

Оптимизация Android приложений под планшеты, обратная совместимость с телефонами

Османов А. С.

- Введение
- Пример «превращения» классического приложения в приложение под планшет
- Совместимость с предыдущими версиями:
 - Без Compatibility API
 - С Compatibility API

- Что было до появления Honeycomb?
 - Не было планшетов?
- Были, и на самом деле не так уж и мало 😊
 - GalaxyTab
 - Archos tablets
 - iPed
 - aPad
 - etc...
- Зачем же тогда создавать версию ОС, специально заточенную под планшеты?

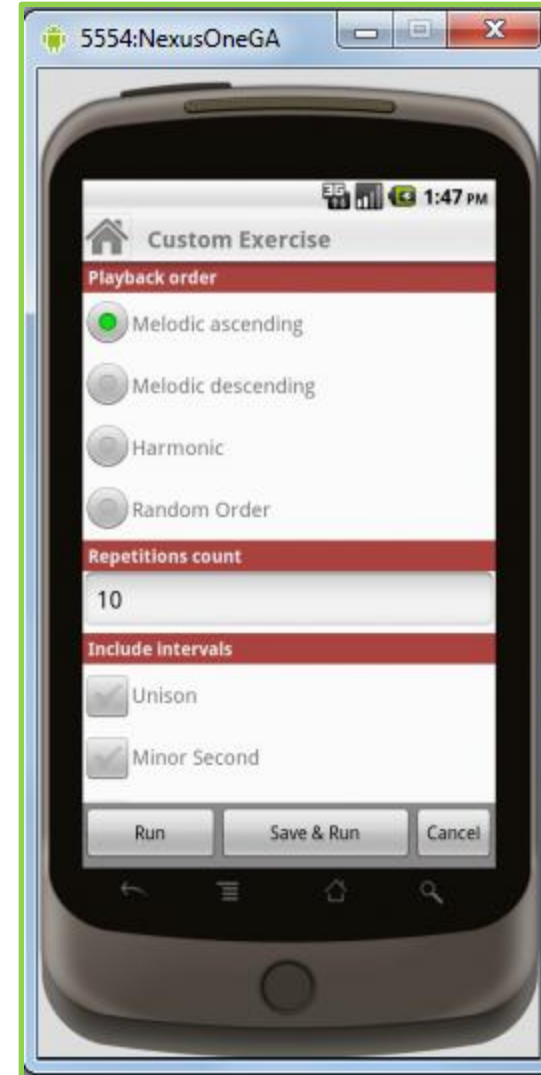
- Все они больше похожи на большие телефоны
 - Компоненты UI остаются теми же самыми, просто больше
- До Android 2.3 не было стандартного способа узнать, что приложение запущено на устройстве с большим экраном, а 100% из вышеперечисленных планшетов API 8 и ниже
- Невозможно фильтровать приложения в маркете

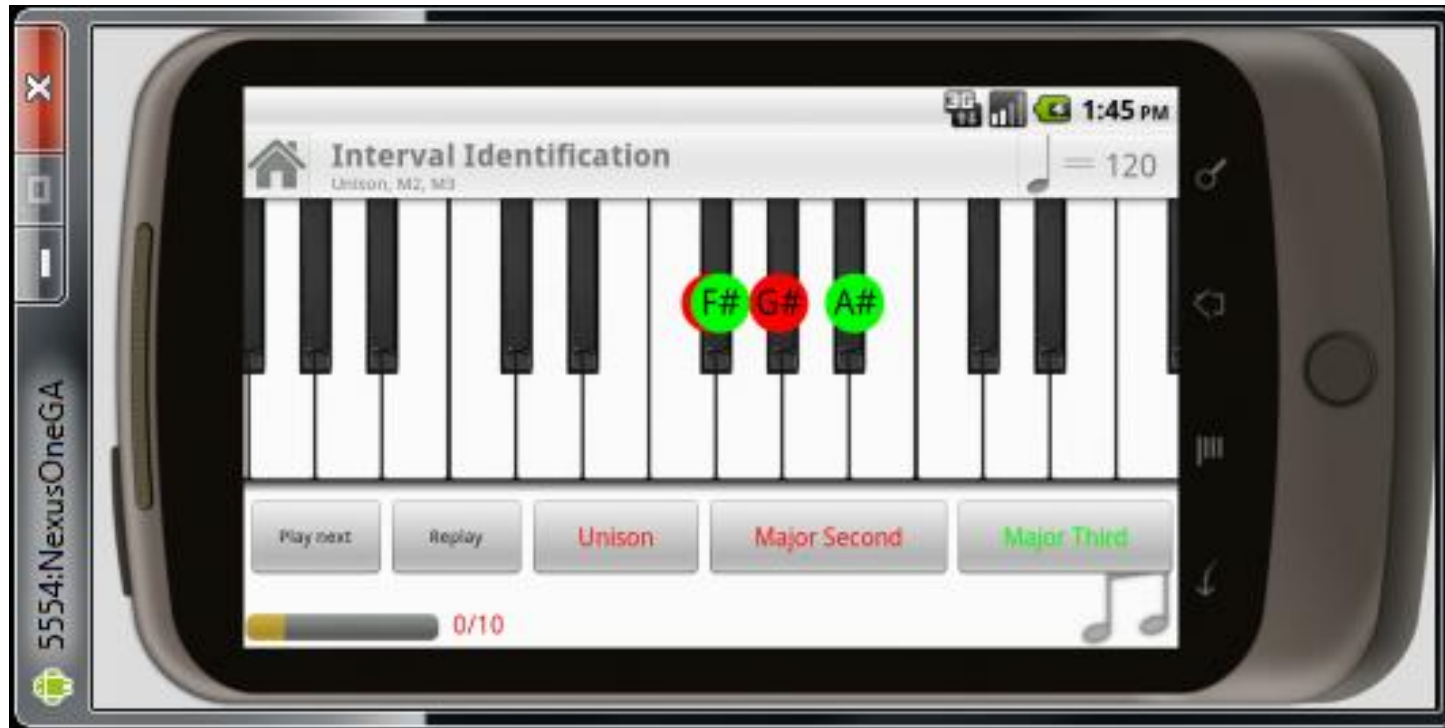
- Стандартные компоненты для оптимизации приложений под большие экраны:
 - **Fragments**
 - **Action Bar**
 - Animations
 - Enhanced widgets
- Возможность нативным способом определять, что вы на планшете, контролировать фильтры на маркете :
 - -v11
 - -xlarge

