



Ett enkelt Androidspel, tutorial

Detta är en introduktion till hur du kommer igång med Android-programmering för smartphones. I huvuduppgiften gör du ett enkelt spel, "Vilket ord tänker jag på?", och laddar ned det till en telefon. Vill du gå vidare på egen hand hittar du mer information på <https://www.kth.se/social/page/komma-igang-med-android/>

Första applikationen, Hello World

Öppna utvecklingsmiljön Eclipse. Välj från menyn: File > New Project > New Android Project och klicka sedan på Next.

Fyll i följande i den nya vyn:

- Project Name, projektnamnet, t.ex. HelloWorld (inga mellanslag)
- Under Build Target, bocka för Android 2.1-update 1
- Under Properties, Application Name, ange applikationens namn, t.ex. Hello World
- Properties, Package name, ange t.ex. se.kth.dittnamn (en namnrymd för dina klasser)
- Properties, Create Activity, ange namnet på huvudklassen (Activity), t.ex. HelloWorldActivity
- Properties, Min SDK Version, ange 7

Klicka på Finish. Du ser nu ditt nya projekt till vänster i Eclipse. Öppna projektet (klicka på +) och därefter src o.s.v. tills du ser din Activity-klass (här: HelloWorldActivity). Öppna filen.

Det enda denna kod gör är att skapa en vy (vars layout definieras i filen `R.layout.main`).

Testkör applikationen genom att högerklicka på projektets ikon (i vänstra vyn) och välja Run As > Android Application.

En emulator startar (det tar ett tag), och som visar en rubrikrad med applikationens namn samt en hälsningsfras, som är hämtad från filen `res/values/Strings.xml`.

- Öppna denna fil i editorn, markera "hello" och ändra värdet (Value) på hälsningsfrasen till något annat. Testkör applikationen igen.

Föra över applikationen till en mobiltelefon

För att föra över ett program till telefonen måste du först skapa en applikationsfil (apk-fil) och därefter göra själva överföringen.

Skapa en applikationsfil

Högerklicka på projektet och välj : Android Tools > Export signed application package



I dialogen väljer du projekt och sedan att skapa en ny keystore. Fyll sedan i detaljer om nyckeln och skapa nyckel. Ange sist vad applikationsfilen ska heta och skapa den.

Föra över applikationen till telefonen

Paketet ska nu föras över till telefonen. Koppla in telefonen med USB-kabeln och montera den (hur anges på telefonen). På datorn, under My Computer, kommer du nu att hitta din telefon som en separat enhet. Kopiera över apk-filen du skapade ovan till denna enhet.

Avmontera telefonen (hur anges på telefonen) och koppla ur USB-kabeln. Starta, från telefonen, applikationen 'Astro File Manager'. Leta rätt på apk-filen, klicka på den och välj att öppna 'Application Manager'. Välj sedan att installera och testa applikationen.

[Om du har en egen telefon kan du ladda ned Astro File Manager från <http://www.androidfreeware.net/download-astro-file-manager.html>]

Gissa ordet

Importerera ett (halv-) färdigt projekt att utgå ifrån

Du ska nu skriva (delar av) spelet Gissa ordet. Du kommer att utgå från ett projekt där vissa delar redan är klara.

På sidan <https://www.kth.se/social/page/komma-igang-med-android/> (sök på "komma igång med android") finns ett antal projekt, i form av zip-filer, att bygga vidare på. Ladda ned projektet GuessMyWord, t.ex. till skrivbordet.

Importerera projektet GuessMyWord till Eclipse på följande sätt:

Välj från menyn File > Import.

Markera, under General, alternativet Existing Projects into Workspace och klicka på Next.

Markera alternativet Select Archive File, klicka sedan på Browse och förflytta dig till mappen som innehåller zip-filen med projektet och markera denna.

Avsluta med Finish.

För att se innehållet i projektet, klicka på pil/triangel-symbolen vid projekthuvudet.

Projektet innehåller en fil med logiken för spelet, GuessMyWordModel, och en fil, GuessActivity, som bland annat ska knyta ihop användargränssnittet med logiken i spelet.

