

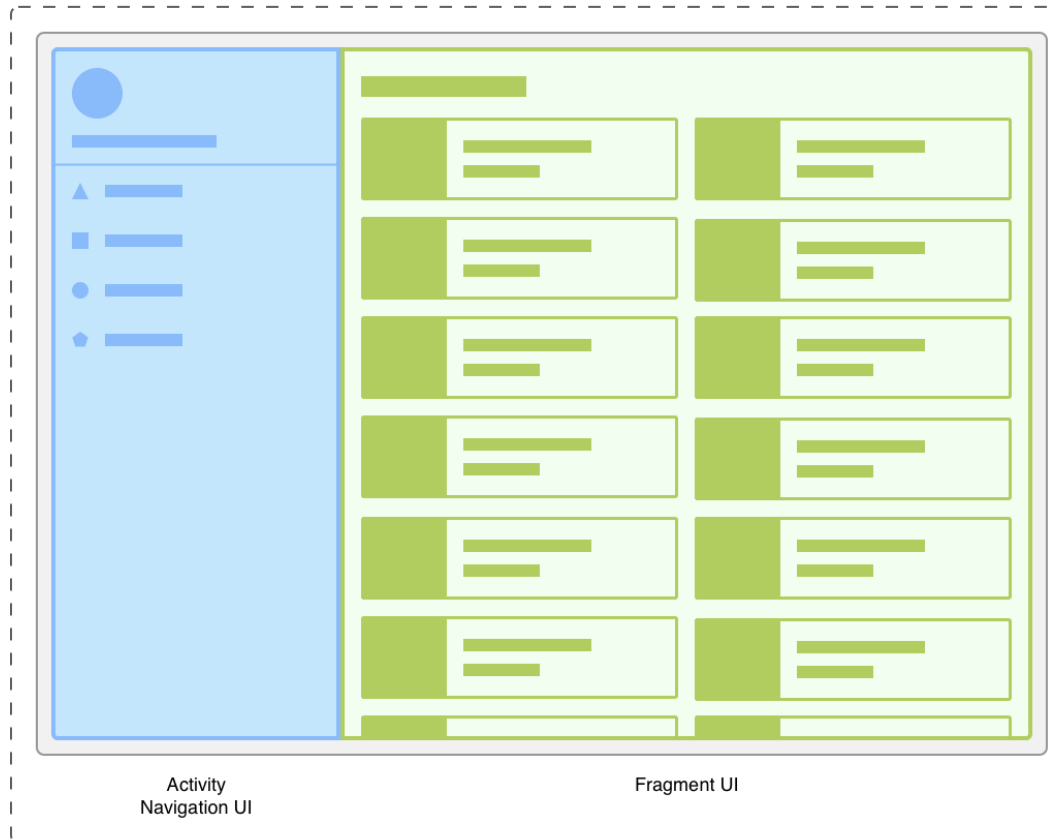
# Fragment

# Fragment

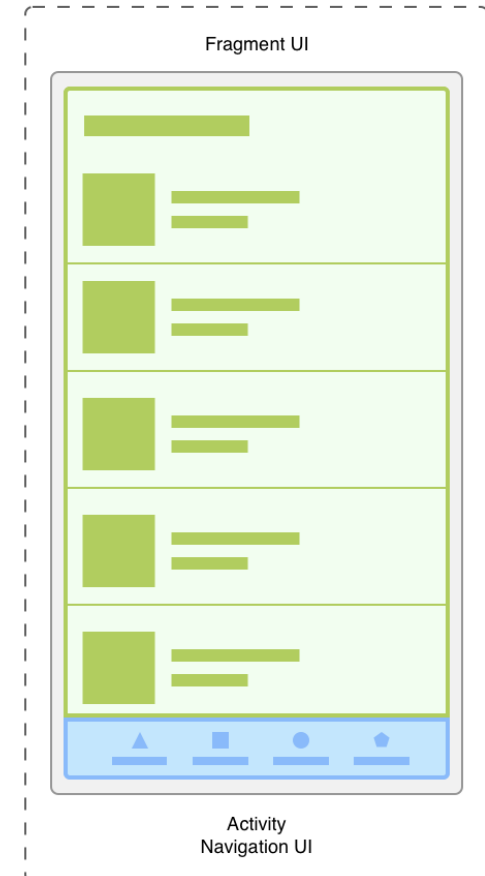
- A Fragment represents a reusable portion of app UI.
- Caratteristiche
  - sezione modulare che si può aggiungere e rimuovere
    - durante l'esecuzione di una activity
  - può gestire i propri eventi
  - vivono solo associati ad una activity
  - ha il proprio ciclo di vita
    - legato a quello dell'activity