- Вы можете явно указать, что приложение только для v11, если это действительно так.

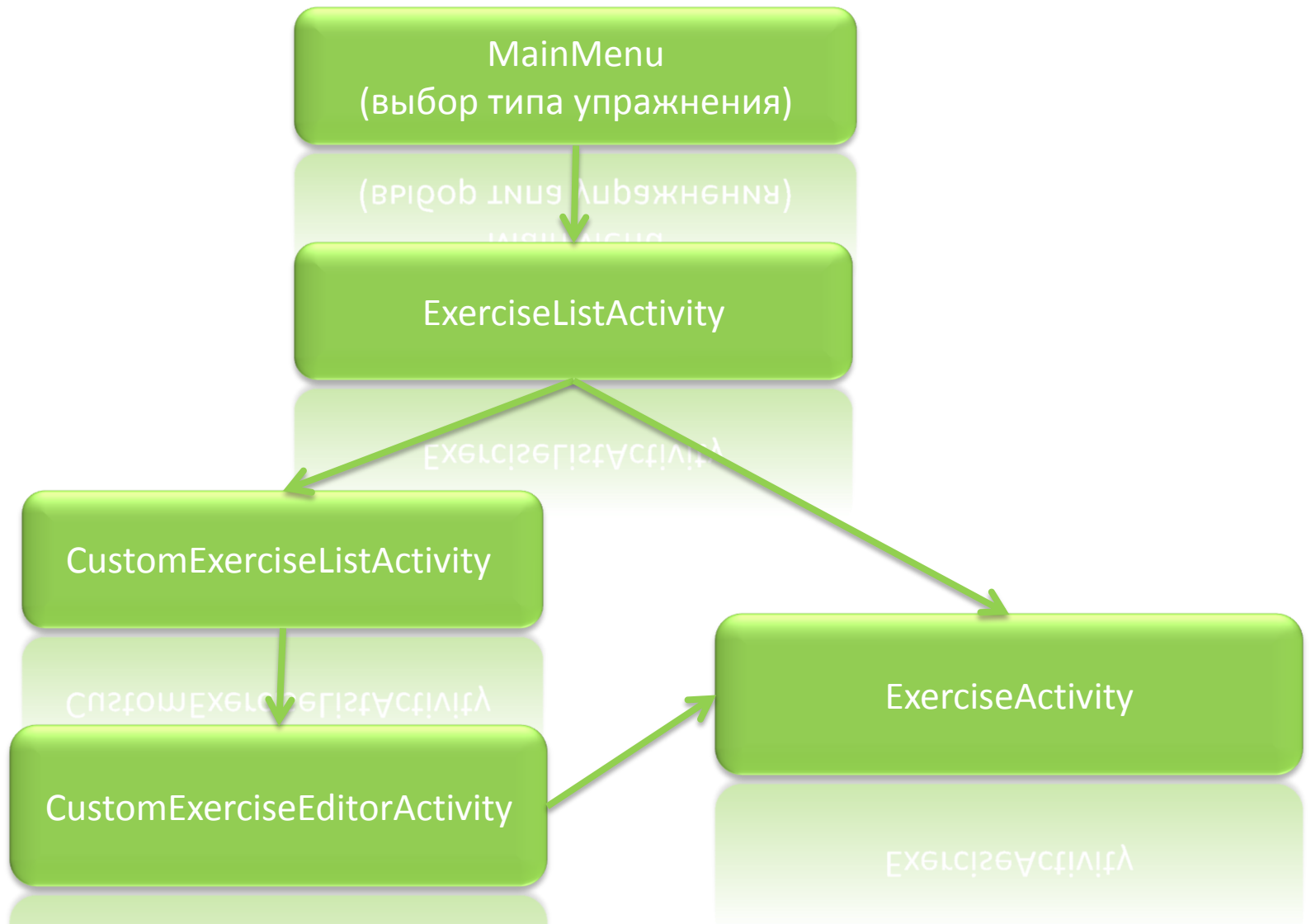
```
<manifest ... >  
  <uses-sdk android:minSdkVersion="11" />  
  <application ... >  
    ...  
  </application>  
</manifest>
```

