



GÖTEBORGS UNIVERSITET
IT-FAKULTETEN
INSTITUTIONEN FÖR TILLÄMPAD IT

Android & java projektrapport

Individuellt projekt, 7,5hp

Fördjupning i programmering
TIG167-H18

Emil Tigell

Datum: 19-02-03

Författare

Emil Tigell

gusemilti@student.gu.se

Innehållsförteckning

Contents

Innehållsförteckning	2
1. Projektet.....	3
2. Tools	3
3. Developer documentation.....	4

1. Projektet

Mitt projekt är en ranking applikation för spelet Civilization 5, spelet som är gammalt och inte längre utvecklas har inte några inbyggda funktioner för att ranka spelare som deltar i multiplayer spel. Tanken med mitt projekt är därför att göra en applikation där spelare sparas i en databas efter att de deltagit i en match. Informationen från denna databas ska sedan skickas i JSON-format från en webserver till androidappen som tar emot informationen och presenterar den. I appen var från början tanken att följande funktionalitet skulle finnas med:

- Lista med alla spelare sorterad efter deras nuvarande rank
- Lista över senaste spelade games
- Spelarprofiler med ranks, ratings, spelade games mm.
- Sökfunktion för att kunna söka efter en enskild spelare eller ett enskilt game.

Av detta kom det mesta med i slutversionen, sökfunktionen fungerar dock endast på spelare och inte games. Den nuvarande listan är inte heller sorterad efter ranks. Listan över spelade games finns för tillfället endast på varje spelares profil och inte som en enskild lista som var tanken från början, det är också därför det inte går att söka efter matcher.

Projektet har tyvärr dragit ut på tiden och jag fastnade helt och hållet under juluppehållet med databas-delen av webservern som var betydligt mer komplicerad än jag först hade trott. Detta ledde till att jag i princip gjorde klart hela klientsidan innan jag hade börjat ordentligt med webservern som endast hade ett skelett med grundläggande klasser osv. På grund av detta är inte projektet slutfört, android-delen av projektet har för tillfället hårdkodade objekt som fungerar för att visa en prototyp av applikationen. Klienten tar alltså inte emot någon JSON för tillfället. Webservern är helt och hållet klar för första versionen av applikationen och skickar riktig JSON med alla spelare, vilka matcher de deltagit i och vilka andra spelare som var med i dessa matcher.

En andra version skulle dock vara att endast skicka de spelare som appen frågar webservern om istället för att skicka allt på en gång, men på grund av samma orsak som skapade fördröjningen var detta inte möjligt i den första versionen då klienten redan var helt klar när webservern byggdes.

Klientsidan innehåller är mestadels ganska basic, jag råkade dock på en intressant youtube-serie om androidutveckling när jag kodade som visade en intressant scroll-animation baserad på en ganska enkel algoritm. Den använde jag och finns att hitta i listadeptern för "players".

2. Tools

Det enda verktyg som jag har använt för första gången i projektet är Android Studios och medföljande emulatorer för att köra applikationen. Jag tycker verktyget var väldigt smidigt

och eftersom jag aldrig har använt mig av en IDE tidigare var det lite ovant att så mycket av kodandet försvann, jag kommer nog att hålla mig borta från IDE:s ett tag till om det inte specifikt är för android då jag fortfarande känner att jag har relativt dålig koll på importer och paketstrukturer. Anledningen till att jag gillade Android Studios väldigt mycket var att den skapade ett GUI åt en som uppdaterades under tiden man arbetade så man hela tiden snabbt kunde kolla att koden blev som man tänkt. En stor nackdel med Android Studios för mig var dock att jag inte kunde köra det hemifrån på min stationära Windows-dator på grund av problem emulator-delen. När jag försökte köra emulatorn crashade programmet med ett felmeddelande om hårdvaruacceleration som man behövde lösa genom att göra ändringar i datorns BIOS, eftersom jag inte hittade några instruktioner för just mitt moderkort så avstod jag helt från att arbeta med klient-sidan på min stationära dator.

Övriga verktyg som använts är Atom(editor), java(developer kit) och sqllite3(databaser) men eftersom jag har arbetat med dessa verktyg en del tidigare och är bekväm med dessa har jag inte några särskilda reflektioner. De API:n som jag använt under projektets gång har varit winstone, sqllite, JSON och servlet API som alla förutom sqllite var helt nya för mig. Det har fungerat ganska smidigt men också tvingat mig att lära mig skriva enkla script för att kompilera och starta programmet eftersom det blev alldeles för många och långa satser för att testa programmet efter ett tag. Från början valde jag att arbeta med JSON API:et från oracle men jag stötte ganska snabbt på bekymmer och prövade det andra API:et från JSON-utvecklarna istället och valde därefter att köra vidare på det eftersom det fungerade bättre för mig.

3. Developer documentation

Under projektets gång har jag använt mig av er wiki med övningarna, era exempel och videos som grund. Detta fungerade mycket bra för framförallt webserver-delen. För att utveckla klientsidan hade jag problem med att bara ta åt mig ren teori och behövde göra mängder med olika små projekt, som att designa Twitter och andra liknande appar layouts eftersom det finns så mycket verktyg för just detta och ni inte kunde gå igenom allt på er wiki (fullt förståeligt). För att kunna göra detta kollade jag igenom en ett par olika youtube-serier som finns länkade här:

https://www.youtube.com/playlist?list=PLp9HFLVct_ZvMa7IVdQyUUyh8t2re9apm
<https://www.youtube.com/channel/UC8mX-MiAnXtp382gHTQg7Sw/videos>

och en till som jag tyvärr inte hittar längre.

Annan dokumentation jag använde mig av var bland annat androids egen utvecklarsida för att göra listadaptern och sökfunktionen. Stack Overflow användes också vid ett flertal tillfällen för att rätta till mindre fel eller för att se exempellösningar på olika designer jag höll på med.