Лабораторная --- Фрагменты

Цели:

Познакомиться с классом Fragment. Создать простое приложение, которое использует Фрагменты для создания двухпанельных или однопанельных пользовательских интерфейсов, в зависимости от размера экрана конкретного устройства.

После завершения этой Лабораторной вы должны будете лучше понимать класс Фрагмент и его жизненный цикл, и как Фрагменты взаимодействуют с Activity которое их содержит.

Упражнение: Фрагменты

В этом упражнении вы создадите приложение, которое использует Фрагменты для отображения симулированных твитов[1] сервиса Twitter. Это приложение представит множественные фрагменты, организованные в различные макеты в зависимости от размера экрана устройства. Один из этих Фрагментов, именуемый FriendsFragment, будет отображать имена различных знаменитостей. Если пользователь выбирает имя из этого Фрагмента, он увидит различные симулированные твиты от этих людей, появившиеся во втором Фрагменте, названным FeedFragment. В этой лабораторной число и имена знаменитостей будет фиксированы (когда мы узнаем больше о классах Пользовательского Интерфейса, работой с Сетью и Сервисами, вы, возможно, захотите расширить это приложение до настоящей Twitter-читалки).

Скриншоты, приведенные ниже изображают пользовательский интерфейс приложения в работе на типовом устройстве с маленьким экраном (напр. смартфон). Этот макет будем называть однопанельным:

[1] Twitter API требует соединения с сетью и аутентификации для получения данных от Twitter. Поскольку мы еще не обсуждали работу с сетью, приложение будет симулировать ленту Twitter.

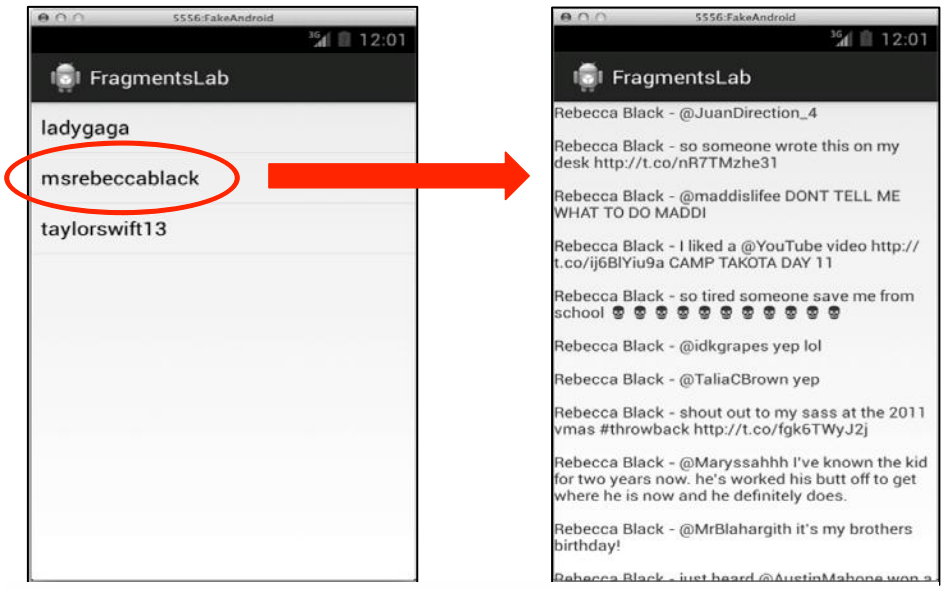

Рис. 1: Когда пользователь нажимает на имя знаменитости, он видит твиты этой знаменитости

Чтобы реализовать этот пользовательский интерфейс, вы будете реализовывать два Фрагмента; один из них FriendsFragment, а второй – FeedFragment. Фрагмент FriendsFragment, изображенный на левой части изображения, приведенного выше, является подклассом класса ListFragment. Если пользователь выбирает имя знаменитости из ListFragment, то Твиты этой персоны будут отображены в FeedFragment. Класс FeedFragment отображает отдельный TextView, обернутый в ScrollView, содержащий все твиты отдельной знаменитости. Если пользователь нажимает на кнопку «Назад» в то время как виден FeedFragment, приложение должно вернуться к предыдущему виду, в котором был виден FriendsFragment.
При запуске на широкоэкранном планшете, однако, приложение будет представлено другим пользовательским интерфейсом, как показано ниже. Этот макет именуется двухпанельным макетом.

