Week 3 - oefening 1: Lifecyle Activity

Doelstelling

Verschillende fasen in de lifecyle van een Activity registreren

Algemene beschrijving

Voor deze en volgende opgaves is het noodzakelijk dat specifieke opties voor ontwikkelaar op uw device of emulator ingesteld kunnen worden. Vóór android 4.2 zaten de ontwikkelaars opties gewoon in het menu, in 4.2 en hoger heeft Google dit verborgen onder 7 keer klikken op Build Number (te vinden onder Settings -> About Phone -> Buildnumber), omdat mensen verkeerde knoppen aanzetten en toen klaagden over problemen. Hierna kan u onder Settings ook 'Developer options' terugvinden.

We maken een eenvoudige Android Applicatie dat bestaat uit een Activity met één fragment met één button (en eventueel één textview). Via de button is het mogelijk de Activity (en dus ook de App) te sluiten.

Volgende versie bouwt u na.

Veel documentatie is te vinden op: developer.android.com

Voorbereiding

Maak een nieuw Android Application Project aan:

Application Name:	Demo Life cyle
Package Name:	be.howest.nmct.android.lifecycle
Target SDK:	21
Activity:	LifeCyleActivity – *layout: activity_lifecyle.xml*
Fragment:	LifeCyleFragment – *layout: fragment_lifecycle.xml*
Path:	*<zelf te kiezen>*

Design

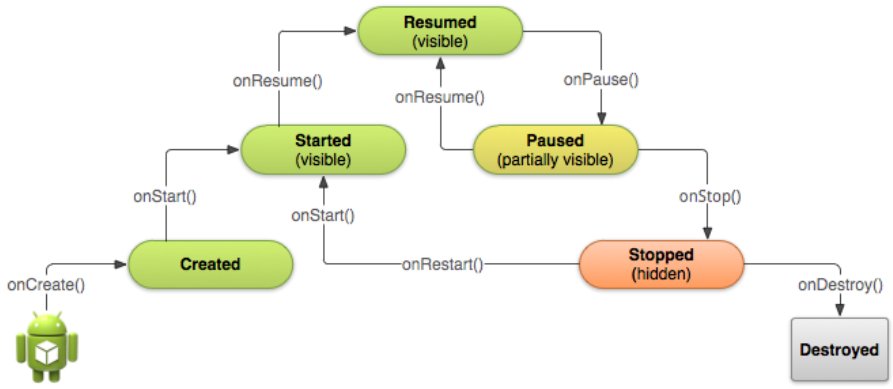
Ga na waar de xml-layout files zich situeren. Net zoals in week 2 maak je een afzonderlijke fragment-klasse (**LifeCyleFragmen.java**) waarin corresponderen layout-file (***fragment_lifecycle.xml***) aangeroepen wordt. Zorg ervoor dat de activity 'LifeCycleActivity' jouw fragment-klasse oproept.

Plaats op het fragment één button. Wanneer de gebruiker op deze button klikt, dient de app gesloten worden:

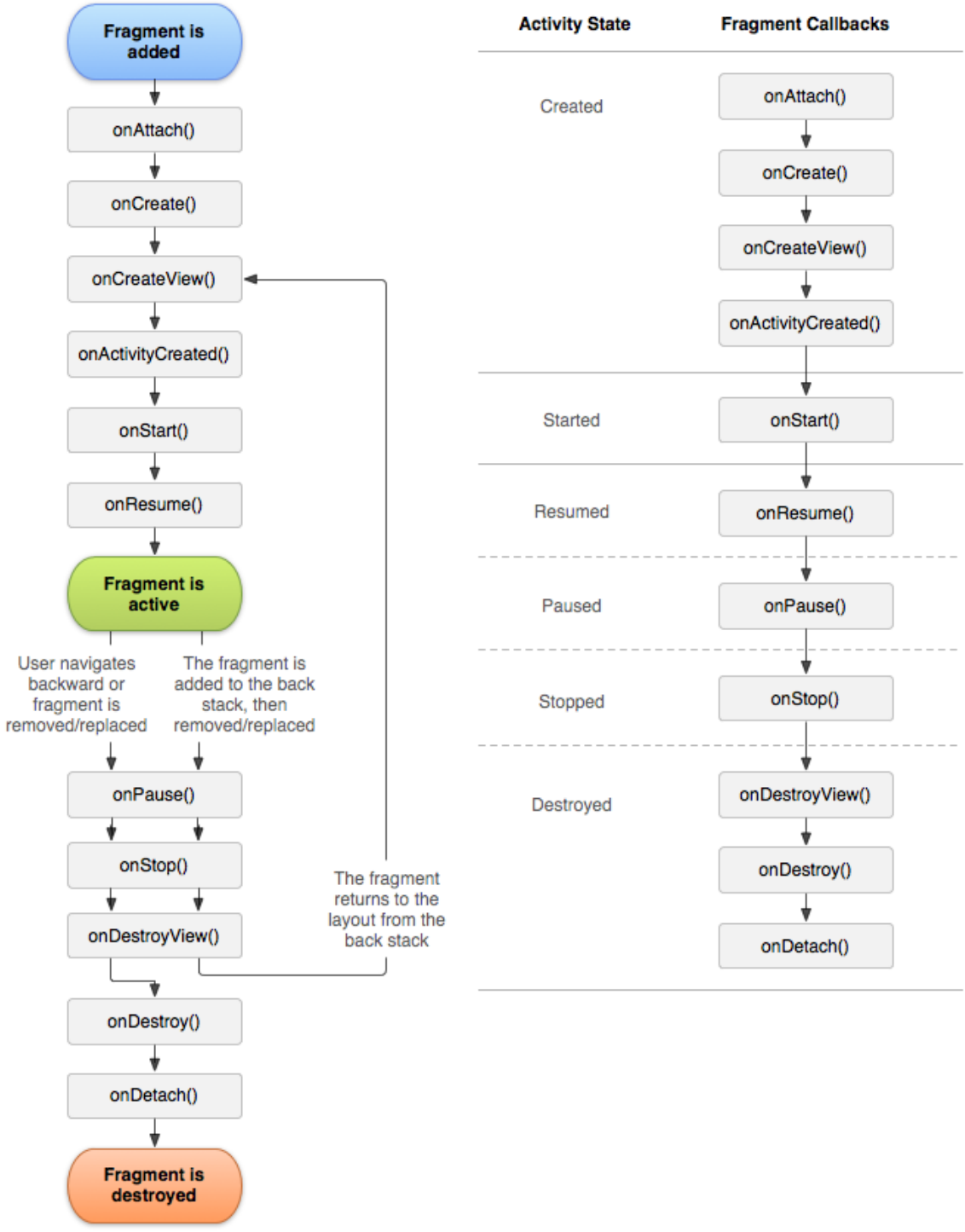
```
getActivity().finish();
```

Code

In de theorieles van week 2 werd de LifeCycle van een activity besproken. Het belangrijke schema hierbij is:



Wanneer jouw activity één (of meerdere) fragments bevat, staat de lifecycle van een fragment o.a. in relatie tot de activity. In deze oefening beperken we ons tot één fragment. Onderstaand schema geeft deze relatie fragment-activity weer.



Verdere (belangrijke) toelichting is na te lezen op:
http://developer.android.com/training/basics/activity-lifecycle/starting.html
http://developer.android.com/reference/android/app/Fragment.html#Lifecycle

In bovenstaand schema vind je

- De verschillende statussen ('states') van een activity/fragments
- De overgangen tussen de verschillende states. Deze worden aangeduid via pijlen. Elke overgang kan via een methode verder uitgewerkt worden door de methode in de Activity te overriden. We spreken over de 'lifecycle methods'

Override elke 'lifecycle method' in de activity en fragment. In de toekomst is het niet nodig om elke methode te overriden. In deze opgave wensen we na te gaan wanneeer welke methode actief wordt. Om dit te testen gaan we in elke methode gaan loggen. Dit doen we door van de Log klasse gebruik te maken:

http://developer.android.com/tools/debugging/debugging-log.html

Voorbeeld van gebruik in de onCreate-methode:

```
Log.d(getClass().getSimpleName(), "onCreate");
```

Via Logcat binnen Android Studio kan je alle logberichten bekijken. Merk op dat er ook een prioriteit heerst onder de verschillende getoonde logberichten. De geselecteerde prioriteit in de combobox is dus de minimum-prioriteit.

Indien alle 'lifecycle methods' van een log-methode voorzien zijn , simuleer je onderstaande situaties. Ga telkens na welke methodes reageren en let hierbij ook op de volgorde. **Schrijf op wat je vaststelt**!

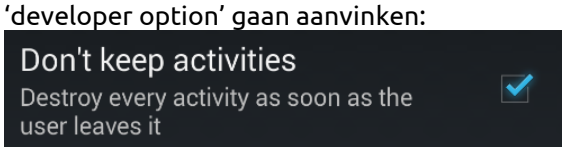
- **situatie 1**: opstarten van de App (via Android Studio)

- **situatie 2**: (App is actief) Drukken op de 'back'-button, druk daarna opnieuw op het launch-icoontje.
- **situatie 3**: (App is actief) Drukken op de 'home'-button. Druk daarna opnieuw op het launch-icoontje.

- **situatie 4**: (App is actief) Drukken op de 'finish'-button binnen in de activity. Druk daarna opnieuw op het launch-icoontje. Het afsluiten van een activity kan via de finish()-methode.

Merk het grote verschil op tussen de Home- en de Back-button!

Opgelet: na het klikken op de button 'Home' kan de activity (app) later alsnog volledig beëindigd worden. Oorzaak kan bijvoorbeeld 'Lack of memory' zijn… Om dit te simuleren kunnen we een 'developer option' gaan aanvinken:



- **situatie 3 bis**: (App is actief) Drukken op de 'home'-button. Druk daarna opnieuw op het launch-icoontje. Wat zijn de verschillen met situatie 2 en situatie 3?

Doel van volgende opgaves: we controleren welke informatie in welke views al dan niet zichtbaar blijven. Hoe kunnen we vermijden dat informatie verdwijnt (in de veronderstelling dat je net het omgekeerde wenst…)

Extra

Bestudeer ook volgende situatie

- **situatie 5**: roteren van de emulator/device.