5.1)

Jeg har ikke valgt å ta hånd om tilstand, ettersom markører hentes fra database ved ny start. Jeg har heller ikke valgt å legge noen form for innsats i å skjule API-key for Google Maps.

Problemstillinger

- Jeg har ikke satt noen logo for applikasjonen. Dette skyldes at UNESCO ikke tillater noen form for bruk av emblemet deres uten tilsøkt tillatelse. Se <http://whc.unesco.org/en/emblem/> for mer informasjon.
- UnescoMapWorker skulle egentlig vært en Service, ettersom den oppfyller to oppgaver samtidig.
- Ved valg av "Refresh" i sidemenyen, blir denne permanent markert.
- Ved visning av informasjon for kartmarkør, blir ikke bildet satt inn før ved andre trykk. Sist hentede bilde henger også igjen for nye markørvalg. Jeg antar at dette skyldes at MapInfo Views ikke oppdateres etter at de vises.