- Простое приложение для тренировки музыкального слуха:
 - Несколько различных типов упражнений
 - Для каждого типа доступен ряд усложняющихся упражнений
 - Каждое упражнение выполняется с виртуальным инструментом





Структура Activities



Как оптимизировать под планшет?

- Совместим список упражнений с экраном упражнения в один activity:
 - **Fragments** – отдельный фрагмент список, отдельный – упражнения
 - **Action Bar** – новый способ навигации и замена Options Menu

- Фрагмент – это часть пользовательского интерфейса внутри Activity.
- Фрагмент имеет свой жизненный цикл внутри Activity
- Фрагмент может быть динамически добавлен, удален, заменен на другой
- Фрагмент может быть повторно использован в любом Activity Вашего приложения
- Действия фрагментами выполняются в виде «транзакций»
 - Могут сохраняться в back stack
 - Можно откатить кнопкой back

Action Bar

Fragment A

Fragment B

- Существующие Activities необходимо переписать в фрагменты
- Меняем базовый класс на Fragment
- Вносим корректировки в код:
 - Меняем процесс инициализации:
 - onCreate()
 - onCreateView()
 - onActivityCreated()
 - Меняем обращения к контексту:
 - getActivity()

- Для этого необходимо создать внешний Activity, в котором будут наши фрагменты
- Фрагменты можно помещать в XML файле, тогда они будут созданы при создании Activity
- Либо фрагменты можно помещать динамически
 - FragmentManager
 - setArguments()

- Взаимодействие компонентов

- Для посылки сообщений от фрагментов activities должны реализовать дополнительный интерфейс, который вы объявите

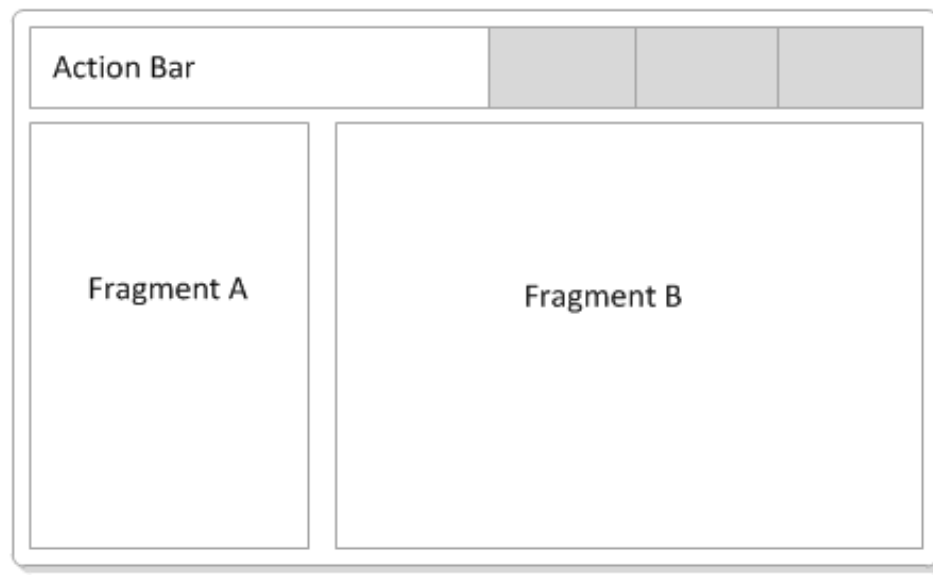
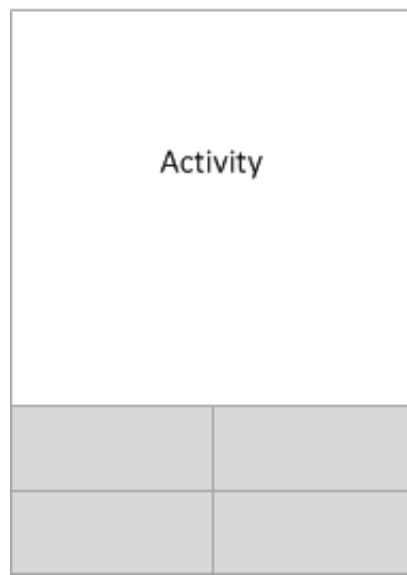
```
try {  
    mExerciseListener = (ExerciseCallback) getActivity();  
} catch (ClassCastException e) {  
    throw new ClassCastException("ExerciseCallback is not implemented");  
}
```