# Riuso di componenti

Large Screen



Small Screen



<https://developer.android.com/guide/fragments>

# Fragment in xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment android:name="com.example.news.ArticleListFragment"
        android:id="@+id/list"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
    <fragment android:name="com.example.news.ArticleReaderFragment"
        android:id="@+id/viewer"
        android:layout_weight="2"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
</LinearLayout>
```

# Fragment Manager

getFragmentManager ()

- Funzionalità

- Gestire la navigazione attraverso l'utilizzo di **FragmentTransaction**.
- Gestire il *back stack*.
- Restituire il riferimento a un Fragment dato il suo id.
- Restituire il riferimento a un Fragment dato il suo tag.
- Restituire il riferimento a un Fragment precedentemente salvato in un Bundle.

# Fragment in java

```
public static class ExampleFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.example_fragment, container, false);
    }
}
```

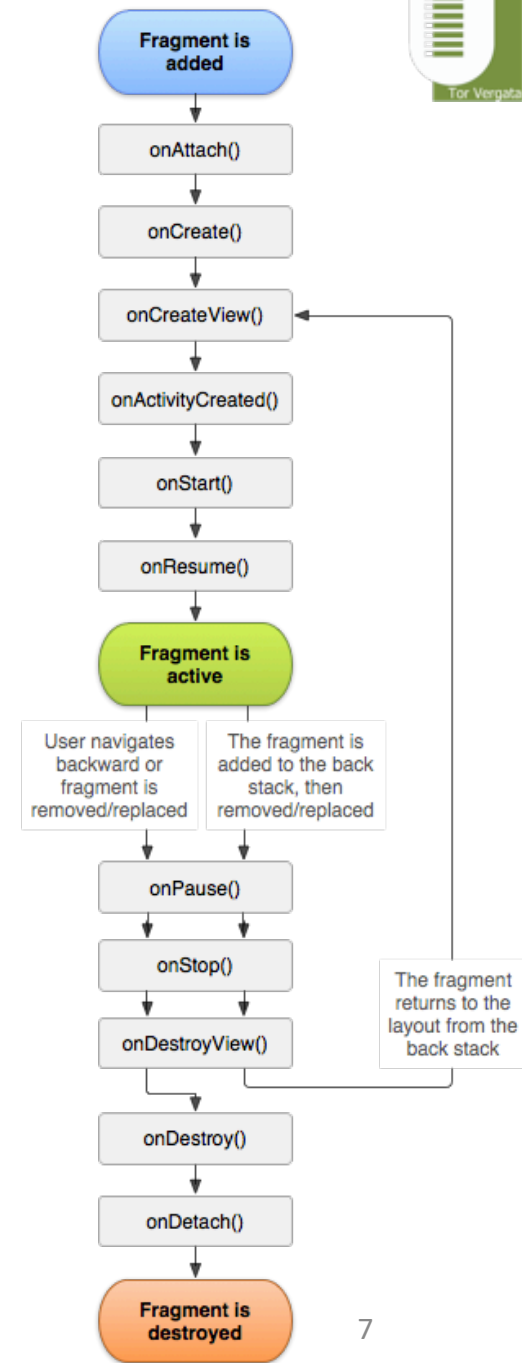
```
FragmentManager fragmentManager = getFragmentManager()
FragmentManager.beginTransaction().add(R.id.fragment_container, fragment);
FragmentManager.commit();
```

