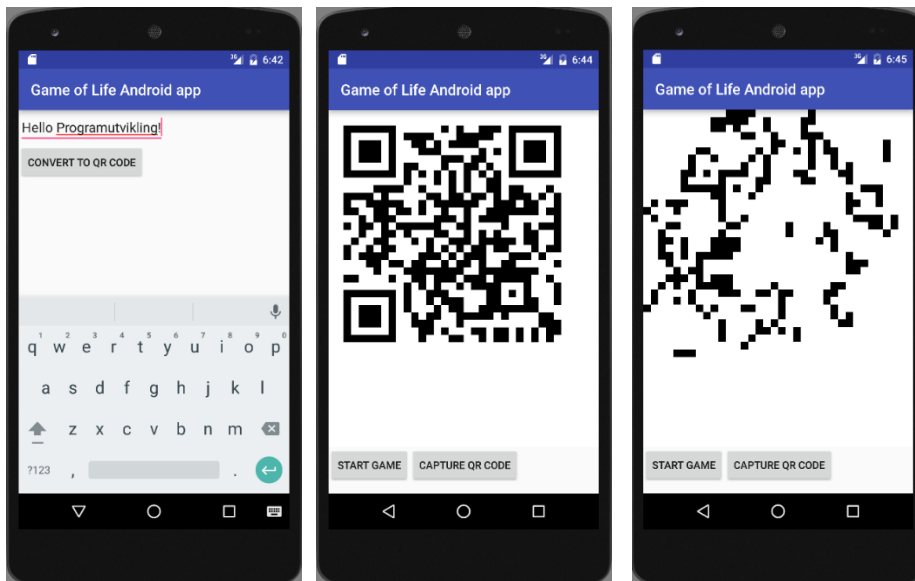


# Utvidelse: Android app

DATS1600 utvidelsesoppgave for semesteroppgaven.

I dette oppgavesettet introduseres programmeringsplattformen Android SDK. Denne oppgaven er utviklet spesielt for studenter som er interessert i app utvikling.

Det skal utvikles en *app* (kort for 'application software') rettet mot Android plattformen. Android ble valgt fordi plattformen er basert på Java. Appen skal la brukeren skrive inn en melding. Denne meldingen skal deretter blir konvertert til en QR kode som plasseres på et Game of Life spillebrett. Appen skal kunne evaluere dette spillet etter spillereglene til Game of Life. Skjermbilder som viser hvordan sluttresultatet kan se ut:



Fra brukerinntast til evaluering av et Game of Life spill. Mønsteret er en QR kode som dekodeer meldingen brukeren tastet inn.

## Introduksjon til Android SDK

Android SDK (software development kit) er en samling av programmeringsverktøy utgitt av Google. Disse verktøyene brukes i utviklingen av Android apps. Programmeringsmiljøet bruker hovedsakelig Java og XML.

Google tilbyr en 'build kit' som skal brukes til å bygge (kompilere) kodefiler til en kjørbar Android app. For en rask oversikt over prosessen til build kit'en, se:

<http://developer.android.com/sdk/installing/studio-build.html>

Å forstå alle detaljene til denne prosessen er ikke viktig på dette tidspunkt. Man må imidlertid være klar over at man bruker Google sine verktøy for å produsere kjørbare programmer og ikke Oracle sine verktøy rettet mot Oracle sin JVM. Per dags dato, kompilerer du sannsynligvis mot Dalvik (Google sin virtuelle Java maskin).

Google har ikke implementert full JDK støtte, spesielt fra nyere versjoner av JDK (som versjon 8). For eksempel, støtter ikke Android SDK, per dags dato, JavaFX biblioteket som er en del av JDK 8. Programmet dere utvikler i semesteroppgaven kan derfor ikke kompileres direkte mot Dalvik på grunn av dette.

