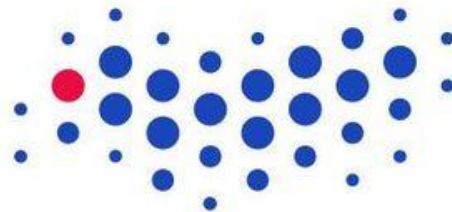


# Практикум на ЭВМ.

Программирование под Android

Занятие 7



ADB

# Android Debug Bridge

- `<sdk>/platform-tools/`
- `devices`
- `install/uninstall`
- `pull/push`
- `shell`
- `logcat`

# Hierarchy Viewer

Save as PNG

Capture Layers

Load View Hierarchy

Evaluate Contrast

Display View

Invalidate Layout

Request Layout

Dump DisplayList

Dump Theme

Profile Node

FrameLayout

@u062dff

id/full\_screen\_container

0

RelativePanelLayout

@8df77cc

id/messages\_list\_layout

0

FrameLayout

@51061e3

12

EmojiPanelView

@e5f07e0

0

ViewPager

@3a73c0d

id/pager

1

253 views

Measure: 1,283 ms

Layout: 9,290 ms

Draw: 36,896 ms

Property

Value

Layout

Measurement

measuredHeightAndState2112

measuredWidthAndState1440

mMeasuredHeight2112

mMeasuredWidth1440

mMinHeight0

mMinWidth0

Miscellaneous

Padding

Show Extras

Load All Views

Filter by class or id:

20%

200%

# Overdraw



Для разработчиков



ВКЛ



Всегда использовать GPU для двухмерного рисования

Показывать обнов. экрана

### Отладка наложения

- ☒ ВЫКЛ
- ☐ Показывать области наложения
- ☐ Выделять области определенного цвета

ОТМЕНА

Включить 4x MSAA

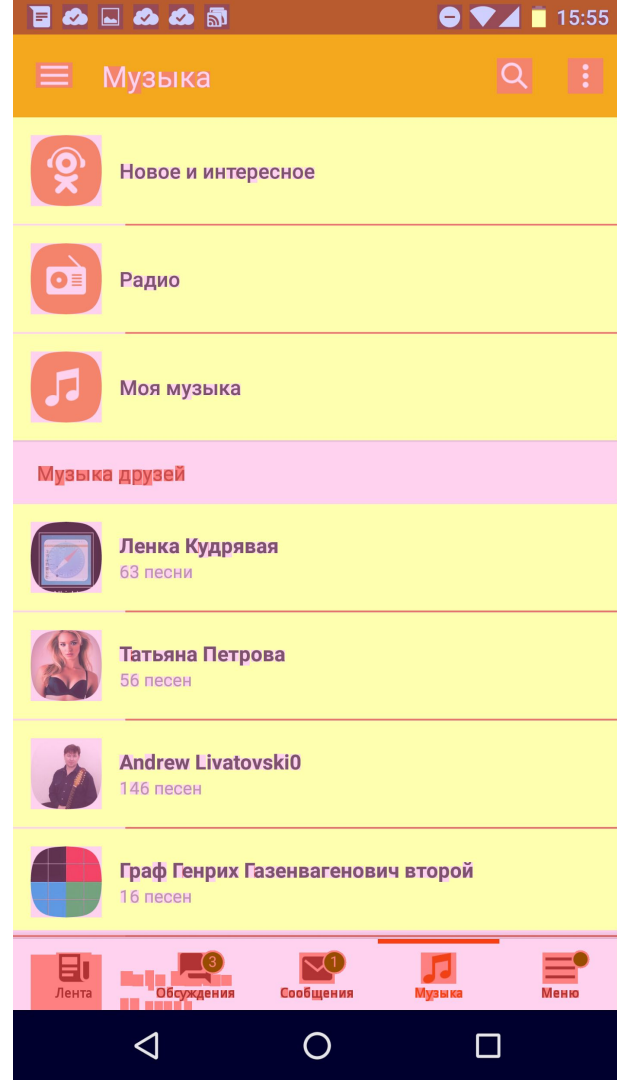
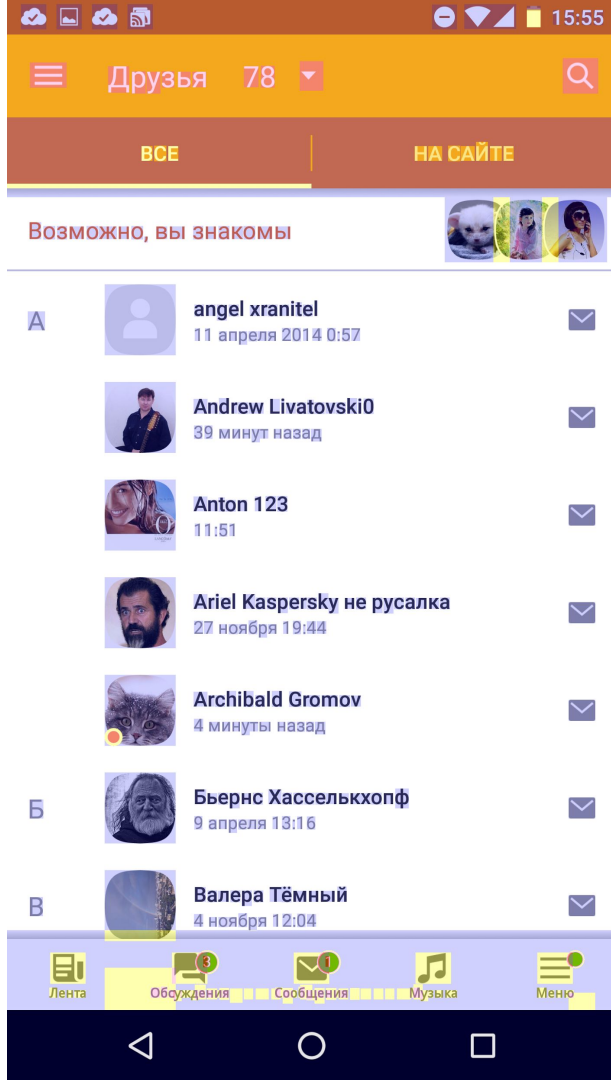
Включить 4x MSAA в приложениях OpenGL ES 2.0