```
ExampleFragment fragment = new ExampleFragment();
fragmentTransaction.add(R.id.fragment_container, fragment);
fragmentTransaction.commit();
```

# Il ciclo di vita 1

## Esempio Carli FirstFragment

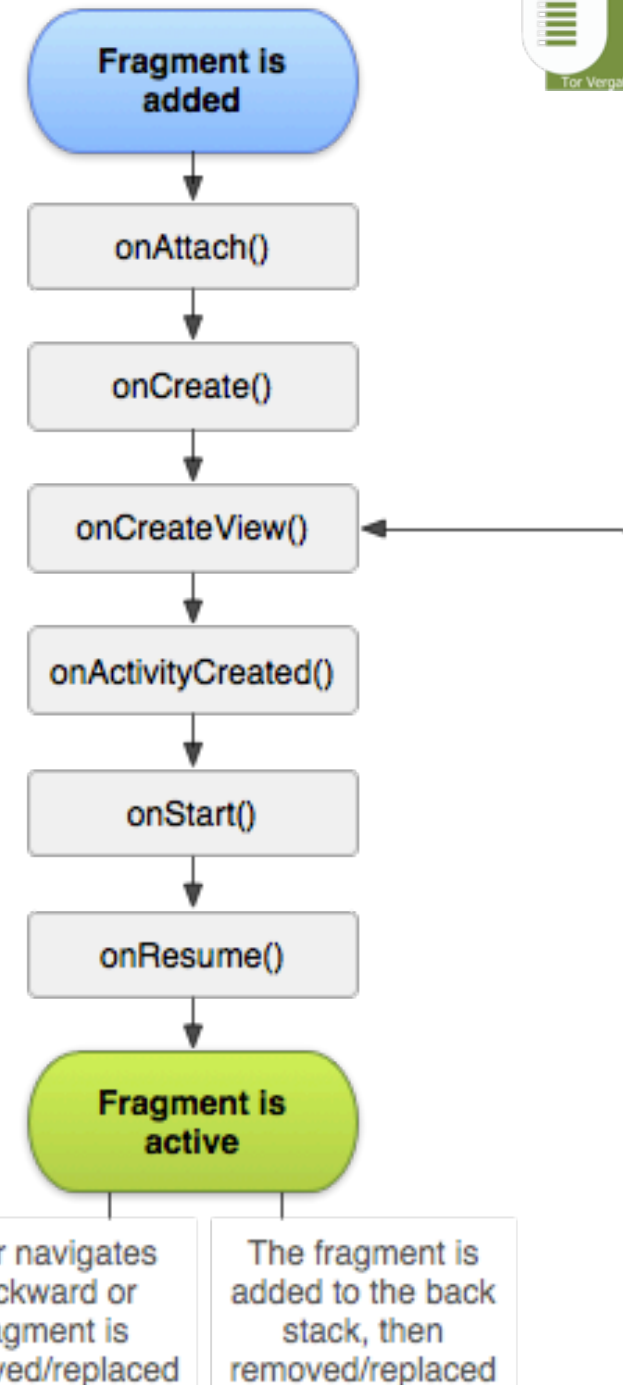
- **Creazione:**
  - Solo costruttore di default
  - metodo `setArgument()` per passare parametri
    - usa il Bundle
- **`onAttach()` quando viene associato all'activity**
  - `getActivity()` per prendere l'activity a cui sono collegato



# Il ciclo di vita 1

## Esempio Carli FirstFragment

- **Creazione:**
  - Solo costruttore di default
