Workshop Android

SKD, NDK & OPENGL ES

PHILIPP GEITZ-MANSTEIN

Inhalt

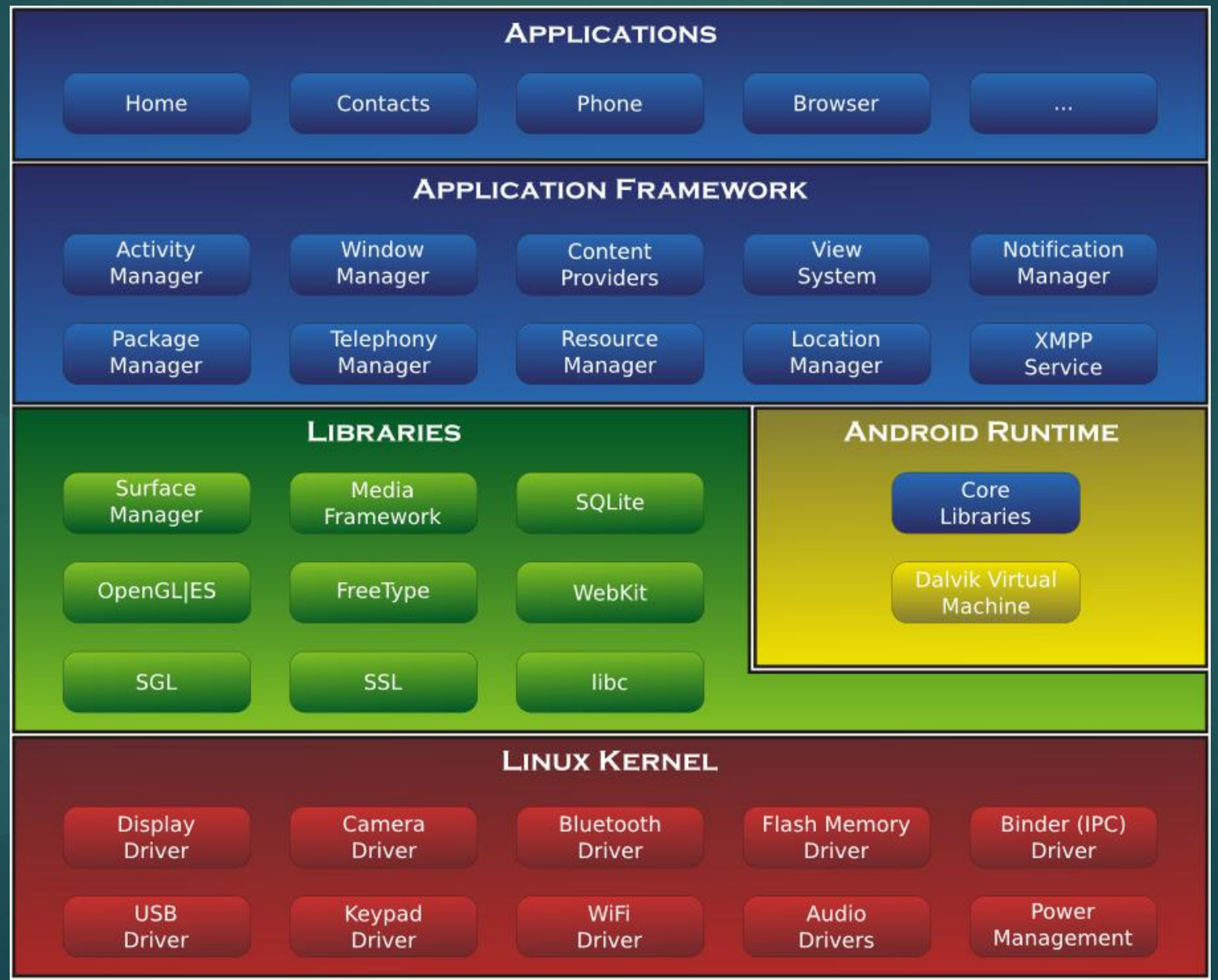
- SDK
- NDK
- OpenGL ES
- Programmieren

Android SDK

Android SDK

- Warum Android?
- Keine Entwicklerlizenzen
- Marktanteile
  - Tablets: 41%
  - Smartphones: 69%
- Entwicklung in Java und/oder C/C++