Det anbefales at du bruker Google sin IDE for utvikling av Android apps. Android Studio er Google sin offisielle IDE for Android og tilbyr verktøyer som ikke er tilgjengelige i andre IDE'er, som Eclipse. Android Studio kan lastes ned fra følgende lenke:

<https://developer.android.com/sdk/index.html>

## Oppsett og utvikling av en enkel app

Google tilbyr god støtte for utviklere som ønsker å komme i gang med Android app utvikling. Gå nøye igjennom følgende tutorial, der du utvikler en enkel app:

<https://developer.android.com/training/basics/firstapp/index.html>

## Aktiviteter

En Android app er bygget opp i aktiviteter. Når en app starter, startes en (main) aktivitet. Som en konsekvens er ikke main prosedyren en del av en Android app. Man kan deretter opprette nye aktiviteter fra en kjørende aktivitet. Slike aktiviteter representerer derfor livssyklusen til en Android app. Før du går videre, påse at du forstår livssyklusen til aktiviteter ved å lese følgende:

<https://developer.android.com/training/basics/activity-lifecycle/index.html>

## Aktivitet – Game of Life

Det skal nå opprettes en aktivitet som representerer utførelse av spillet Game of Life. Det antas at dere allerede har implementert dette spillet som en del av semesteroppgaven. Det vil være en fordel å ha en kodelstruktur der spillklassen har minst mulig (eller ingen) avhengigheter til klasser som ikke har noe med spillet å gjøre.

### Overføring av kode

Klassene relatert til spillet, dvs. klasser som trengs for å representere spillet i appen, skal flyttes over til kodelstrukturen til appen. Det er flere måter å gjøre dette på: integrasjon med Gradle, pakking av relevante klasser i jar, eller manuell kopiering av kode.

For å unngå eventuelle kompatibilitetsproblemer, foreslås det å manuelt kopiere over kode fra JavaFX programmet deres. Det vil si: opprett modell-klasser relatert til spillet, kopier over kode fra gruppeprosjektet, og korrigjer eventuelle kompileringsproblemer (sannsynligvis mest relatert til JavaFX).

### Visning av spill

Opprett en klasse (foreslått navn: GameViewer) som representerer muligheten for å vise et Game of Life spillebrett grafisk. Denne klassen skal utvide (extends) android.view.View og overkjøre metoden onDraw(Canvas canvas). Påse at korrekte konstruktører blir opprettet for å kunne opprette et View objekt. Se dokumentasjonen for view for informasjon:

<http://developer.android.com/reference/android/view/View.html>

### Ny aktivitet

Opprett en ny aktivitet som representerer oppgaven (representasjon av et Game of Life spill).

Deretter, opprett et objekt av den opprettede klassen i XML layouten til den nye aktiviteten.

Eksempel på XML tag:

```
<include android:id="@+id/game" ... />
```

Den opprettede aktiviteten skal startes fra hovedaktiviteten (inntasting av en melding fra bruker). Dette kan for eksempel gjøres ved et knappklikk fra startvinduet, slik som i skjermbildet. For å starte en ny aktivitet, må et objekt av typen Intent opprettes og deretter kalle på metoden `startActivity`.

### Tegning av grafikk

Grafikk tegnes i `onDraw` metoden. Sett deg inn i funksjonalitet relatert til grafikk som tilbys av [android.graphics.Canvas](http://developer.android.com/reference/android/graphics/Canvas) klassen. Tegn noen enkle grafikkelementer for å verifisere at oppsettet av den nye aktiviteten fungerer.

Opprett en `draw` metode i klassen relatert til spillet med følgende parametere: `draw(Canvas canvas, Paint paint)`. Bruk denne metoden til å tegne spillebrettet til `canvas`. Du kan på dette tidspunktet opprette et mønster manuelt for testing, slik som i det andre oppgavesettet i semesteroppgaven.