- Activities могут напрямую вызывать методы фрагмента или передавать Bundle при создании как setArguments

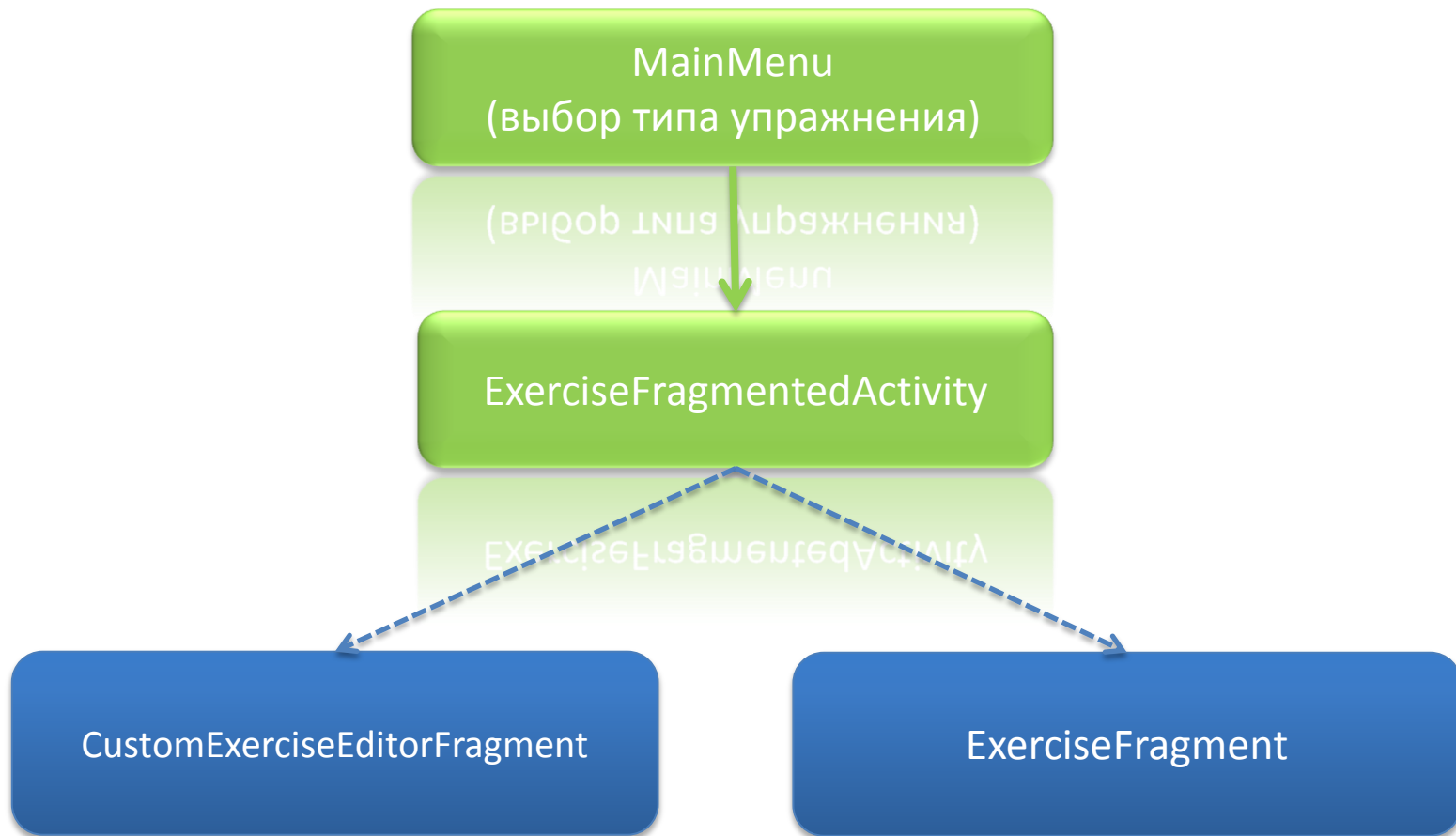
Добавление фрагментов

```
ExerciseFragment exerciseFragment = (ExerciseFragment) getFragmentManager()  
    .findFragmentByTag(EXERCISE_FRAGMENT_TAG);  
findViewById(R.id.exercise_container).setVisibility(View.VISIBLE);  
  
exerciseFragment = new ExerciseFragment();  
FragmentTransaction ft = getFragmentManager().beginTransaction();  
ft.replace(R.id.exercise_container, exerciseFragment,  
    EXERCISE_FRAGMENT_TAG);  
ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN);  
ft.addToBackStack(null);  
ft.commit();  
exerciseFragment.update(task, exerciseId);
```