Android SDK - Architektur



Android SDK – Projekt, Ordner & Emulator

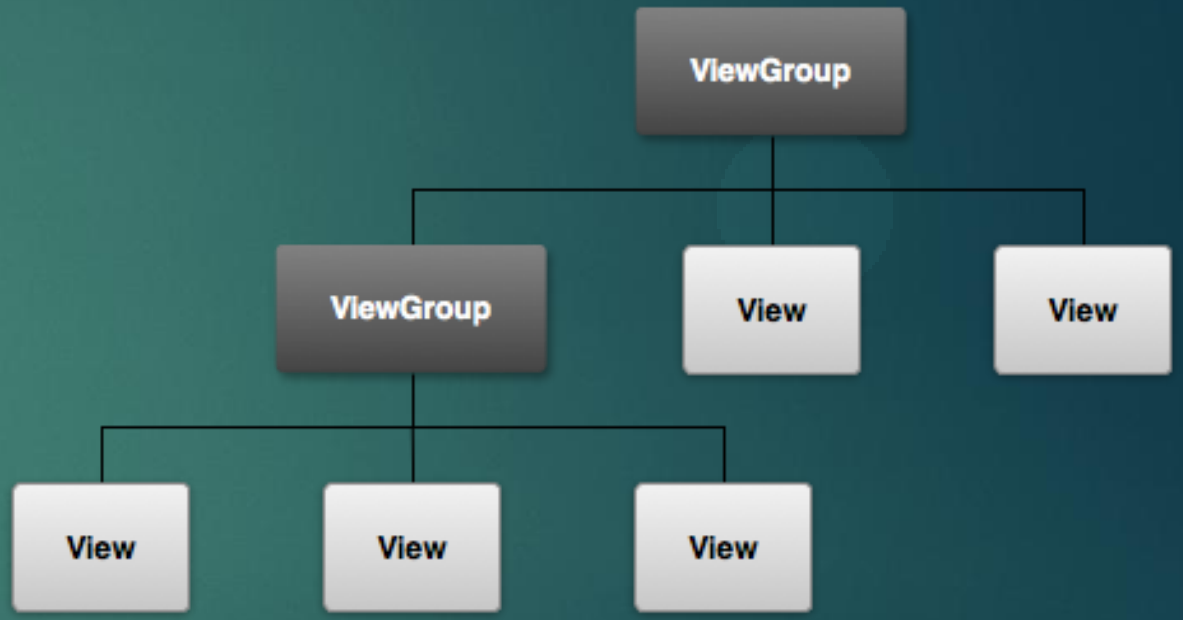
- Projekt erstellen
- Ordnerstruktur
- Emulator erstellen

Android SDK - Activity

- Keine „Main-Methode“
- Sondern spezielle callbacks

Android SDK - Views

- User Interface
- Verschiedene Eigenschaften
  - Z.B. Höhe und Breite
- Besteht aus
  - ViewGroup
    - Z.B. LinearLayout
  - View
    - Z.B. TextView



Android SDK - Views

Beispiel Layout:

```
<LinearLayout
xmlns:android=„http://schemas.android.com/apk/res/android“
    android:layout_width=„fill_parent“
    android:layout_hight=„fill_parent“
    android:orientation=„vertical“>
    <TextView
        android:id=„@+id/textView_hello“
        android:layout_width=„warp_content“
        android:layout_hight=„warp_content“ \>
</LinearLayout>
```

Android SDK - Views

Beispiel Activity:

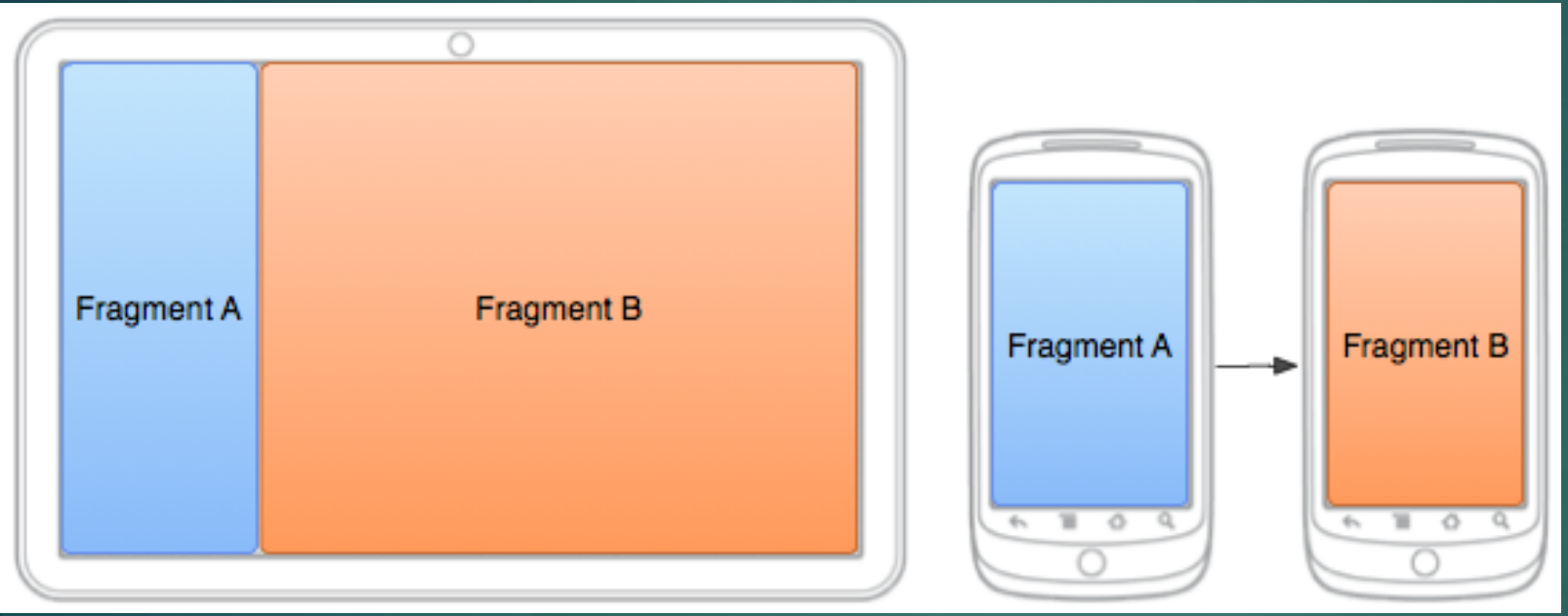
```
public class MyActvity extends Activity{

    private TextView message;

    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        message = (TextView) findViewByID(R.id.textView_hello);
        message.setText(R.string.hello_world);
    }
}
```

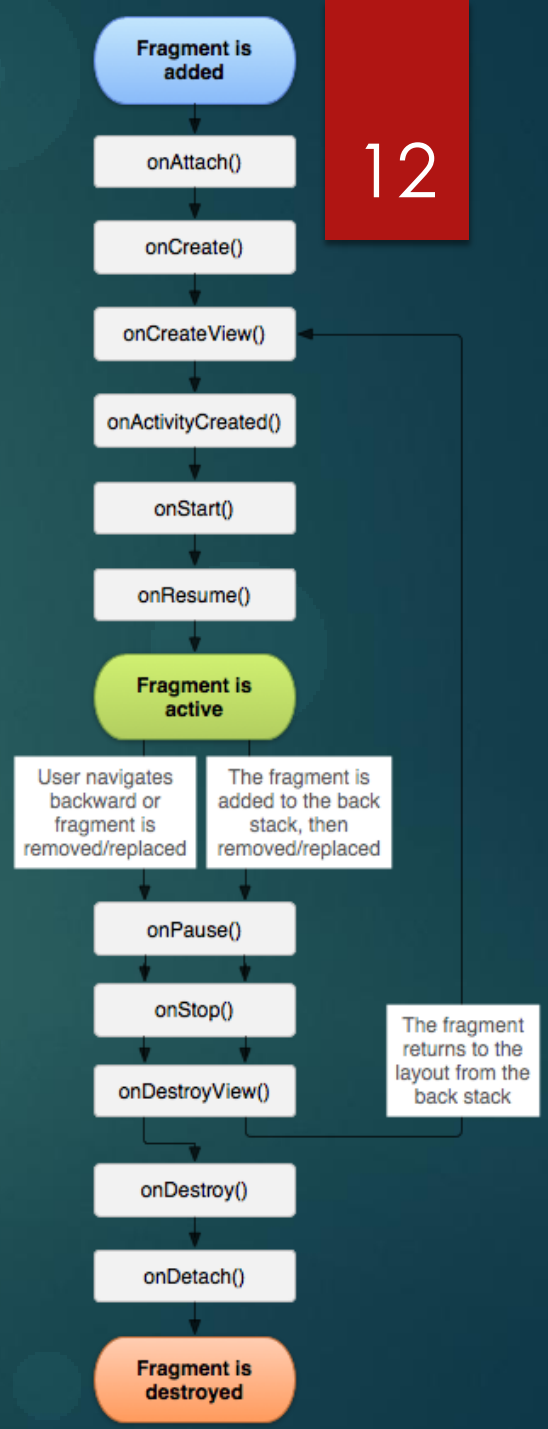
Android SDK - Views

Beispiel Activity erweitert:

```
public class MyActvity extends Activity{
    …
    @Override
    public void onCreate(Bundle savedInstanceState){
        …
        Button b = new Button(this);
        b.setText(„Klick mich!“);
        b.setOnClickListener( new OnClickListener(){
            @Override
                public void onClick(View v){
                    …
            }
        });
        addContentView(b, new LayoutParams(LayoutParams.WRAP_CONTENT,
LayoutParams.WARP_CONTENT));
    }
}
```