Откл. аппарат. наложения

Всегда использовать GPU при компоновке экрана

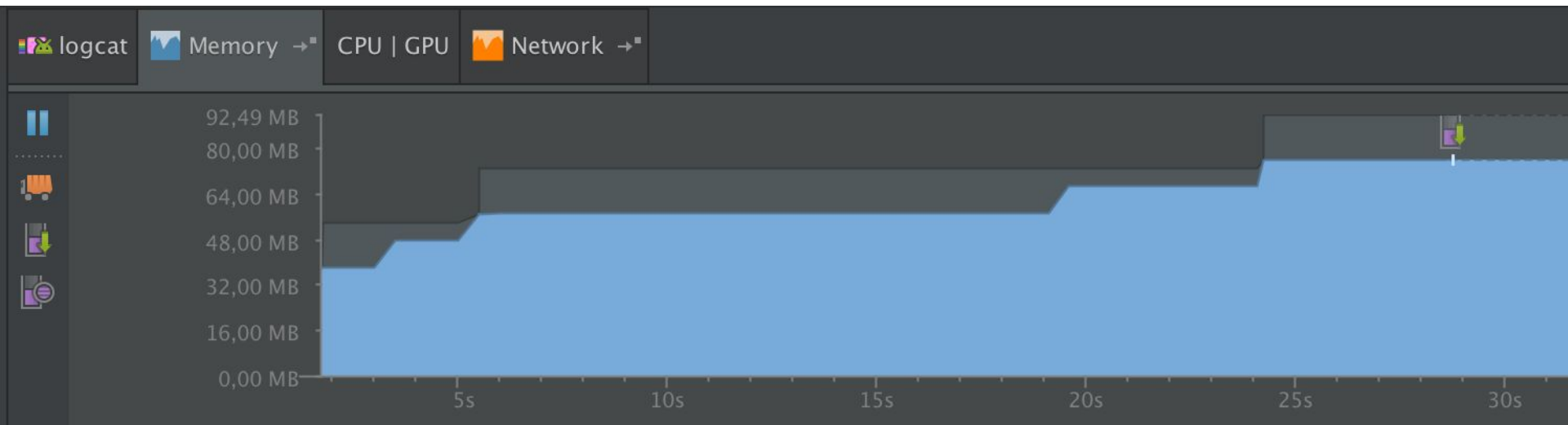






# Memory Analyzer

```
public final class LeakActivity extends Activity {  
    private byte [] array = new byte [10_000_000];  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        new AsyncTask<Void, Void, Void>() {  
            @Override  
            protected Void doInBackground(Void... params) {  
                try {  
                    Thread.sleep(30_000);  
                } catch (InterruptedException e) {  
                }  
                return null;  
            }  
        }.execute((Void)null);  
    }  
}
```