### Animasjon

Android SDK tilbyr god støtte for animasjon, spesielt relatert til animering av grafiske objekter og GUI elementer. Dette er et avansert alternativ som bare burde velges av studenter som ønsker å sette seg inn i Android sin støtte for animasjon og grafikk. Mer informasjon:

<http://developer.android.com/training/building-graphics.html>

Et enklere alternativ er å manuelt håndtere animeringen av spillebrettet. Først, må en boolsk variabel opprettes i `GameViewer` klassen som bestemmer om `canvas` skal animeres eller ikke. Hvis brettet skal animeres, betyr det at `canvas` skal tegnes på nytt etter et `nextGeneration` kall til spillobjektet. Følgende pseudo-kode representerer `onDraw` metoden med animasjon:

```
public void onDraw(Canvas canvas) {
    < draw background >
    < draw game board >
    if (isAnimating) {
        < next generation call >
        < put thread to sleep to cap fps >
        invalidate(); // request new draw call
    }
}
```

For å oppnå et gitt antall bilder i sekundet (fps) må tråden puttes i søvn. Hvor lenge tråden skal være i søvn må kalkuleres ut ifra hvor lenge `nextGeneration` metoden bruker på sine kalkuleringer. Funksjonen `sleep` i `Thread` klassen brukes til å putte en tråd i søvn.

### Kommunikasjon mellom aktiviteter

Det er nå (sannsynligvis) to aktiviteter i appen: en aktivitet for å sende en melding skrevet av bruker og en aktivitet for å representere spillet *Game of Life*. Det skal nå gjøres kommunikasjon mellom disse to aktivitetene. Når brukeren går til den andre aktiviteten (spillet), skal meldingen brukeren skrev inn konverteres til en QR kode. Denne koden skal deretter konverteres til et spillemønster, slik som vist i skjermbildet i starten av dokumentet.

For å muliggjøre slik kommunikasjon, må alle tilstander til et objekt som skal oversendes bli konvertert til en strøm av bytes (eng: *byte stream*). Når mottaker mottar denne strømmen, kan et nytt objekt opprettes med samme tilstander som det originale objektet. En slik operasjon kan oppnås med Javas 'serialization' objekter, som kan leses om her:

<https://docs.oracle.com/javase/tutorial/jndi/objects/serial.html>

Du har to valg når det kommer til kommunikasjonen mellom de to aktivitetene i denne oppgaven:

1. Overføre meldingen, som er av typen String (er Serializable). En slik dataoverføring ble demonstrert i Android tutorialen dere startet med.
2. Opprette et spill-objekt, med QR koden på brettet, for deretter å overføre dette objektet til den andre aktiviteten. Dette valget krever at spillklassen er Serializable.

Selv om det første valget er mest naturlig for dette problemet, anbefales det at du velger det andre valget; da vil du få nyttig erfaring med å gjøre klasser Serializable. Hvis du gjør klassen serializable kan du også senere eksperimentere med å lagre spillet som en binær fil.

For mer informasjon angående kommunikasjon mellom aktiviteter, se dokumentasjonen for Intent klassen, spesielt putExtra metodene (som f.eks. putExtra(String name, Serializable value) ).

### Konvertering fra String til QR kode

QR (Quick Response) kode brukes til å dekode data. Formatet er konstruert slik at maskiner kan enkelt lese og tolke data. QR er den 2-dimensjonale versjonen av strekkode (eng: bar code).

Håndteringen av QR data skal utføres via biblioteket zxing, som er lokalisert her:

<https://github.com/zxing/zxing>

Dette biblioteket skal inkluderes i prosjektet deres ved hjelp av Gradle, et byggesystem (eng: build tool) som automatiserer kompilering og linking til eksterne biblioteker som zwing. Gradle, som er integrert i Android Studio, kan leses om her: <http://gradle.org/>

I filen build.gradle, som skal være en del av Android Studio prosjektfilene, legg til følgende linje i blokken dependencies:

```
compile 'com.google.zxing:core:3.3.0'
```

Oppdater prosjektet slik at zwing er integrert.

