libGDX

$\mathcal{TK}$

GODOT
Game engine

# Introduction

lib

libG

libGD

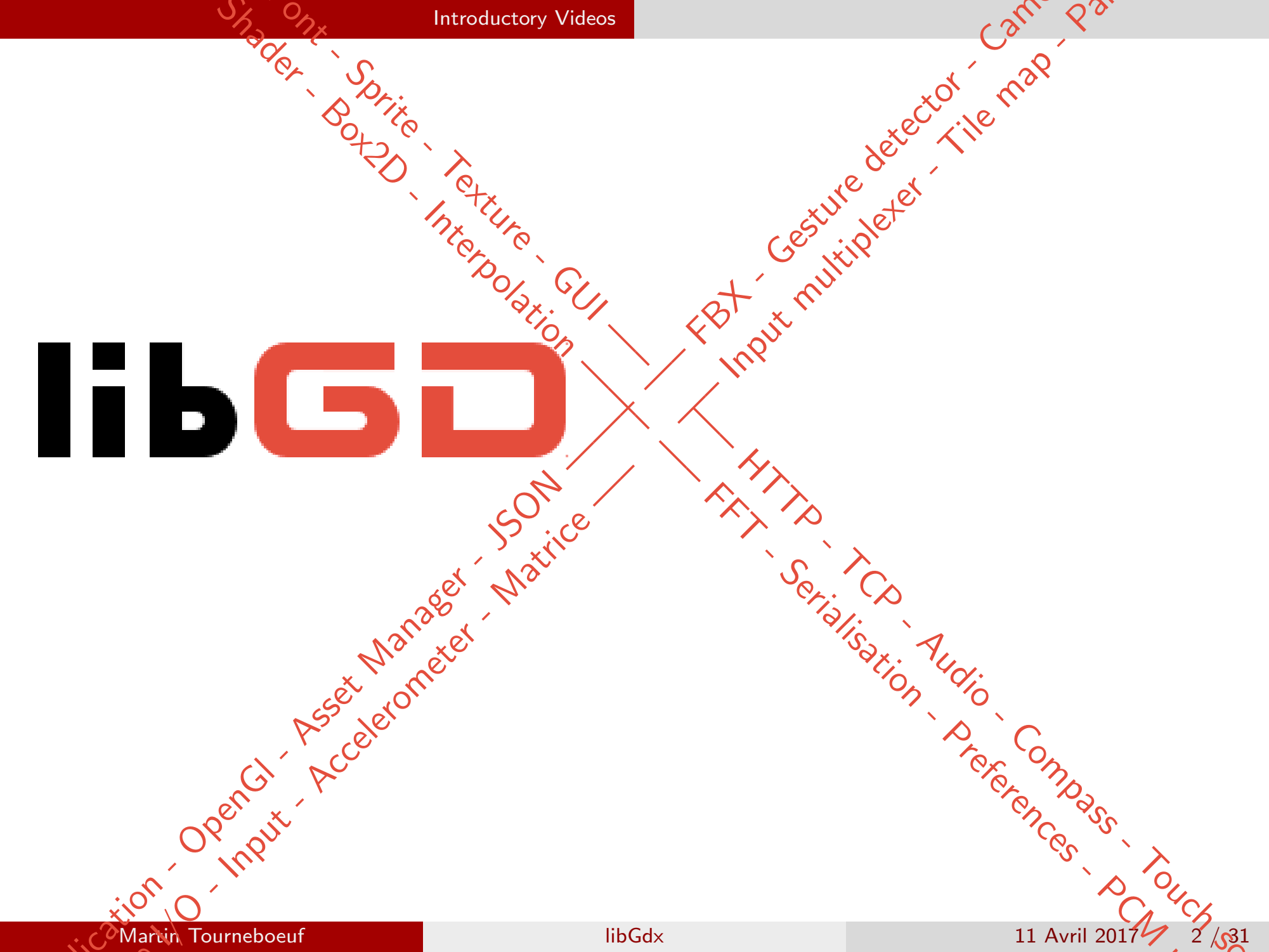
libGD

FBX - Gesture detector - Came
Input multiplexer - Tile map - Pa

Shader - Box2D - Interpolation - Sprite - Texture - GUI - FBX - Gesture detector - Input multiplexer - Tile map

Shader - Box2D - Interpolation - Sprite - Texture - GUI - FBX - Gesture detector - Input multiplexer - Tile map - JSON - Matrice - Asset Manager - OpenGl - Accelerometer - Input - I/O

# libGD


libGD
Shader - Box2D - Interpolation - Sprite - Texture - GUI
FBX - Gesture detector - Input multiplexer - Tile map
JSON - Matrice
HTTP - TCP - Audio - Compass
FFT - Serialisation - Preferences - PCM
Application - OpenGL - Asset Manager
I/O - Input - Accelerometer

# libGDX

Shader - Box2D - Interpolation - Sprite - Texture - GUI - FBX - Gesture detector - Input multiplexer - Tile map - JSON - Asset Manager - OpenGl - Matrice - Accelerometer - Input - I/O - HTTP - TCP - Audio - Compass - Touch - FFT - Serialisation - Preferences - PCM

# the Java open-source cross-platform

libGDX

Shader - Box2D - Font - Sprite - Texture - Interpolation - GUI - FBX - Gesture detector - Camera - Input multiplexer - Tile map - Pa