Class Name	Total Count	Heap Count	SizeofShallow...	Retain...	Instance
byte[]	2126	29	0 6022706022701		f 0 = {ru.ifmo.android_2015.toolsdemo.ui.LeakA
LeakActivity (ru.ifmo.android_2015.toolsdem	6	6	208 1248 6000923		f 1 = {ru.ifmo.android_2015.toolsdemo.ui.LeakA
LeakActivity\$1 (ru.ifmo.android_2015.toolsde	6	6	32 192 5014479		f 2 = {ru.ifmo.android_2015.toolsdemo.ui.LeakA
Object[] (java.lang)	2583	492	0 23660 4020540		f 3 = {ru.ifmo.android_2015.toolsdemo.ui.LeakA
AsyncTask\$SerialExecutor (android.os)	1	1	16 16 4011672		f 4 = {ru.ifmo.android_2015.toolsdemo.ui.LeakA
ArrayDeque (java.util)	1	1	20 20 4011671		f 5 = {ru.ifmo.android_2015.toolsdemo.ui.LeakA
AsyncTask\$SerialExecutor\$1 (android.os)	6	6	16 96 4011664		
AsyncTask\$3 (android.os)	6	6	32 192 4011658		
FinalizerReference (java.lang.ref)	1594	318	36 11448 1404604		
DexCache (java.lang)	14	1	40 40 167240		
int[]	5858	313	0 166808 166808		
PhoneWindow (com.android.internal.policy)	7	7	227 2250 141102		

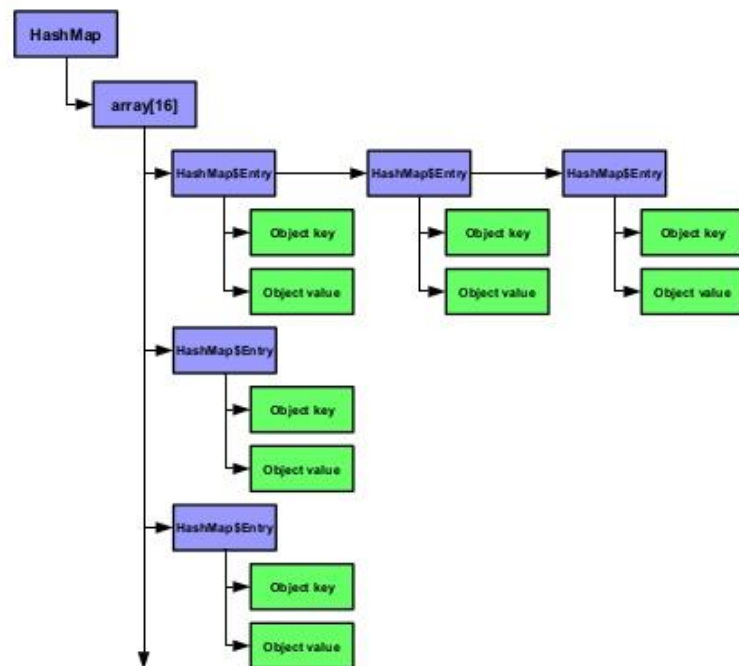
Reference Tree	Depth
ru.ifmo.android_2015.toolsdemo.ui.LeakActivity@316597632 (0x12dee580)	7
this\$0 in ru.ifmo.android_2015.toolsdemo.ui.LeakActivity\$1@315303456 (0x12cb2620)	6
this\$0 in android.os.AsyncTask\$3@315303488 (0x12cb2640)	5
val\$r in android.os.AsyncTask\$SerialExecutor\$1@314583104 (0x12c02840)	4
mFuture in ru.ifmo.android_2015.toolsdemo.ui.LeakActivity\$1@315303456 (0x12cb2620)	6
this\$0 in android.os.AsyncTask\$2@314583072 (0x12c02820)	6
mOuterContext in android.app.ContextImpl@315253248 (0x12ca6200)	8
mContext mPrivateFactory in com.android.internal.policy.PhoneLayoutInflater@314883088 (0x12c4bc10)	8

# ArrayMap

# HashMap



## HashMap Structure



# ArrayMap

```
public final class ArrayMap<K, V> implements Map<K, V> {  
    ...  
    int[] mHashes;  
    Object[] mArray; // keys & values  
    int mSize;  
    ...  
}
```