  - metodo `setArgument()` per passare parametri
    - usa il Bundle
- **`onAttach()` quando viene associato all'activity**
  - `getActivity()` per prendere l'activity a cui sono collegato





# Il ciclo di vita 2

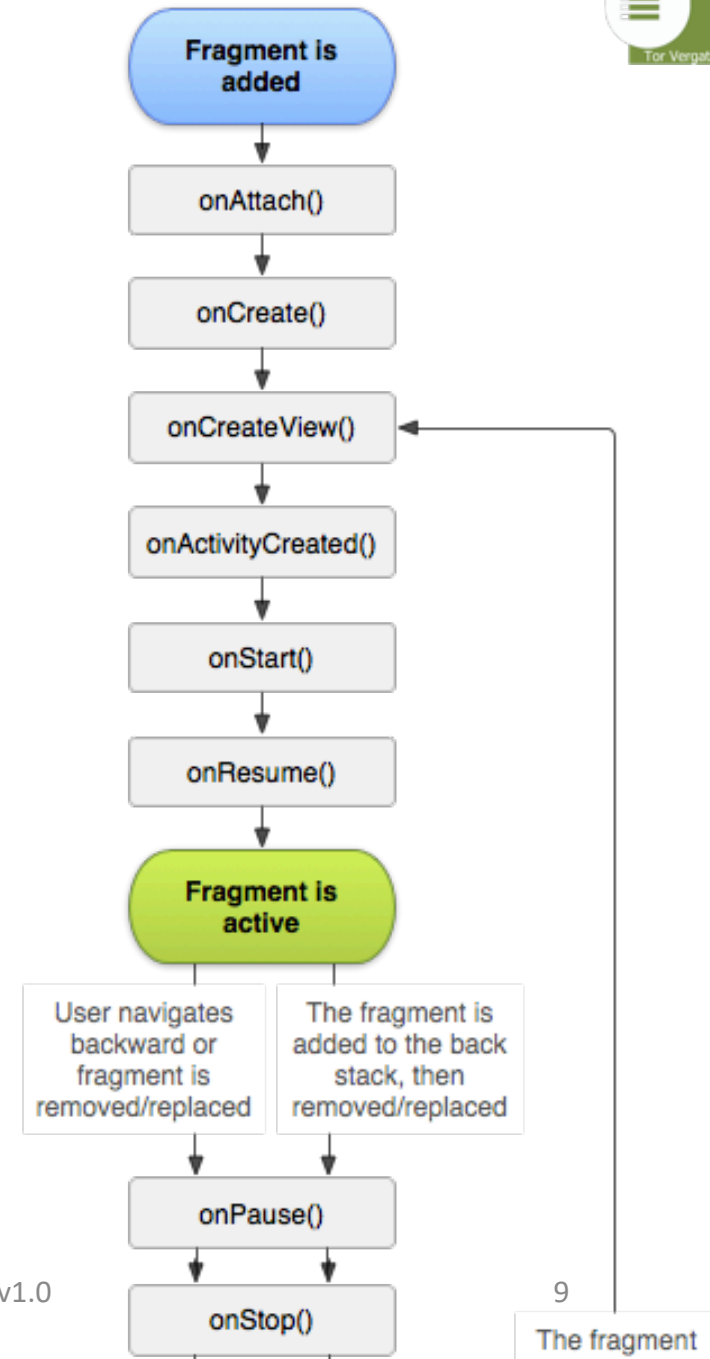
## • Grafica:

- onCreate inizializza ma non dovrebbe gestire elementi grafici
- onCreateView: restituisce la root view associata al fragment
  - Nota: potrebbe non contenere elementi grafici
  - Il LayoutInflater va usato per allocare gli elementi grafici

## • onActivityCreated

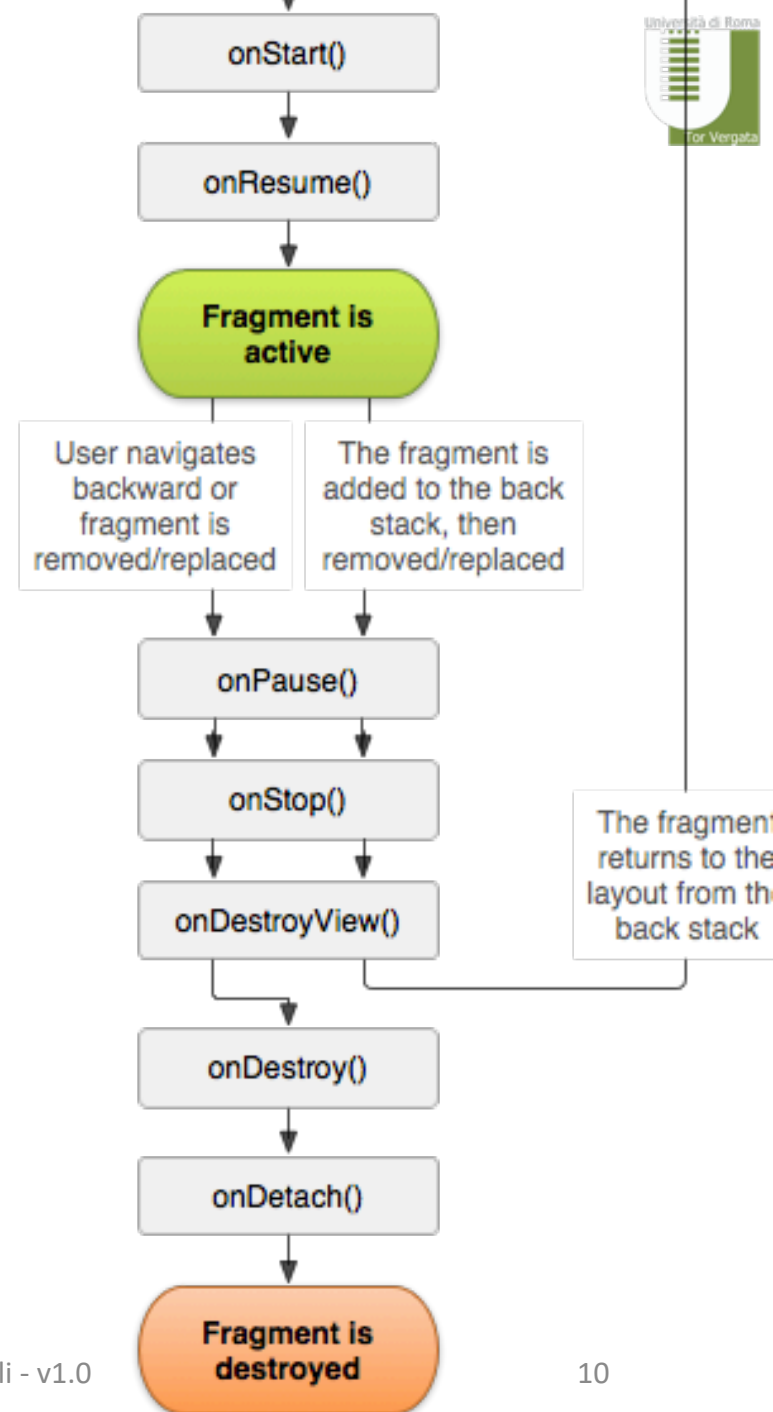
- l'activity è creata
- sono finite tutte le attività legate al metodo onCreate

## • Le altre chiamate sono legate a quelle dell'activity



# Il ciclo di vita 3

- Le altre chiamate sono legate a quelle dell'activity



# Backstack

- Il tasto back serve per navigare nella “storia” delle schermate
- La sequenza dei componenti da visualizzare e memorizzata nel Backstack
- Cambio di Activity
  - le schermate associate alle activity sono aggiunte automaticamente al backstack
- Cambio di Fragment
  - devo aggiungere manualmente la transizione al backstack
  - `transaction.addToBackStack( ... )`

# Esempio

- Tutorial First Fragment project