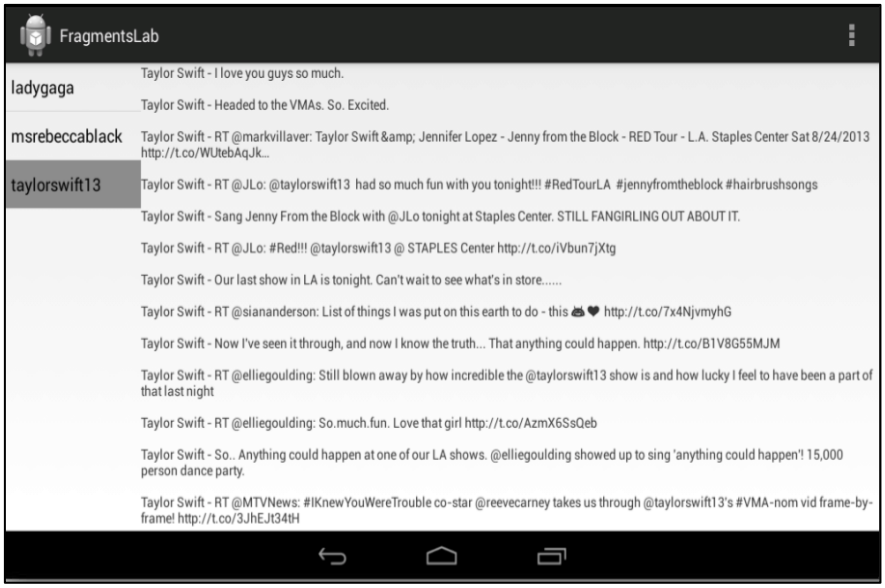

Рис. 2. Двухпанельный пользовательскпий интерфейс планшета.

В этом случае, приложение отображает оба Фрагмента в одно и то же время. Вы реализуете код, который управляет макетами приложения. **Обратите внимание**: Вы не создаете два различных приложения. Вы создаете одно приложение, которое запускается как на мобильном телефоне, так и на планшете. Для дополнительной информации, см. документацию (http://developer.android.com/guide/practices/tablets-and-handsets.html)
См. ролик FragmentsLabPhone.mp4 чтобы увидить приложение в работе на телефоне. См. ролик FragmentsLabTablet.mp4 чтобы увидеть работу приложения на планшете.

Заметки по реализации:

Это приложение достаточно схоже с приложениями, обсуждаемыми в лекции. Вам предоставлен скелет приложения, включая все необходмые макеты и файлы ресурсов. Не изменяйте идентификаторы ресурсов (ID) в этих файлах , т.к. это может нарушить работоспособность тестов.
Вам потребуется модифицировать две области исходного кода лабораторной. Обе эти области помечены комментарием, содержащим слово “TODO.” Обе части кода находятся в файле MainActivity.java. Чтобы это сделать вам потребуется посмотреть и понять ID – идентификаторы ресурсов, содержащиеся в макете main_activity.xml находящемся в папках res/layout и res/layout-large.

1. В MainActivity.java, найдите “TODO 1.” Добавьте исходный код, необходимый чтобы добавить FriendsFragment в двухпанельный макет.
2. В MainActivity.java, найдите “TODO 2." Добавьте исходный код чтобы заменить FeedFragment расположенный в однопанельном макете.

Тестирование и отправка:

Тесты для данной Лабораторной находятся в Lab3c_FragmentsLabTestPhone и Lab3c_FragmentsLabTestTablet. Тест-кейс Lab3c_FragmentsLabTestPhone должен запускаться на на телефоне. Тест-кейс Lab3c_FragmentsLabTestTablet должен запускаться на Планшете (мы использовали Nexus 7). Вы можете запустить все тест-кейсы с помощью контекстного меню по правой кнопке Run As>Android Junit Test, или по одному, используя контекстное меню по отдельному классу тест-кейса. Классы теста используют Robotium.

Чтобы отправить файлы на проверку:

1. Ваш проект должен быть очищен от сгенерированных исходников с помощью build -> clean. В Android Studio при приобрезовании проекта в Gradle-проект, использовать в командной строке команду gradlew clean.
2. Файлы вашего проекта должны быть сжаты в zip архив с именем ActivityLabSubmit.zip.

Во время реализации приложения на различных стадиях лабораторной, запускайте тест кейсы как можно чаще, чтобы увидеть, делаете ли вы успехи и как эффективно продвигаетесь. Как только все тесты будет пройдены, используйте инструкции по отправке отчета. Эта лабораторной имеет два тест кейса, один для телефонов, другой для планшетов. Оба теста будут использованы для отправленных вами файлов.

Предупреждения:

1. Эти тест-кейсы запускались на эмуляторе с помощью Galaxy Nexus AVD с API level 18 при 768MB RAM (для телефонного теста) и Nexus 7 с API level 18 при 768MB RAM (для планшетного теста). Чтобы ограничить проблемы конфигурации, вам следует протестировать ваше приложение на подобных AVD.

Дополнительно:

Это упражнение может быть слишком простым для людей с хорошей программистской базой. Если вы подпадаете под эту категорию, есть несколько предложений по улучшению приложения, что приведет к повышению сложности.

1. Добавьте необходимый код для управления состоянием приложения, чтобы оно сохраняло свое состояние (какой Френд и/или Фид был выбран) после реконфигурации (например, поворот экрана).
2. Используйте инструменты разработчика, чтобы мониторить жизненный цикл классов Activity и Fragment.

3. Пристальнее посмотрите на поддержку множественных экранов. Рекомендуем документацию (http://developer.android.com/guide/practices/screens_support.html) для различных main_activity.xml файлов, один для больших экранов и один для меньших устройств. Прочитайте «Supporting Multiple Screens». После прочтения вернитесь и взгляните на ваше приложение. Можете ли вы сделать различные макеты для устройств портретной и альбомной ориентации для телефонов и для больших экранов (двухпанельный макет)?