JSON - Matrice - HTTP - TCP - Audio - Compass - Touch s

Asset Manager - FFT - Serialisation - Preferences - PCM

Application - OpenGl - Accelerometer - I/O - Input

# the Java open-source cross-platform

# libGDX

# game framework


— Introductory Videos

1. Overview of libGdx
   - What is libGdx ?
   - Who uses libGdx ?
   - How can all that be “easy” ?
2. Low level bindings example : openGL
   - Android
   - Desktop
   - GLES
3. High level Api
   - Architecture
   - Tests

— Conclusion

# Plan

— Introductory Videos

1. Overview of libGdx
   - What is libGdx ?
   - Who uses libGdx ?
   - How can all that be “easy” ?
2. Low level bindings example : openGL
   - Android
   - Desktop
   - GLES
3. High level Api
   - Architecture
   - Tests

— Conclusion


"libgdx is a cross-platform game development framework. Write your game once in Java, deploy to Windows, Linux, Mac OS X, Steam, Android, iOS, HTML5/WebGL"

- Initiated in 2010 by Mario Zechner
- Open Source : `https://github.com/libgdx/libgdx`
- Community : 5000 forks + nice wiki
- Website : `http://libgdx.badlogicgames.com` with a game repo


"libgdx is a cross-platform game development framework. Write your game once in Java, deploy to Windows, Linux, Mac OS X, Steam, Android, iOS, HTML5/WebGL"

- Initiated in 2010 by Mario Zechner
- Open Source : `https://github.com/libgdx/libgdx`
- Community : 5000 forks + nice wiki
- Website : `http://libgdx.badlogicgames.com` with a game repo

"LibGdx is big framework accessible with little difficutly"


"libgdx is a cross-platform game development framework. Write your game once in Java, deploy to Windows, Linux, Mac OS X, Steam, Android, iOS, HTML5/WebGL"

- Initiated in 2010 by Mario Zechner
- Open Source : `https://github.com/libgdx/libgdx`
- Community : 5000 forks + nice wiki
- Website : `http://libgdx.badlogicgames.com` with a game repo

"LibGdx is big framework accessible with little difficutly" (me)

# Features

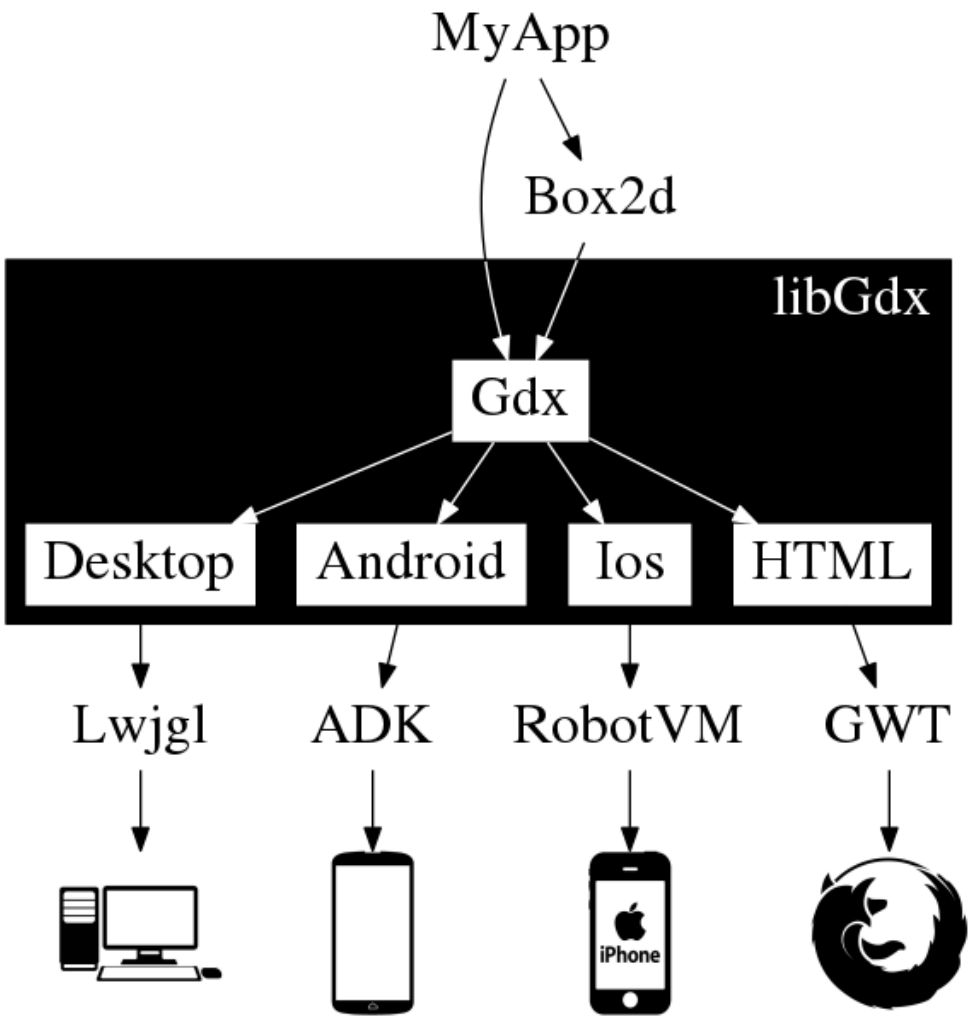
| Low level | Mid Level | High Level |
|---|---|---|
| Graphics | Texture<br>SpriteBatch | UI skinnable<br>3D (fbx support) |
| Input | Gesture detector<br>Input multiplexer | Box2d<br>Bullet (via jni) |
| File I/O | AssetManager<br>Threaded IO | Serialisation |
| Audio | Math Utility | FFT<br>mp3/ogg decoding |
| Application Managment | Disposable | Screen management |
| Networking | !! | Extensions : freetype |

