

# Android Dag 2 (lektion 5)

Først klar ved lektionens begyndelse

Plan:

08.00 Forelæsning (V1.01 - i plenum)

Hvordan er eksamenformen til 3-ugersprojektet i BrInt,  
og er eksamensplaceringen i slutningen af uge 4?

Hvordan viser man sin app på egen android mobil i stedet for en emulator.

Udlån af telefoner

Hvad skal vi aflevere torsdag d. 3. oktober?

Gruppekонтракт, mødeprotokol, hvad skal der ske ved fravær/ikke lavet arbejde,  
mødedisciplin, rollen som 'den sure mand'/indpisker (går på omgang)

Individuel Galgeleg-opgave

08.30 Gruppearbejde - hjælpelærere kommer kl 9

A-grupper med Magnus i V1.03,

B-grupper med Nicolai i V1.04,

C-grupper med Duran i vandrehallen

**Husk navneskilte ved hver gruppe**

Kontakt den hjælpelærer der har rettet din aflevering hvis du har spørgsmål

Snak med de andre grupper, der laver det samme projekt

11.30 Forelæsning (V1.01 - i plenum)

Hvad skal vi aflevere torsdag d. 3. oktober?

En samlet mappe med opgaver, materiale etc findes [her](#) (tilgængelig under 'links')

## Driller emulatoren?

Få fejlretning via [USB](#) (selv [på Windows](#)) til at virke med en fysisk telefon. [Ansøg](#) eventuelt om at låne en fysisk telefon.

Kan du ikke få det til at virke så prøv [Genymotion](#) - gratis til personlig brug.

# Opgaver

## Opsamling af opgave fra sidste gang

Præsenter din løsning på android-programmeringsopgaven for mindst 1 anden - og se mindst én anden løsning.


1. Vis den kørende app
2. Vis XML-layouts
3. Vis Java-koden
4. Fortæl hvad der sker hvor og hvordan du har valgt at gøre det

Et par eksempler på fine løsninger ligger [BrInt 03 - Fine\\_løsninger.zip](#). Hent dem og diskutér to og to om I er enige i deres svar på spørgsmålene.

Udfør følgende (på din aflevering fra sidst)

[Øvelse 1: Prøv nogle intents i AndroidElementer](#)

[Øvelse 2: Versionering af et projekt](#)

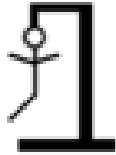
1. Ændr dit eget program til at bruge et par implicitte intents
2. Opret nogle flere aktiviteter.
  - a. Commit **først** alle dine ændringer til det lokale Git repo.
  - b. Opret så en aktivitet.
    - i. Undgå  New
    - ii. Brug i stedet New / Activity og vælg Empty Activity
  - c. Undersøger derefter hver enkelt fil for hvilke ændringer der er kommet

Prøv at bruge debuggeren

Når ud er færdig så arbejd videre på jeres eget projekt

Bemærk: \*1. Iteration Prototype sendes til kunden senest d 3/10 kl.18

Vent til kl 10 eller alle i gruppen er færdige med opgaven ovenfor (arbejd på dit eget projekt i stedet)



## Projekt galgeleg

Krav:

- I skal lave en Android-app med et simpelt galgeleg-spil
- App'en skal vælge et tilfældigt ord, skjult for spilleren og holde styr på spillerens gæt.
- Spilleren skal så gætte ordet. Hver gang han gætter forkert dukker der mere af den hængte mand op. Grafikfiler og et fungerende eksempel på programlogikken er udleveret
- Eksemplet er et **standard** Java-projekt med en main-metode (GalgelegTest) der demonstrerer hvordan man benytter programlogikken (et Galgelogik-objekt)
- Grafikken ligger i mappen GRAFIK
- Programlogikken skal holdes adskilt fra brugergrænsefladen. Du skal ikke ændre i galgelogikken!
- De funktioner der er i den udleverede Galgelogik-klasse må ikke flyttes ind i brugergrænsefladen
- I stedet skal en række aktiviteter, der bruger et Galgelogik-objekt, programmeres:
  - Hovedmenu
  - Selve spillet,
  - Hjælp
  - etc .... f.eks Highscore

## Udleverede filer

På <https://github.com/nordfalk/Galgeleg> kan du hente et standard Java-projekt med med grafik og et eksempel på en main()-metode, der benytter logikken, samt en JUnit test af logikken.