En QR kode kan representeres i programmet via et objekt av typen QRCode, en klasse som ligger i pakken com.google.zxing.qrcode.encoder. Bruk [dokumentasjonen til zwing](#) til å finne ut hvordan man oppretter et slikt objekt fra en String.

### Konvertering fra QR kode til Game of Life mønster

Vurder hvor i kodenstrukturen QR koden skal konverteres til et mønster. Selve konverteringen kan enkelt utføres ved hjelp av et objekt av typen ByteMatrix, som ligger i samme pakke som QRCode. Et slikt objekt kan hentes ved hjelp av metoden getMatrix() i QRCode.

Etter at oppgaven over er utført, kan konverteringen (String til QR bilde) testes. Kjør appen i en virtuell enhet (virtuell Android mobiltelefon) på din datamaskin, skriv en melding, og konverter til spillemønster. Når QR koden vises på skjermen (som et Game of Life spillemønster), kan du verifisere denne ved å lese QR koden med din mobiltelefon (bruk en app som tolker QR koder). Din mobiltelefon skal nå vise meldingen du skrev inn.

### Videre utvikling

Etter utførelse av oppgavene over skal du ha implementert et minimalt program som dekker minimumskravene for oppgavesettet. Programmet burde imidlertid utvides slik at det kan enkelt brukes av sluttbrukere. Følgende aspekter burde vurderes og implementeres:

- **Automatisk celledørrelse.** Størrelsen til en QR kode er avhengig av dataen som dekodes. Desto mer data, desto større blir QR koden. Programmet burde være implementert slik at mønsteret er tilpasset skjermen. Dvs. juster cellSize variabelen etter størrelsen til QR koden.
- **Justering av celledørrelse.** Ved utførelsen av spillet, vil levende celler sannsynligvis gå utenfor skjermen. Det kan være naturlig å la brukeren 'flytte' på brettet (dvs. endre hvilken celle som vises på midten av skjermen) eller la brukeren kunne endre på celledørrelsen (f.eks. via en slider).
- **Visuell representasjon.** Den visuelle representasjonen av spillet vist i skjermbildet i starten av dokumentet er ikke spesielt attraktivt. Bruk av fargeoppsett vil gjøre representasjonen mer attraktivt. Alternativt, kan bruker velge en farge eller et fargeoppsett. Som alltid, tillates det bruk av fantasi i det grafiske oppsettet. For eksempel, tilbyr Android SDK bra støtte for 3D grafikk.

En funksjonalitet som burde være støttet er å konvertere et fotografi, tatt med Android enheten, til et GoL spillemønster. Anbefalt fremgangsmåte er å første konvertere bildet til et binært bilde ved å bruke 'halftone' teknikken. Dette binære bildet, der hver pixel enten er 0 eller 1, kan overføres direkte til et GoL spillebrett (pixler korresponderer direkte med celler på brettet).

Hvis halftone fremgangsmåten følges, vil jeg anbefale å implementere teknikken dithering. Denne teknikken gir gode resultater og burde være relativt rett-frem å implementere. Floyd-Stainberg dithering er her mest anbefalt:

[https://en.wikipedia.org/wiki/Floyd%E2%80%93Steinberg\\_dithering](https://en.wikipedia.org/wiki/Floyd%E2%80%93Steinberg_dithering)

<https://developer.android.com/training/camera/photobasics.html>

## Besvarelse

Android app prosjektet er sannsynligvis et separat kodeprosjekt fra JavaFX prosjektet. Android prosjektet kan inkluderes i semesteroppgaven i en separat mappe.

Følgende må være inkludert i mappen:

- Java prosjektet (kodefilene er viktigst)
- Dokumentering av appen i en PDF fil. PDF filen skal presentere de ulike funksjonene (aktivitetene) som støttes av appen. Bruk skjermbilder/fotografier i presentasjonen.