| | | | |
|---|---|---|---|
| unity3 | Unity 3D | Unity is a feature rich, fully integrated development engine for the creation of interactive 3D con... | 7.90% of apps<br>11.63% of installs |
| | Cocos2D-X | Cocos2d-x is an open-source mobile 2D game framework written in C++. | 2.10% of apps<br>4.41% of installs |
| libGDX | libgdx | The libgdx project is a cross-platform game development library written in Java with some JNI code ... | 1.52% of apps<br>2.53% of installs |
| | AndEngine | AndEngine is a free Android 2D OpenGL Game Engine. | 0.85% of apps<br>0.82% of installs |
| YOYOGAMES | Gamemaker: Studio | Cross platform game development framework | 0.42% of apps<br>0.24% of installs |
| cocos2d | Cocos2D | Cocos2d is an open-source mobile 2D game framework. | 0.26% of apps<br>0.20% of installs |

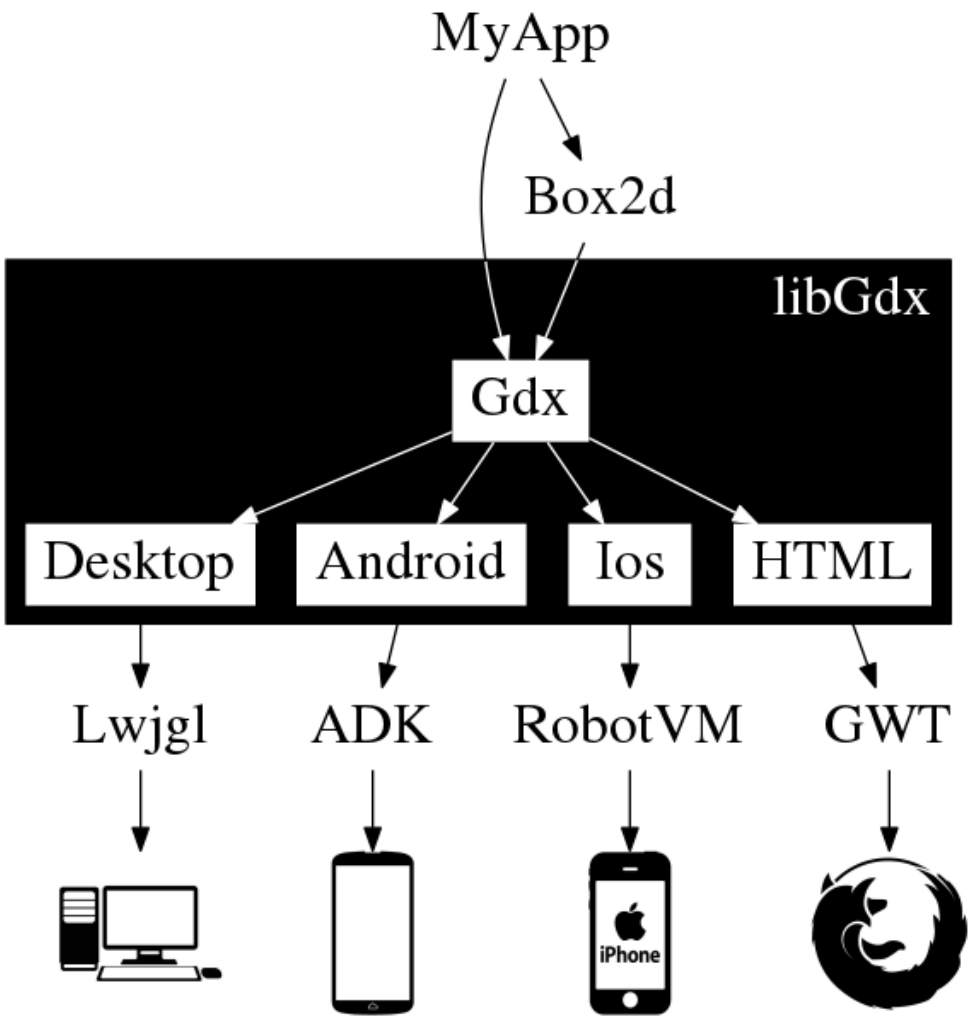
From `www.appbrain.com/stats/libraries/tag/game-framework/android-game-frameworks`


+10M download.
From `https://play.google.com/store/apps/details?id=com.nianticproject.ingress`

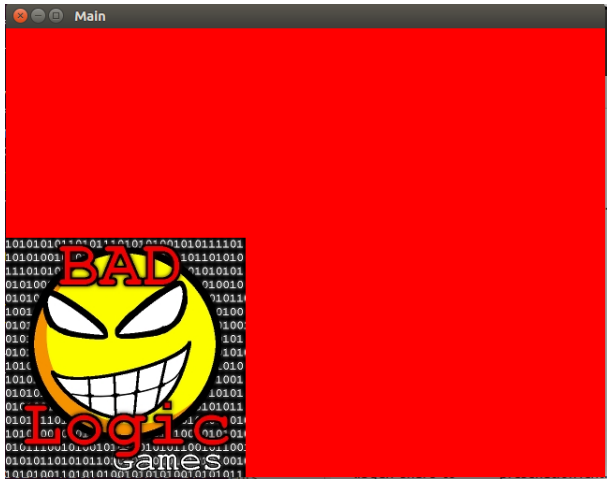

+500k download at 2$.
From `https://play.google.com/store/apps/details?id=com.bithack.apparatus`