# SparseArray

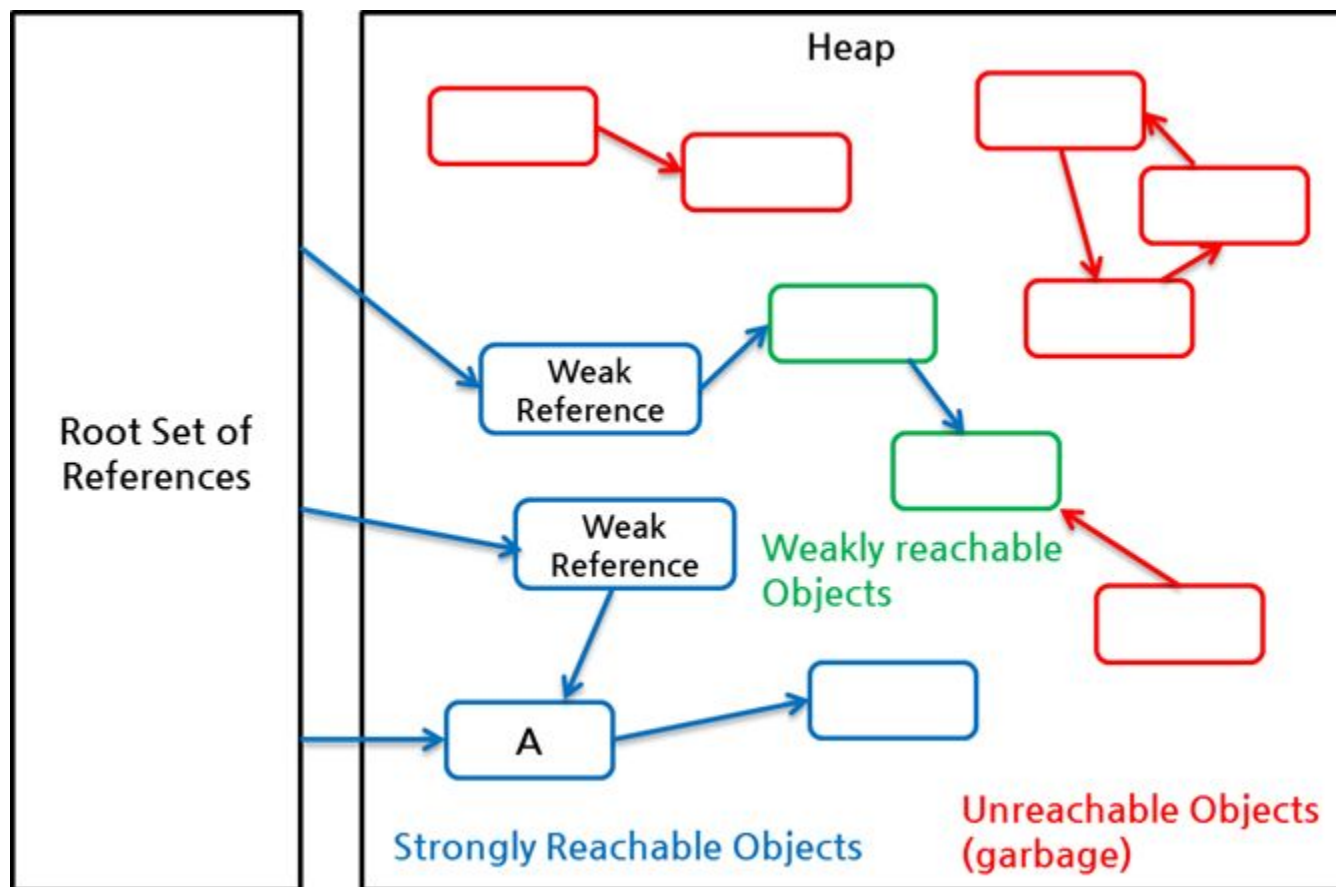
```
public class SparseArray<E> implements Cloneable {  
    ...  
    private int[] mKeys;  
    private Object[] mValues;  
    private int mSize;  
    ...  
}
```



# Versus

```
for (int i = 0; i < 1_000_000; i++) {  
    hashMap.put(i, new Object()); // 52 387 120  
    arrayMap.put(i, new Object()); // 32 062 208  
    sparseArray.put(i, new Object()); // 16 388 605  
}
```