- ActionBar – это замена title bar в pre v11
- Позволяет располагать в себе элементы управления
 - Вкладки
 - Опции из options menu
 - Кастомные компоненты
 - Заголовок, иконка



- Action Bar показывает те опции меню, которые предоставляет открытый фрагмент
- Для этого необходимо:
 - onCreateOptionsMenu(Menu, MenuInflater)
 - android:showAsAction
 - always
 - never
 - ifRoom
 - withText
 - setHasOptionsMenu(true);





Interval Identification

1/10

In this exercise you will need to listen to an interval played and identify it.



Custom Tasks

New custom task...



Pre-defined Tasks

1. Unison, M2, M3

ascending, single root

2. Unison, M2, M3

descending, single root

3. Unison, M2, M3

harmonic, single root

4. Unison, M2, M3

ascending, no common tone

5. Unison, M2, M3

descending, no common tone

6. Unison, M2, M3

harmonic, no common tone

7. Unison, M2, M3, P4, P5

ascending, single root

8. Unison, M2, M3, P4, P5

descending, single root

9. Unison, M2, M3, P4, P5

harmonic, single root

10. Unison, M2, M3, P4, P5

ascending, no common tone

Unison Major Second Major Third



Interval Identification

In this exercise you will need to listen to an interval played and identify it.

Custom Tasks
New custom task...

Pre-defined Tasks

1. **Unison, M2, M3**
ascending, single root

2. **Unison, M2, M3**
descending, single root

3. **Unison, M2, M3**
harmonic, single root

4. **Unison, M2, M3**
ascending, no common tone

5. **Unison, M2, M3**
descending, no common tone

6. **Unison, M2, M3**
harmonic, no common tone

7. **Unison, M2, M3, P4, P5**
ascending, single root

8. **Unison, M2, M3, P4, P5**
descending, single root

9. **Unison, M2, M3, P4, P5**
harmonic, single root

10. **Unison, M2, M3, P4, P5**
ascending, no common tone

Playback order

- Melodic ascending
- Melodic descending
- Harmonic
- Random Order

Repetitions count

10

Include intervals

- Unison
- Minor Second
- Major Second
- Minor Third
- Major Third
- Perfect Fourth
- Diminished 5th

Run

Save & Run

Cancel

Поддержка маленьких экранов

- Планшеты ориентированы на использование в landscape положении, но никто не мешает пользователям повернуть их в портретный режим
- Возможно, приложение будет запускаться на v11+ устройствах с маленьким экраном
- В этих случаях, скорее всего, фрагменты не подойдут и придется откатиться к «стандартной» модели Activity
- Что делать?

- Используется дополнительная Activity, hostящая один фрагмент:

```
<?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" android:id="@+id/ex_activity">
```

```
  <fragment android:name="ru.exaybachay.pearhc.ExerciseFragment"
```

```
    android:id="@+id/list_fragment" android:layout_weight="2"
```