MyApp
Box2d
libGdx
Gdx
Desktop
Android
Ios
HTML
Lwjgl
ADK
RobotVM
GWT
iPhone

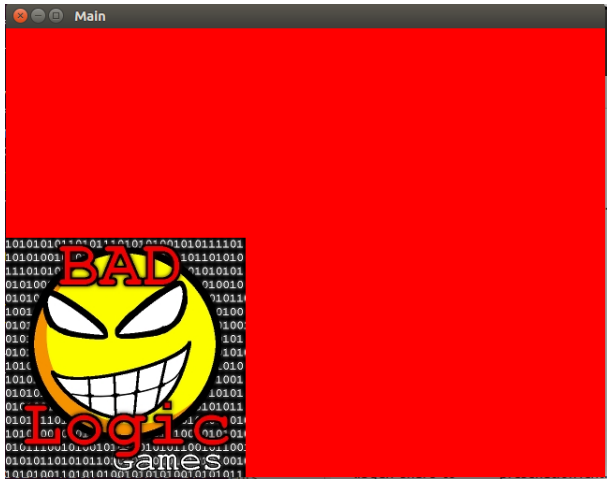
```java
package com.tutorial.helloGame;

import com.badlogic.gdx.ApplicationAdapter;
import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.Texture;
import com.badlogic.gdx.graphics.g2d.
      SpriteBatch;

public class HellloGame extends
      ApplicationAdapter {
  SpriteBatch batch;
  Texture img;

  @Override
  public void create () {
    batch = new SpriteBatch();
    img = new Texture("badlogic.jpg");
  }

  @Override
  public void render () {
    Gdx.gl.glClearColor(1, 0, 0, 1);
    Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT)
          ;
    batch.begin();
    batch.draw(img, 0, 0);
    batch.end();
  }
}
```

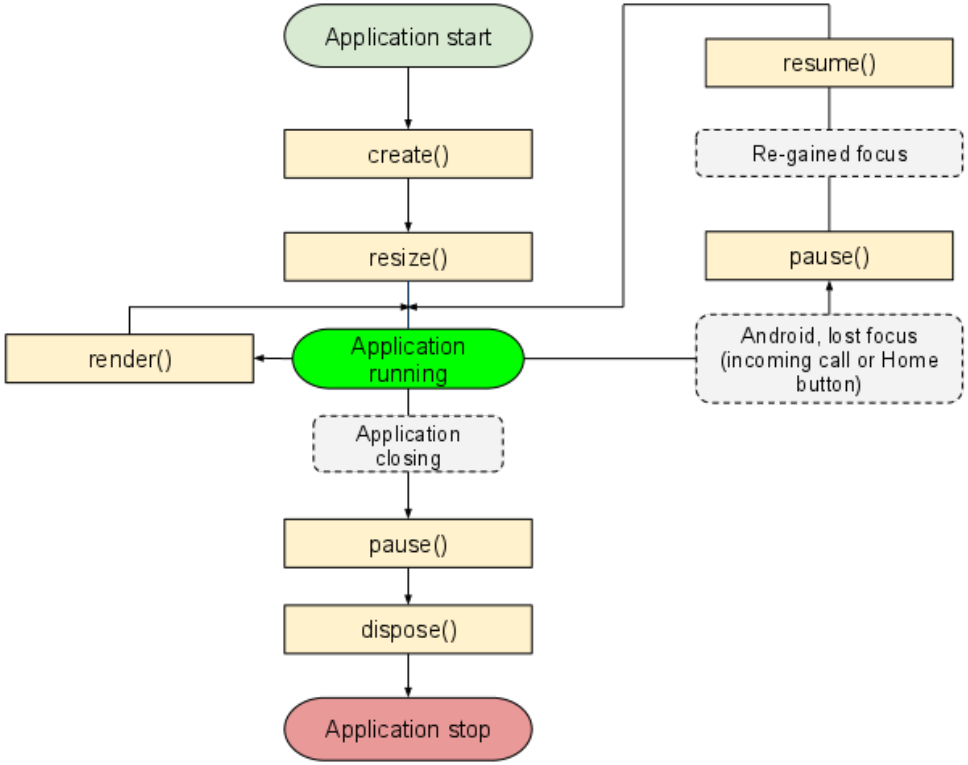


```java
package com.tutorial.helloGame;

import com.badlogic.gdx.ApplicationAdapter;
import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.Texture;
import com.badlogic.gdx.graphics.g2d.
      SpriteBatch;

public class HellloGame extends
      ApplicationAdapter {
  SpriteBatch batch;
  Texture img;

  @Override
  public void create () {
    batch = new SpriteBatch();
    img = new Texture("badlogic.jpg");
  }

  @Override
  public void render () {
    Gdx.gl.glClearColor(1, 0, 0, 1);
    Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT)
          ;
    batch.begin();
    batch.draw(img, 0, 0);
    batch.end();
  }
}
```



```java
package com.tutorial.helloGame;

import com.badlogic.gdx.ApplicationAdapter;
import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.Texture;
import com.badlogic.gdx.graphics.g2d.
      SpriteBatch;

public class HellloGame extends
      ApplicationAdapter {
  SpriteBatch batch;
  Texture img;

  @Override
  public void create () {
    batch = new SpriteBatch();
    img = new Texture("badlogic.jpg");
  }

  @Override
  public void render () {
    Gdx.gl.glClearColor(1, 0, 0, 1);
    Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT)
          ;
    batch.begin();
    batch.draw(img, 0, 0);
    batch.end();
  }
}
```