Bemærk: Du kan **ikke** lave Android-udvikling i et standard Java-projekt. Du skal oprettet dit **eget** tomme Android-projekt og lægge relevante filer fra standard java-projektet derind.

Hvis du går i stå - så kan logikken også ses i [lekt02\\_aktiviteter](#) i AndroidElementer

**Har du problemer med blank skærm i designeren når du opretter nyt projekt, eller andre underlige ting?**

Så åbn app/build.gradle og ret din compile SDK version 1 ned (fra den nyeste til den

næstnyeste). Den nyeste har SDK har ofte nogle fejl. Eksempel (fra 2018):

```
ret compileSdkVersion 28
til compileSdkVersion 27
og targetSdkVersion 28
til targetSdkVersion 27
og implementation 'com.android.support:appcompat-v7:28.0.0'
til implementation 'com.android.support:appcompat-v7:27.1.1'
```

Og tryk på “Sync project”. Hvis du får “Failed to find target with hash string 'android-27” så tryk på “Install missing platform(s) and sync project”

## Fragmenter - intro

Åbn AndroidElementer og prøv Hovedaktivitet i [lekt04\\_fragmenter](#) .

Sammenlign med [lekt02\\_aktiviteter](#).

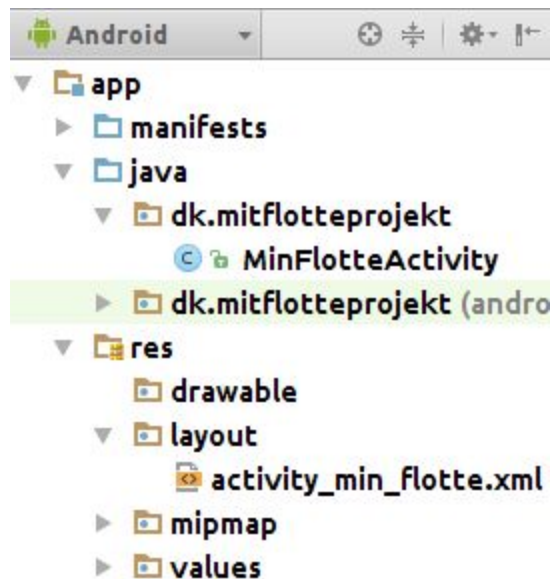
Prøv derefter [HovedaktivitetMedViewPager.java](#)

Se om du kan lave din egen Galgeleg om til at være fragmentbaseret ?

Husk at versionere dit projekt først! - og lav en kopi af dit arbejde så du kan komme tilbage.

## Kom i gang med fragmenter - kopiér fra AndroidElementer

Du har dit flotte projekt åbent og versioneret:



Åbn nu AndroidElementer og kopier pakken [lekt04\\_fragmenter](#) ind. Du skal også have Galgeleg.java og layouts lekt01\_tre\_knapper.xml, lekt04\_fragmenter.xml og logo.png ind i projektet:

Slet derefter forkerte imports fra Hovedmenu\_frag og importér den rigtige R-fil

```
import dk.nordfalk.android.elementer.R;
```

```
import lekt02_aktiviteter.*;
```

Ryd op i andre ting, der ikke er med, slet f.eks.

```
Intent i = new Intent(getActivity(), Indstillinger_akt.class);
```

```
startActivity(i);
```

og fra Velkomst\_frag slet

```
iv.startAnimation(AnimationUtils.loadAnimation(getActivity(), R.anim.egen_anim));
```

Ret derefter i manifestet få Hovedaktivitet er nævnt, eller kopier dens onCreate() ind i din egen flotte aktivitet.

## Afleveringsopgave

Aflever din galgeleg-app som en samlet ZIP-fil i Campusnet Opgaver (se krav ovenfor - i [Galgeleg forprojekt](#)).

Inkludér det lokale GIT-repo (det sker af sig selv - det er mappen `.git`, der ligger i projektmappen)

Tjek ZIP-filens størrelse - og slet mapperne `build` og `.gradle` fra ZIP-filen før du uploader