Android SDK - Fragments





Android SDK - Fragments

Beispiel Fragment setzen:

```
...
MyFragment fragment = new MyFragment();
FragmentTransaction transaction = getFragmentManager().beginTransaction();
transaction.add(R.id.fragment_container,fragment);
transaction.commit();
...
```

Fragment austauschen:

```
...
NewFragent newFrag = new NewFragment();
FragmentTransaction transaction = getFragmentManager().beginTransaction();
transaction.replace(R.id.fragment_container,newFrag);
Transaction.addToBackStack(null);
transaction.commit();
...
```

Android SDK - AsyncTask


```
Class MyTask extends AsyncTask<Params, Progress, Result>{
    protected Result doInBackground(Params… params){
        …
        if(!isCancelled()){
            …
            publishProgress(…);
            …
        }
        …
    }

    protected void onProgressUpdate(Progress… progress){
        …
    }

    protected void onPostExecute(Result result){
        …
    }
}
```

```
MyTask task = new myTask();
task.execute(…);
…
task.cancel(true);
```

Android NDK

Android NDK

- Entwicklung: C/C++
- Einsatzmöglichkeiten:
  - JNI: C/C++ ↔ Java
  - Native Activity → einige Erweiterungen nicht verfügbar
- Vorteile:
  - Häufig schneller
  - Existierender C/C++ auf Android portieren
- Nachteile:
  - Erhöht Komplexität sehr
- Gute Kandidaten:
  - Eigenständige, CPU intensive, wenig Speicher verbrauchende Programmteile

Android NDK - JNI

- Projekt in C/C++ Projekt konvertieren
- Ordner JNI erstellen

Android NDK - JNI

- Projekt in C/C++ Projekt konvertieren
- Ordner JNI erstellen
- C/C++ Code schreiben
- Libary einbinden im Java Code

Android NDK - JNI

In C/C++-Teil: helloworld.c

```
#include <string.h>
#include <jni.h>

JNIEXPORT jstring JNICALL
Java_mein_package_Klasse_funktionsName(
    JNIEnv* env, jobject obj){

    return (*env)->NewStringUTF(env,“Hello World!“);

}
```

Für C++:

extern „C“{…}

Im Java-Teil: Klasse.java

```
Package mein.package
public class Klasse(){

    static{
        System.loadLibrary(„helloworld“);
    }

    public native String funktionsName();
    …
}
```

Android NDK - JNI

- Projekt in C/C++ Projekt konvertieren
- Ordner JNI erstellen
- C/C++ Code schreiben
- Libary einbinden im Java Code
- Android.mk schreiben

Android NDK – Android.mk


```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE    := libhelloworld
LOCAL_SRC_FILE  := helloworld.c

include $(BUILD_SHARED_LIBRARY)
```

Android NDK - JNI

- Projekt in C/C++ Projekt konvertieren
- Ordner JNI erstellen
- C/C++ Code schreiben
- Libary einbinden im Java Code
- Android.mk schreiben
- Kompilieren