### Desktop

```
LwjglApplication
LwjglCanvas
LwjglAudio
LwjglFiles
LwjglFileHandle
LwjglGL10
LwjglGL11
LWjglGL20
LwjglGraphics
LwjglInput
LwjglMusic
LwjglNet
LwjglPreferences
LwjglServerSocket
LwjglSocket
LwjglSound
```

### IOS

```
IOSApplication
IOSAudio
IOSFiles
IOSFileHandle
IOSGL20
IOSGraphics
IOSInput
IOSMusic
IOSNet
IOSPreferences
IOSServerSocket
IOSSocket
IOSSound
```

### Core Interfaces

```
Application
Audio
Files
FileHandle
GL10
GL11
GL20
Graphics
Input
Music
Net
Preferences
ServerSocket
Socket
Sound
```

All mid-level and high-level classes of libgdx classes use these interfaces

### HTML/WebGL

```
GwtApplication
GwtAudio
GwtFiles
GwtFileHandle
GwtInput
GwtGL20
GwtGraphics
GwtInput
GwtMusic
GwtNet
GwtPreferences
GwtSound
```

### Android

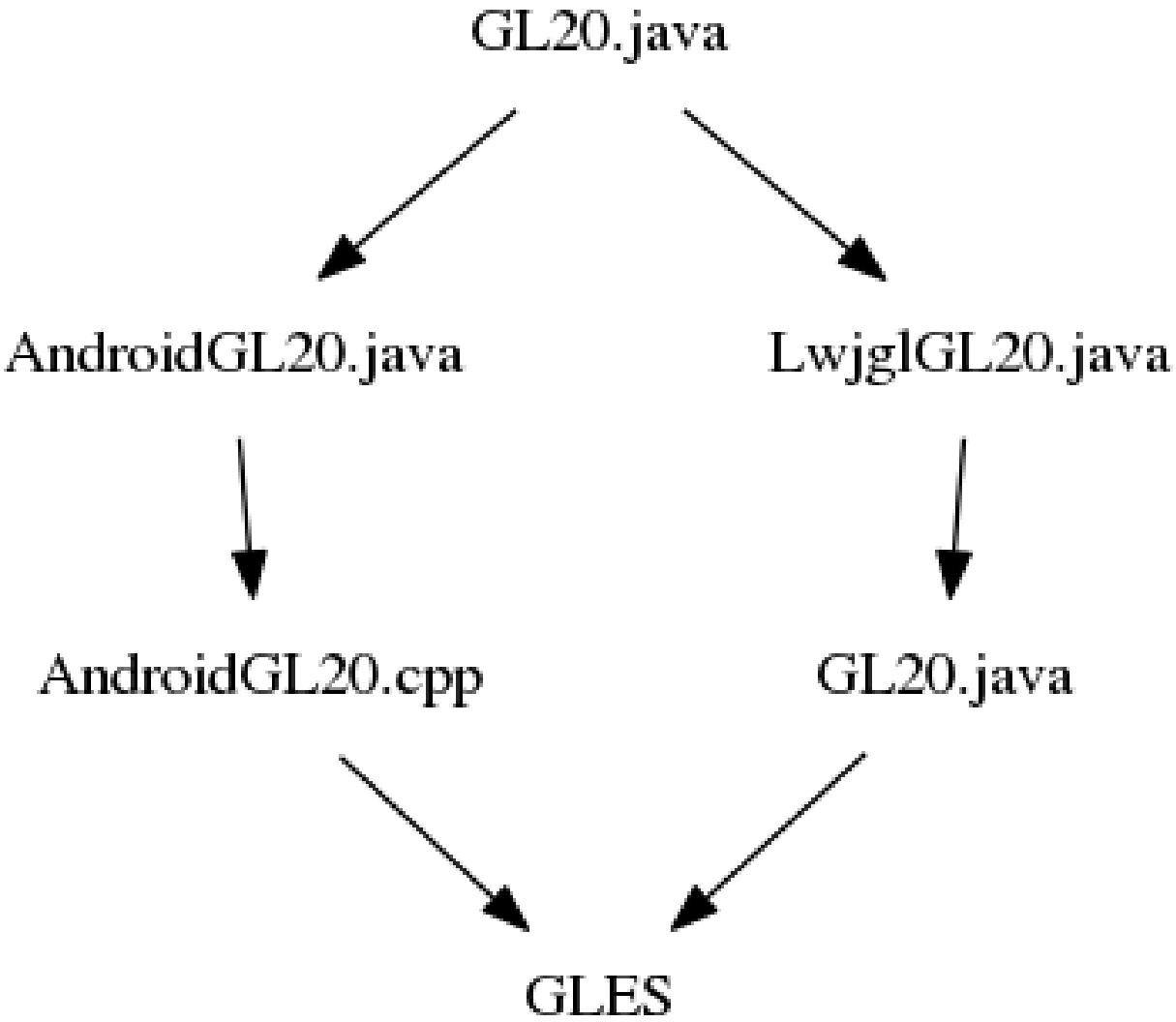
```
AndroidApplication
AndroidWallpaper
AndroidDaydream
AndroidAudio
AndroidFiles
AndroidFileHandle
AndroidGL10
AndroidGL11
AndroidGL20
AndroidGraphics
AndroidInput
AndroidMusic
AndroidNet
AndroidPreferences
AndroidServerSocket
AndroidSocket
AndroidSound
```