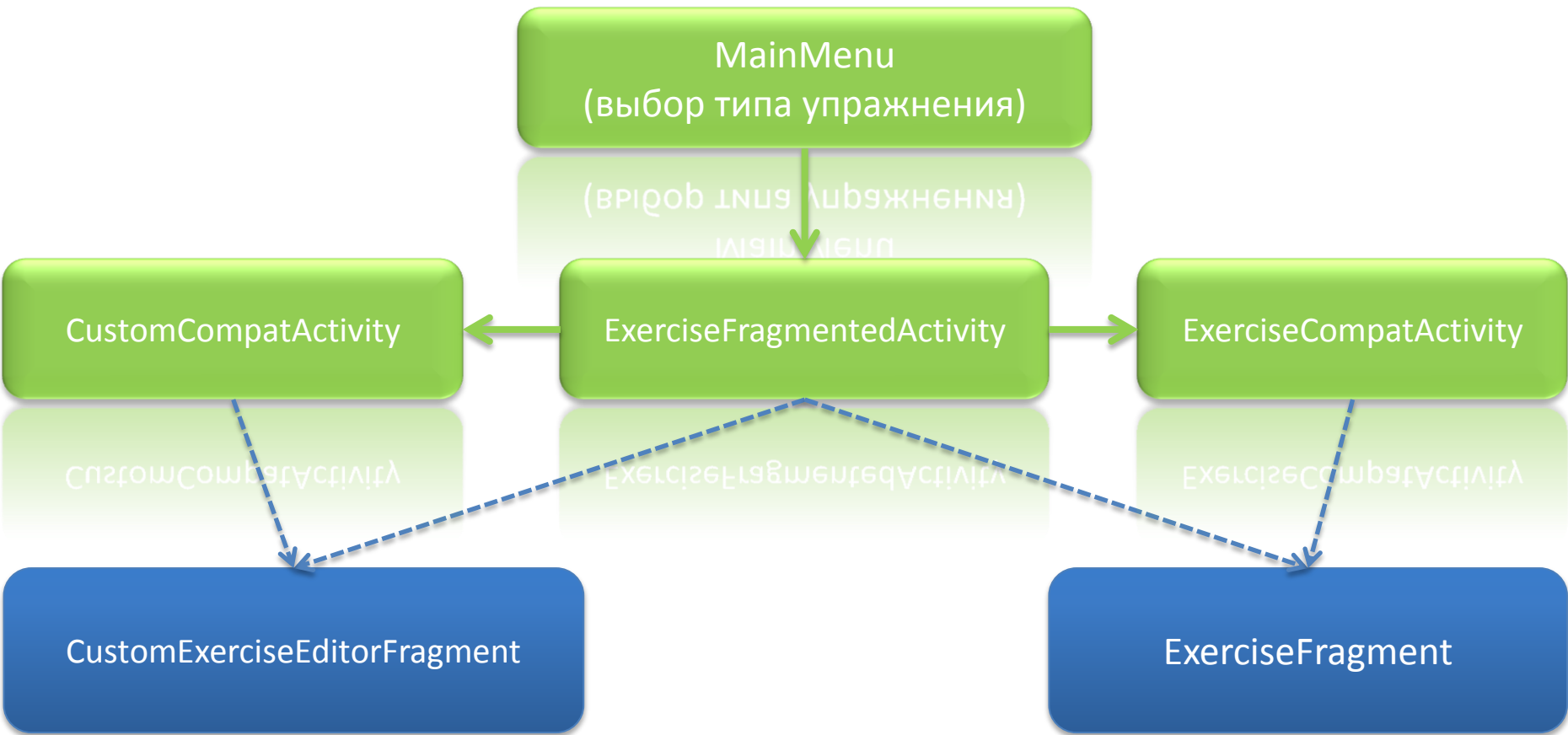
```
    android:layout_width="match_parent"
```