Android NDK - JNI

- $NDK/ndk-build

Gdbserver : [arm-linux-androideabi-4.6] libs/armeabi/gdbserver

Gdbsetup : libs/armeabi/gdb.setup

Compile thumb: helloworld <= helloworld.c

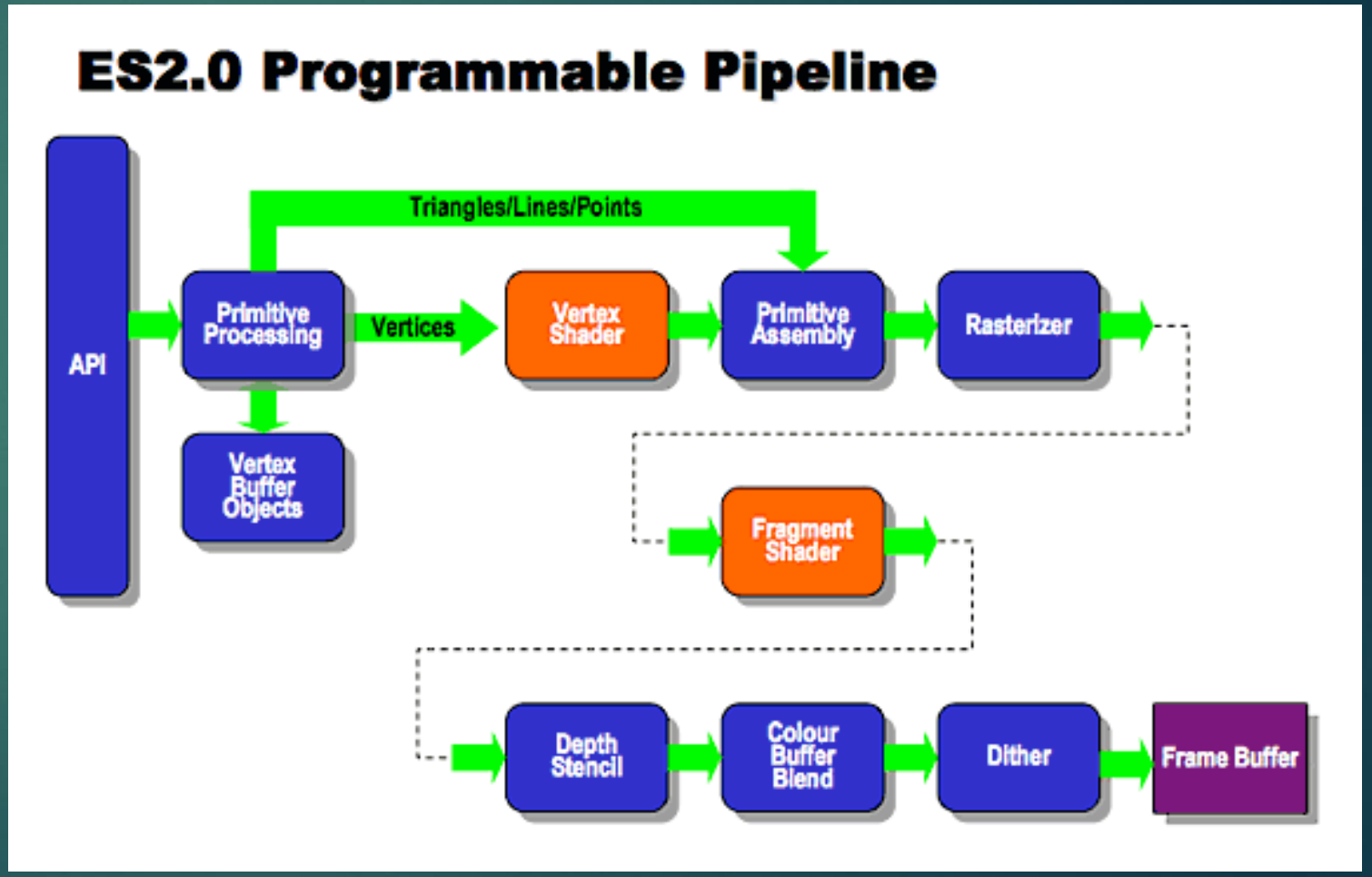
SharedLibrary : libhelloworld.so

Install : libhelloworld.so => libs/armeabi/libhelloworld.so

OpenGL ES

OpenGL ES

- OpenGL ES 1.0/1.1
  - Gegen OpenGL 1.3/1.5 spezifiziert
  - Nur immediate mode
- OpenGL ES 2.0
  - Gegen OpenGL 2.0 spezifiziert
  - Kein immediate mode
  - Vertex- und Fragmentshader
    - OpenGL ES Shading Language 1.0
    - precision mediump float;
  - Nicht abwärtskompatibel
- OpenGL ES 3.0
  - Gegen OpenGL 3.1 spezifiziert
  - Abwärtskompatibel zu 2.0
  - OpenGL ES Shading Language 3.0



OpenGL ES 2.0 Beispiel


```
…
glUseProgram(program);
posHandle = glGetAttribLocation(program,“vPosition“);
glVertexAttribPointer(posHandle, 2, GL_FLOAT, GL_FALSE, 0, vertices);
glEnableVertexAttribArray(posHandle);

glDrawElements(GL_TRIANGLE, 3, GL_UNSIGNED_SHORT, verticesOrder);
…
```

OpenGL ES in Android

- Besondere View
  - GLSurfaceView
    - setEGLContextClientVersion(2)
    - setRenderer(Renderer)
- Braucht Renderer
  - GLSurfaceView.Renderer
    - onDrawFrame
    - onSurfaceChanged
    - onSurfaceCreated

Programmieren

Programmieren

- https://github.com/duglah/workshop