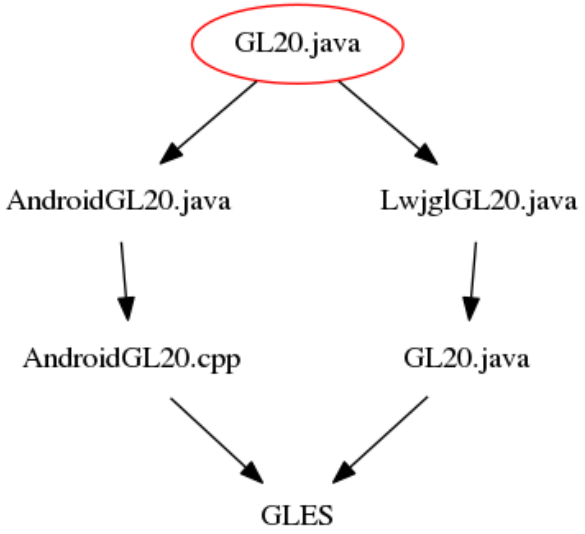
# Plan

— Introductory Videos

1. Overview of libGdx
   - What is libGdx ?
   - Who uses libGdx ?
   - How can all that be “easy” ?
2. Low level bindings example : openGL
   - Android
   - Desktop
   - GLES
3. High level Api
   - Architecture
   - Tests

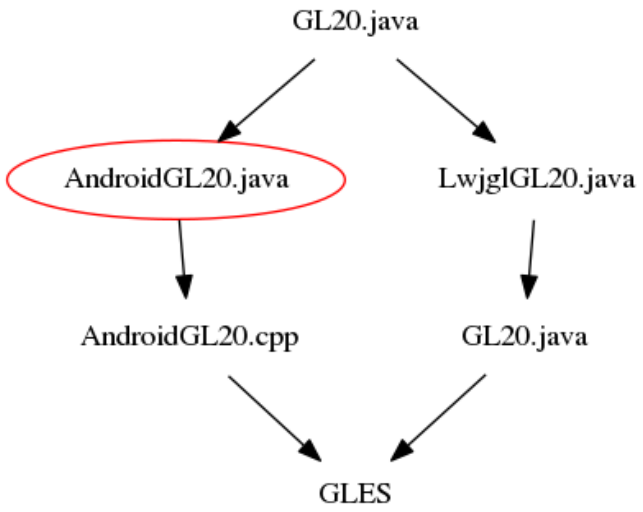
— Conclusion

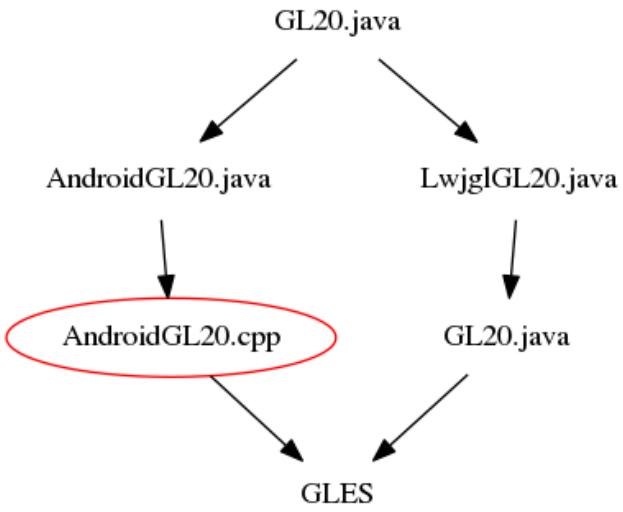

GL20.java
AndroidGL20.java
LwjglGL20.java
AndroidGL20.cpp
GL20.java
GLES



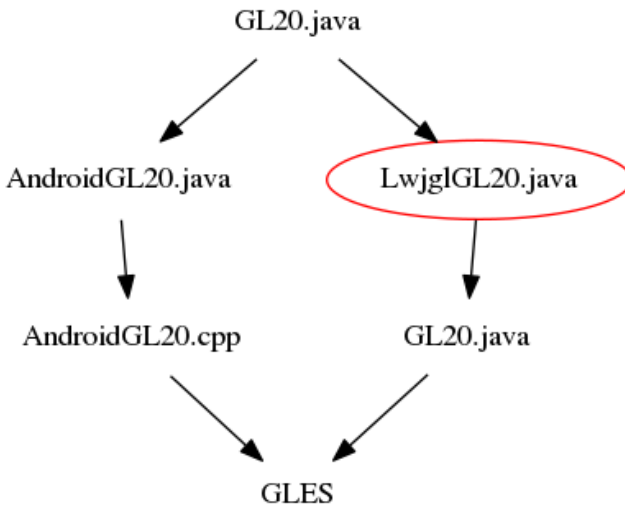

```
1  package com.badlogic.gdx.graphics;
2
3  import java.nio.Buffer;
4  import java.nio.FloatBuffer;
5  import java.nio.IntBuffer;
6
7  public interface GL20 {
8      public void glActiveTexture (int texture);
9      public void glBindTexture (int target, int texture);
10     public void glBlendFunc (int sfactor, int dfactor);
11     public void glClear (int mask);
12 }
```

```java
package com.badlogic.gdx.backends.android

import java.nio.Buffer;
import java.nio.FloatBuffer;
import java.nio.IntBuffer;

import com.badlogic.gdx.graphics.GL20;

public class AndroidGL20 implements GL20 {
    static {
        System.loadLibrary("gdx");
        init();
    private static native void init ();
    public native void glActiveTexture (int texture);
    public native void glAttachShader (int program, int
        shader);
    public native void glBindAttribLocation (int program, int
        index, String name);
}}
```