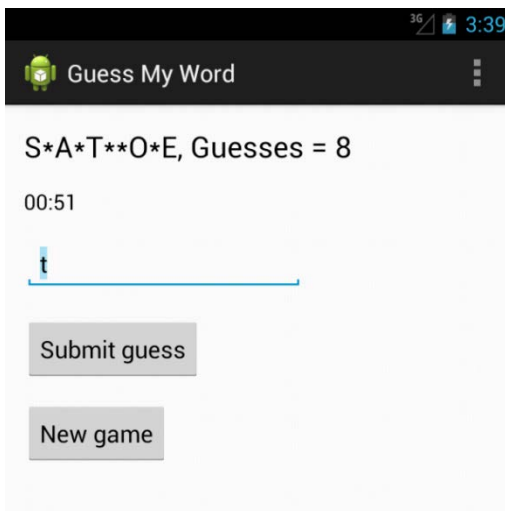
Layout

Det första du ska göra är att skapa ett användargränssnitt, d.v.s. de komponenter där användaren kan mata in text och programmet skriva ut statusen i spelet.



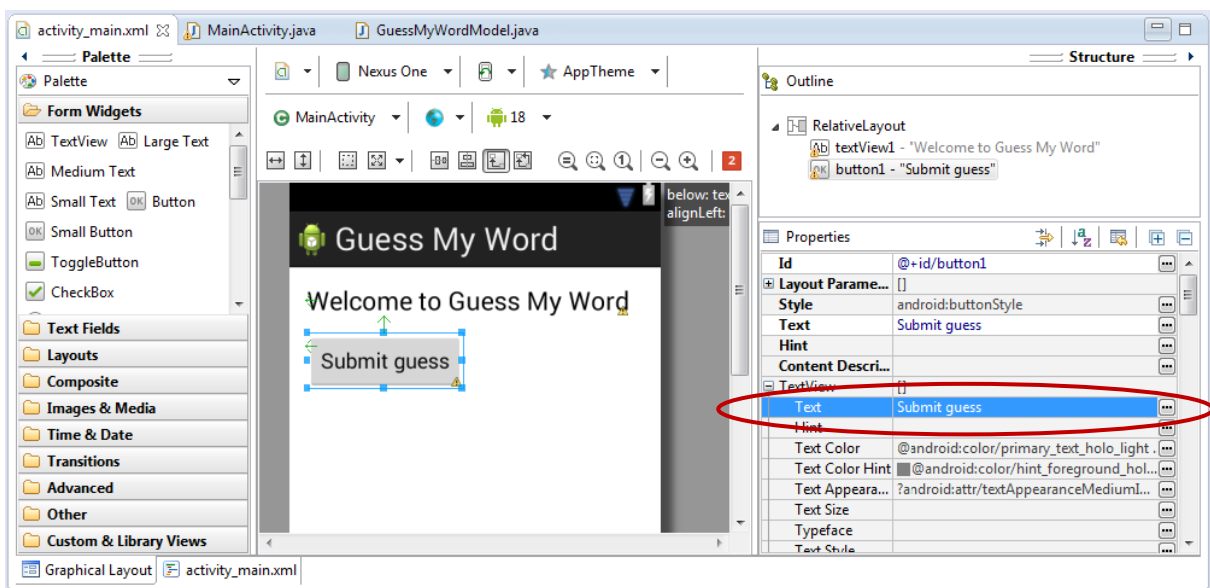
Öppna mappen res/layout och dubbelklicka på filen main.xml. I denna fil beskrivs hur layouten för användargränssnittet ska se ut. Markera fliken "Graphical Layout" för att se hur det ser ut.

För tillfället ser du endast en textvy, med en text som förklarar hur spelet ska spelas. Du ska nu lägga till en vy där användaren kan mata in bokstäver, en tidtagare samt knappar för att registrera en gissning och starta en ny omgång. Tanken är att det ska se ut som nedan när programmet körs.



Lägg till en knapp

Klicka och dra en knapp, Button, från paletten in i layouten. Släpp knappen under textvyn. Ändra texten på knappen genom att högerklicka på den och välja "Show in properties". I fönstret under layouten scollar du till "Text" och skriver in texten som du vill ska synas på knappen, t.ex. "Submit guess", se nedan.





Lägg till fler komponenter

Dra på samma sätt in ett inmatningsfält, EditText, ovanför knappen. Du hittar inmatningsfältet under Text Fields, välj den översta komponenten, Plain Text.

Lägg också till en tidtagare, Chronometer, ovanför inmatningsfältet. Du hittar tidtagaren under Time and Date.

Lägg till sist till ytterligare en knapp, för att starta om spelet, längst ned.

Det bör se ut ungefär som på föregående sida.

Testkör

Vi kan nu testköra applikationen för att bättre se hur layouten ser ut. Högerklicka på projektets ikon (i vänstra vyn) och välj Run As > Android Application (det kommer att ta ett tag innan emulatorn startar).

Inget händer om du trycker på någon av knapparna, det får vi lägga till.