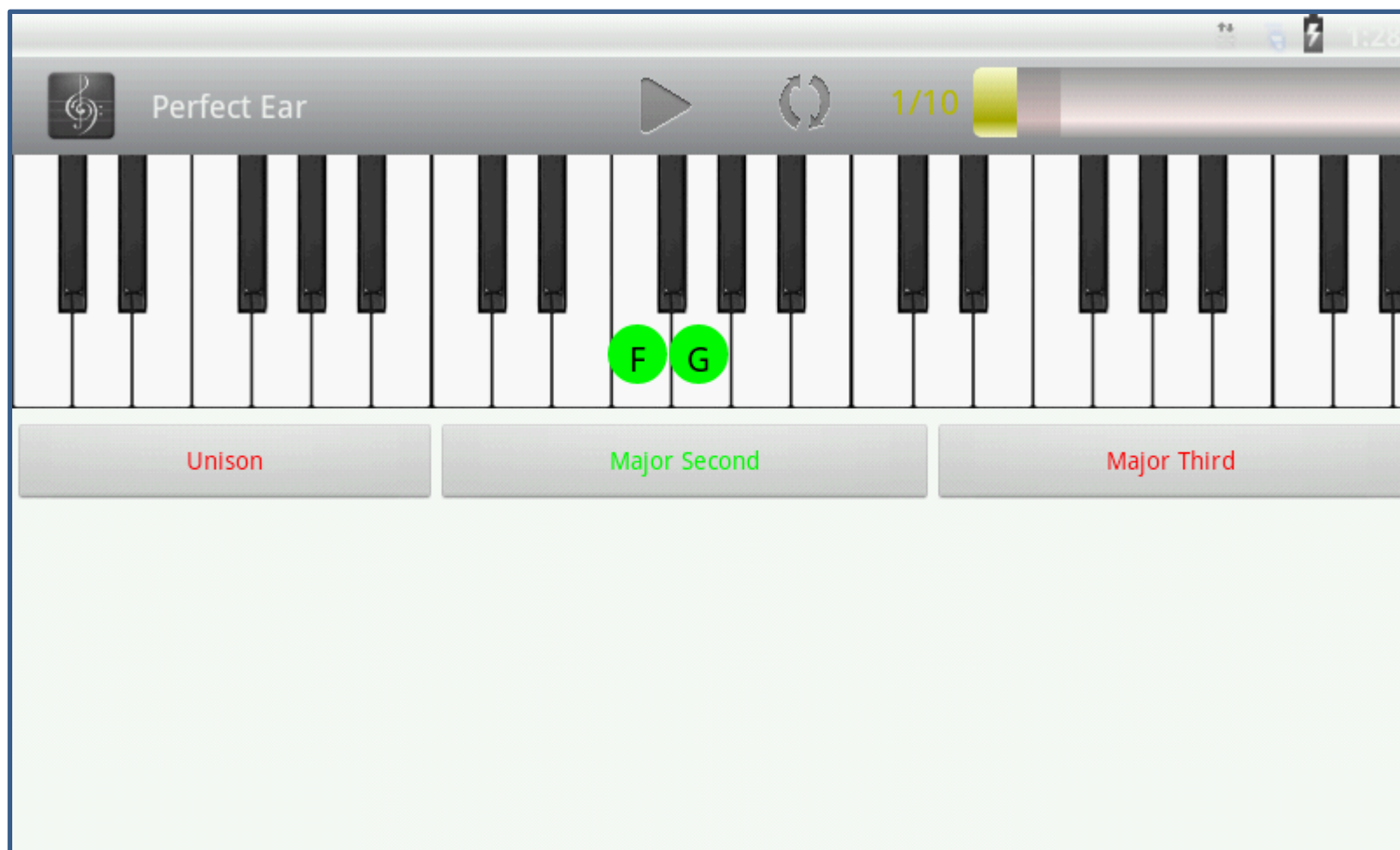
```
    android:layout_height="match_parent" />
```

```
</LinearLayout>
```