# WeakReference



```
private static class MyAsyncTask extends AsyncTask {
    private final WeakReference<Activity> activityRef;

    public MyAsyncTask(Activity activity) {
        this.activityRef = new WeakReference<>(activity);
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);

        Activity activity = activityRef.get();
        if (activity != null) {
            ...
        }
    }
}
```

Proguard

# Proguard

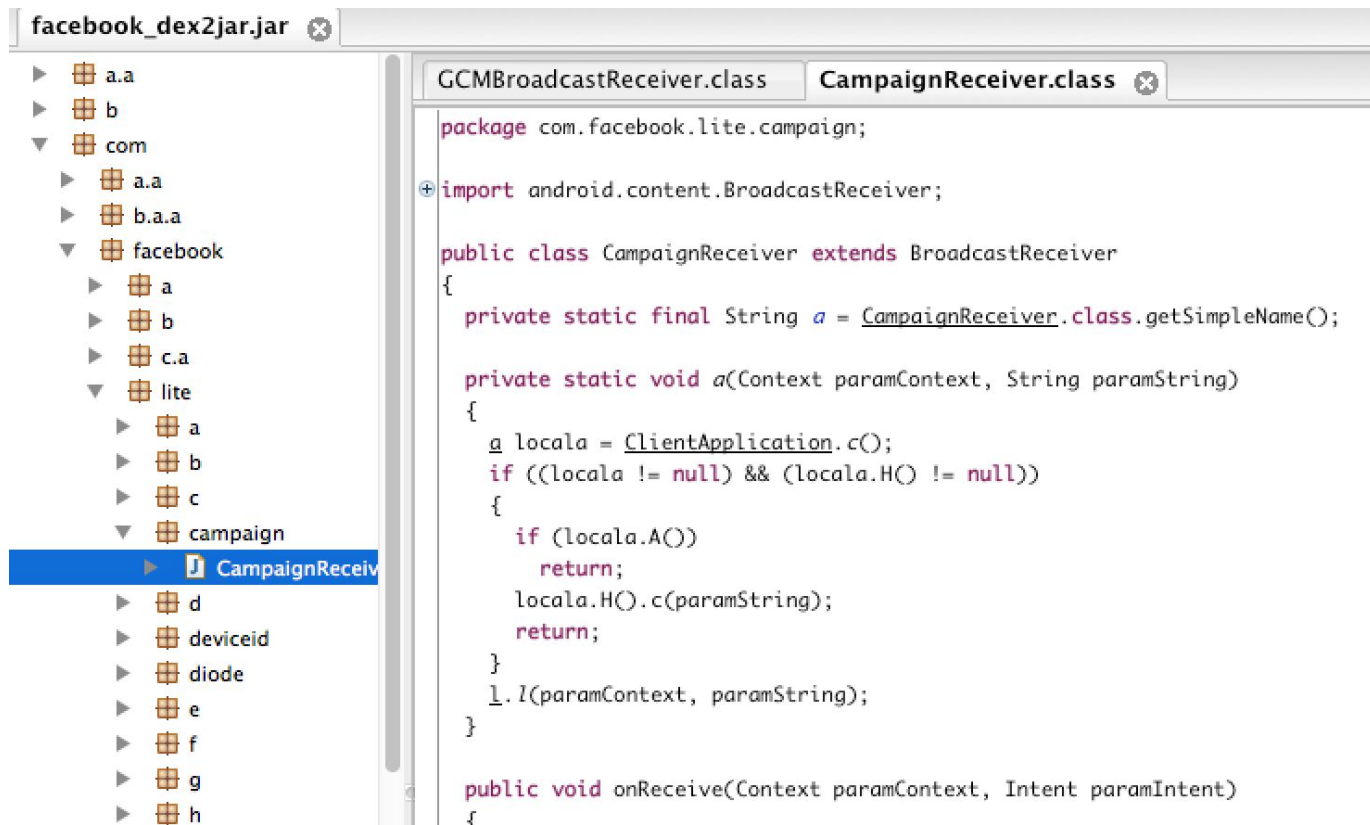
- Удаляет неиспользуемый код
- Оптимизирует
- “Запутывает”

# dex2jar

- Вытаскивает jar из apk



# dex2jar





# build.gradle

```
buildTypes {  
    release {  
        minifyEnabled true  
        proguardFiles  
            getDefaultProguardFile('proguard-android.txt'),  
            'proguard-rules.pro'  
    }  
}
```

Вопросы?