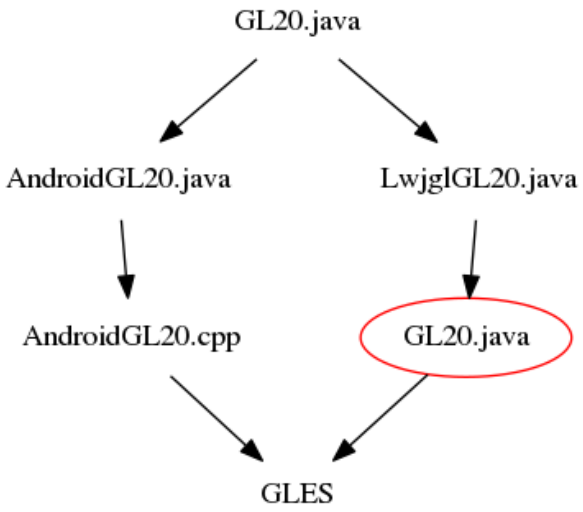





```
#include "AndroidGL20.h"
#include <GLES2/gl2.h>
#include <GLES2/gl2ext.h>
/*
 * Class:     com_badlogic_gdx_backends_android_AndroidGL20
 * Method:    glActiveTexture
 * Signature: (I)V
 */
JNIEXPORT void JNICALL
    Java_com_badlogic_gdx_backends_android_AndroidGL20_glActiveT
    (JNIEnv *, jobject, jint texture)
{
glActiveTexture( texture );
}
```

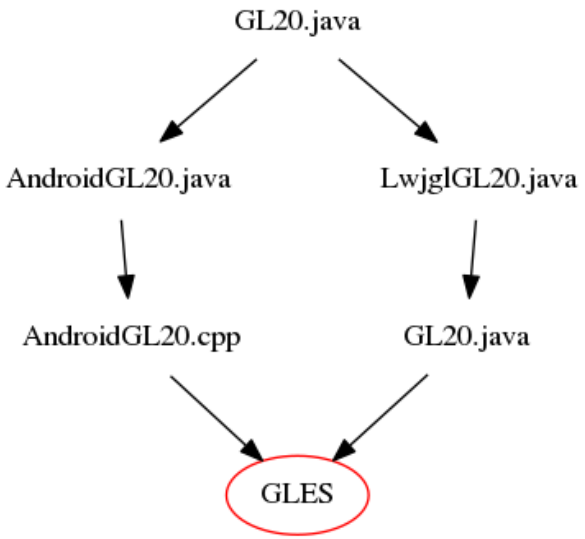


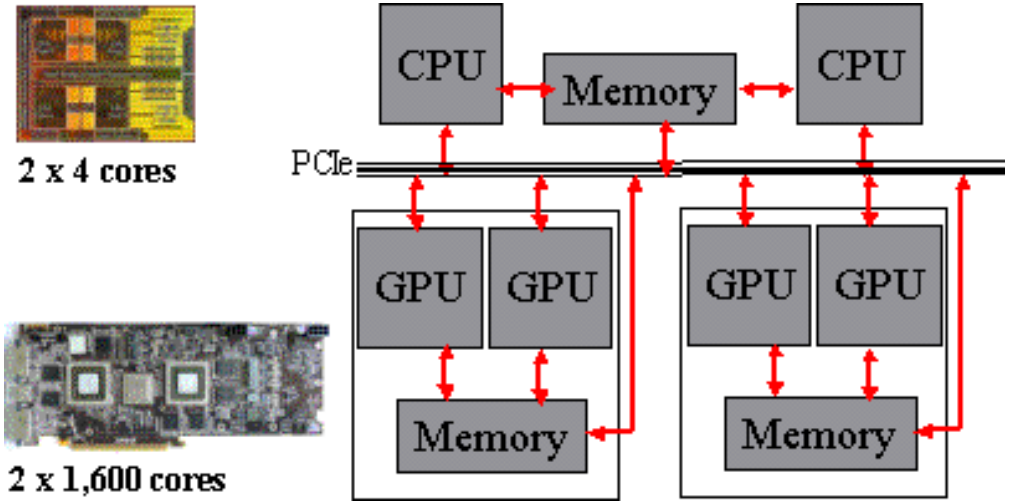

```java
package com.badlogic.gdx.backends.lwjgl;

import org.lwjgl.opengl.GL13;
import com.badlogic.gdx.utils.GdxRuntimeException;

class LwjglGL20 implements com.badlogic.gdx.graphics.GL20 {
    public void glActiveTexture (int texture) {
        GL13.glActiveTexture(texture);
    }
}
```




```
package org.lwjgl.opengl;

import org.lwjgl.util.generator.*;
import org.lwjgl.util.generator.opengl.*;

import java.nio.*;

@DeprecatedGL
public interface GL13 {
    void glActiveTexture(@GLenum int texture);
}
```

```
void GLAPIENTRY
_mesa_ActiveTexture(GLenum texture)
{
   const GLuint texUnit = texture - GL_TEXTURE0;
   GLuint k;
   GET_CURRENT_CONTEXT(ctx);

   if (MESA_VERBOSE & (VERBOSE_API|VERBOSE_TEXTURE))
      _mesa_debug(ctx, "glActiveTexture %s\n",
                  _mesa_enum_to_string(texture));

   if (ctx->Texture.CurrentUnit == texUnit)
      return;

   k = _mesa_max_tex_unit(ctx);

   assert(k <= ARRAY_SIZE(ctx->Texture.Unit));

   if (texUnit >= k) {
      _mesa_error(ctx, GL_INVALID_ENUM, "glActiveTexture(texture=%s)",
                  _mesa_enum_to_string(texture));
      return;
   }

   FLUSH_VERTICES(ctx, _NEW_TEXTURE);

   ctx->Texture.CurrentUnit = texUnit;
   if (ctx->Transform.MatrixMode == GL_TEXTURE) {
      /* update current stack pointer */
      ctx->CurrentStack = &ctx->TextureMatrixStack[texUnit];
   }
}
```