- Создайте альтернативный layout файл для -xlarge экранов либо -portrait
- Для не xlarge layout оставьте только фрагмент
- В коде это поможет вам определить, на каком вы устройстве работаете:
 - `boolean mDualScreen =
 findViewById(R.id.exercise_container) != null;`
 - `getResources().getConfiguration().screenLayout;`
- В зависимости от этого либо создавайте фрагмент, либо открывайте Activity, содержащий этот же фрагмент

Структура Activities + Fragments





- Стоит задача запускать единое приложение на более ранних версиях Android без Fragments API (all in one)
 - Без compatibility библиотеки
 - С compatibility библиотекой

- В ранних версиях, отсутствуют все API, используемые при работе с фрагментами
 - Нельзя повторно использовать fragments внутри activity
 - Отсутствуют классы, методы
- Многое достаточно легко обходится в ресурсах, но в коде легко не обойти

- Использовать отдельные ресурсы для v11:
 - layout-v11
 - menu-v11
- Вести параллельные ветви для fragments и non-fragments:
 - FragmentAwareActivity
 - FragmentUnawareActivity
 - Выбирать между ними в runtime

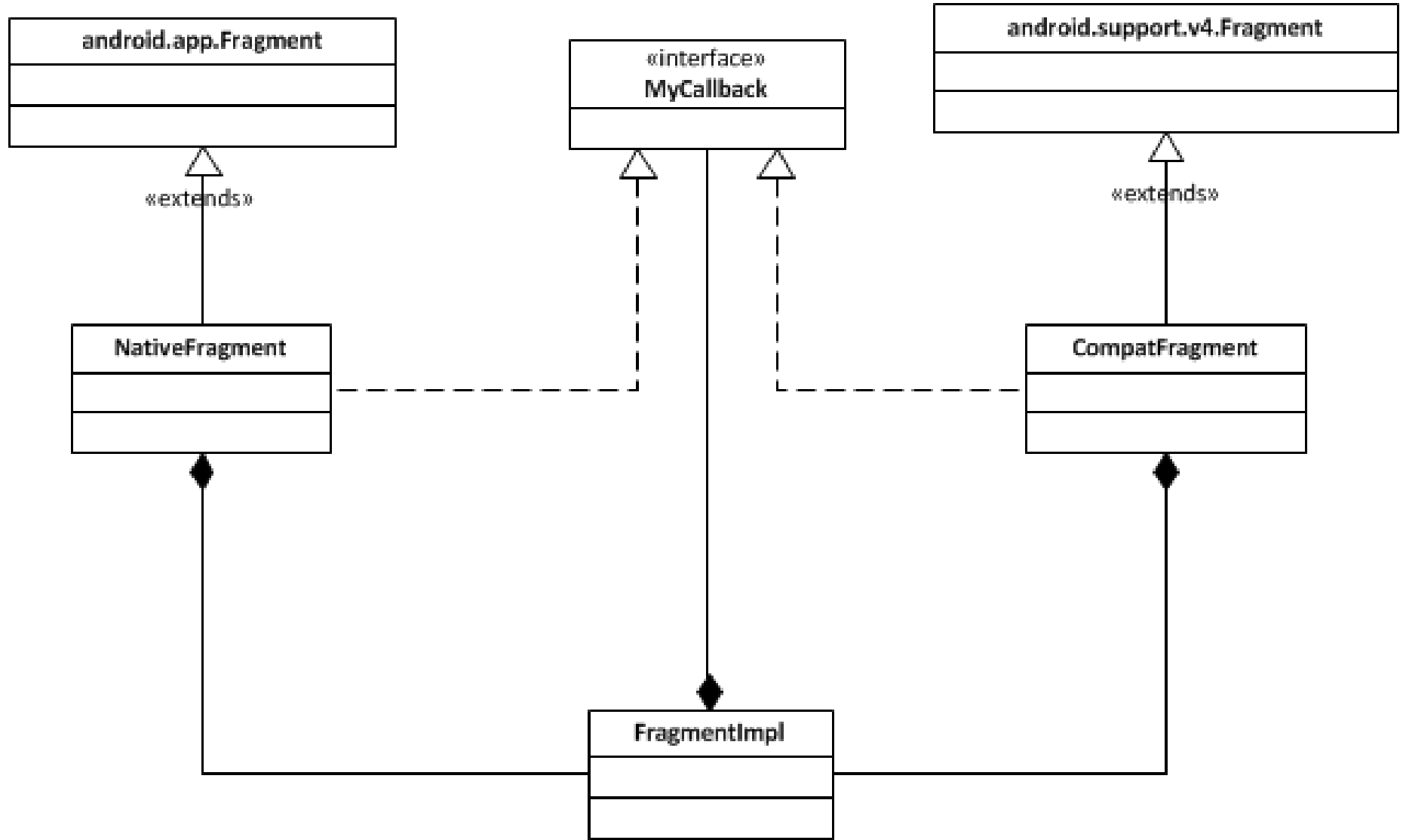
- Статическая библиотека, созданная Google для поддержки некоторых API Honeycomb в версиях вплоть до v4
- Призвана устранить необходимость ведения двух ветвей activities для honeycomb и pre honeycomb
- Установить можно через SDK and AVD Manager
 - Android compatibility package, revision 1
 - android-support-v4.jar

- При попытке написания all in one приложения возникают затруднения
 - Разные пакеты, в которых находятся классы нативных и compatibility не дают просто использовать те же классы
 - Необходимость наследовать FragmentActivity
 - Измененные имена методов
 - `getSupportFragmentManager()`
 - Классы являются отдельной реализацией и НЕ пытаются использовать fragments API, когда работают под v11

- Использовать только compatibility библиотеку
- Для простых случаев работает, но необходимо помнить, что является отдельной реализацией и поддерживает не все API Honeycomb
- Позволяет полностью избежать необходимости дублирования кода, таким образом задача совместимости аналогична задаче оптимизации под маленькие экраны

- Заменить базовые классы с `android.app.Fragment` на `android.support.v4.Fragment`
- Заменить базовые классы с `android.app.Activity` на `android.support.v4.FragmentActivity`
- Заменить имена методов из `v11` на соответствующие им методы из `compat lib`
 - Префикс `Support`
- Создать альтернативные XML ресурсы для меню БЕЗ `android:showAsAction`, либо убрать атрибуты из всех ресурсов и добавлять `showAsAction` в коде через `MenuCompat`
- Убрать или обернуть обращения к `ActionBar`

- Все же придется дублировать код, но дублируемые участки можно и нужно свести к минимуму
- Выносить логику в другие классы (внутренний), оставляя только различающиеся вызовы в дублирующихся (внешних) классах
- Внешние классы напрямую вызывают методы внутренних
- Для вызова методов внешних классов использовать интерфейсы
 - Каждый внешний класс реализует интерфейс
 - Внутренний класс держит ссылку на интерфейс
- Обязательно чистить ссылки в `onDestroy()`



- Объединяйте Ваши последовательные Activity в единые с использованием фрагментов
- Выносите навигацию каждого фрагмента в ActionBar
- Учитывайте, что в портретном режиме фрагменты могут не подойти, либо экран может быть недостаточно большим
- Используйте альтернативные XML ресурсы для разметки, меню (-v11, -xlarge)
- При проектировании главного меню отдавайте предпочтение паттерну Dashboard


- Если хотите переносить на более ранние платформы проектируйте заранее с расчетом на минимальную необходимость дублирования кода



 Interval Identification

 Interval Comparison

 Scale Identification

 Chord Identification

 Chord Inversion

 Interval Singing



Settings

Exit

Спасибо за внимание

Q&A