Logiken i programmet

Steg 1, Läsa av en inmatad bokstav

Öppna filen GuessActivity.java. I denna fil knyts användargränssnittet och logiken samman.

Börja med att "avkommentera", d.v.s. ta bort "//", på raderna 30-38¹. Kontrollera att inga röda markeringar syns i koden (i annat fall be en assistent om hjälp). Dessa rader gör att vi i programmet kan referera till de komponenter du just lagt till i layouten.

På rad 44, under "onClick" ska vi skriva kod för vad som ska hända om användaren klickar på knappen. Vi börjar med att lägga till kod som läser av vad användaren matat in och skriver ut detta i textvyn överst på displayen. Byt ut kommentaren på rad 44 mot följande text (före raden med "}").

```
String inputString = textInput.getText().toString();  
char guess = inputString.charAt(0);  
textOutput.setText("The character = " + guess);
```

Provkör igen, Run As > Android Application, och se vad som händer när användaren matar in en bokstav.

¹ Om radnummerna inte visas, placera markören i vänstra kanten av texteditorn, högerklicka och välj "Show Line Numbers".



Prova vad som händer om användaren trycker på knappen utan att ha matat in en bokstav. Vi måste tydligen kontrollera *om* användaren matat in något innan vi läser av texten. Ändra koden till (fortfarande fr.o.m. rad 44)

```
String inputString = textInput.getText().toString();
if(inputString.length() > 0) {
    char guess = inputString.charAt(0);
    textOutput.setText("The character = " + guess);
}
else {
    showToast("Please enter a letter!");
}
```

Testkör igen.

Steg 2, Spelet

Vi ska nu använda oss av spellogiken, så att användaren kan se vad hon gissat rätt hittills samt antalet gissningar. Så här ska det se ut från rad 44:

```
String inputString = textInput.getText().toString();
if(inputString.length() > 0) {
    char guess = inputString.charAt(0);
    guessModel.handleGuess(guess);

    int n = guessModel.getNoOfGuesses();
    String soFar = guessModel.getGuessSoFar();
    textOutput.setText(soFar + ", Guesses = " + n);
    textInput.selectAll();
}
else {
    showToast("Please enter a letter!");
}
```

- `guessModel.handleGuess(guess)` lägger till en gissning till modellen
- `guessModel.getNoOfGuesses()` anger antalet gissningar hittills
- `guessModel.getGuessSoFar()` ger en text där de korrekt gissade tecken syns men övriga är markerade med stjärnor

Testkör programmet igen.



Steg 3, Gratulerar

Det som saknas är att programmet talar om när man gissat hela ordet. Vi kan testa om spelet är klart med `guessModel.isSolved()`, som svar med sant eller falskt.

Lägg till följande på raden efter `textOutput.setText(...)`

```
if(guessModel.isSolved()) {  
    showToast("GONGRATULATIONS");  
}
```

Testkör igen. Om du inte sett det redan, så kan du hitta de ord som slumpas ut någonstans i en av java-filerna (fusk?).

Steg 4, Tidtagning och nytt spel

Vi vill att ett nytt spel och tidtagning ska startas om användaren trycker på knappen "New game".

Gå till raden som börjar "private class NewGameClickListener". Två rader nedanför denna rad hittar du ett nytt "onClick". Det är här vi ska definiera vad som händer när användaren trycker på denna knapp.

Lägg till följande på raden efter "onClick".

```
guessModel.reset();  
String soFar = guessModel.getGuessSoFar();  
textOutput.setText("The word: " + soFar);  
stopWatch.setBase(SystemClock.elapsedRealtime());  
stopWatch.start();
```

Denna kod startar om spelet (slumpar ett nytt hemligt ord), skriver ut stjärnor för varje bokstav samt nollställer klockan och startar den.

Vi bör också lägga till att klockan stoppas när ett spel är klart. Den tidigare if-satsen som kontrollerar om gissningen är korrekt (kring rad 44) utökas till

```
if(guessModel.isSolved()) {  
    showToast("GONGRATULATIONS!!!");  
    stopWatch.stop();  
}
```

Testkör programmet igen.

För över programmet till en telefon

Och nu till det bästa – spela på en telefon. Se instruktionerna i början av detta dokument. Prova ditt program på riktigt.



KTH Teknik och hälsa

Steg 4, Customize?

Kanske du har en egen idé om hur programmet kan förbättras. Fråga en assistent som kanske kan hjälpa dig med det.

Det färdiga programmet finns i filen GuessMyWordFinal.zip. Där har jag också lagt till att textInput rensas efter varje gissning (så att inte användaren behöver göra det själv inför varje ny gissning).



ANDROID