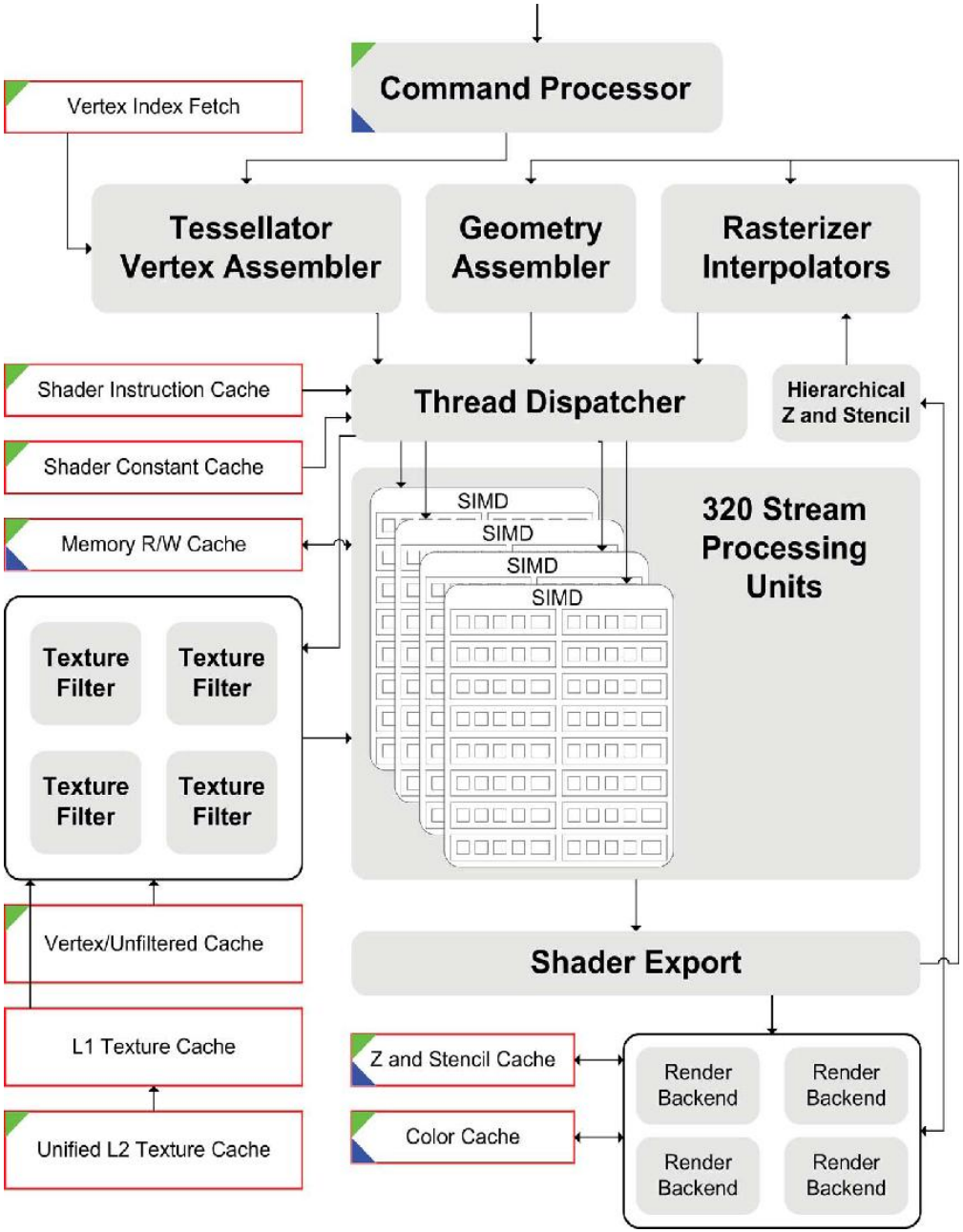





- High latency (1.1 ms vs 0.07 ms)
- High bandwidth (200 Gb/s vs 20 Gb/s for RAM)
- Buffering

```
!!ARBvp1.0
TEMP vertexClip;
DP4 vertexClip.x, state.matrix.mvp.row[0], vertex.position;
DP4 vertexClip.y, state.matrix.mvp.row[1], vertex.position;
DP4 vertexClip.z, state.matrix.mvp.row[2], vertex.position;
DP4 vertexClip.w, state.matrix.mvp.row[3], vertex.position;
MOV result.position, vertexClip;
MOV result.color, vertex.color;
MOV result.texcoord[0], vertex.texcoord;
END
```

```
!!ARBfp1.0
TEMP color;
MUL color, fragment.texcoord[0].y, 2.0;
ADD color, 1.0, -color;
ABS color, color;
ADD result.color, 1.0, -color;
MOV result.color.a, 1.0;
END
```


Command Processor
Vertex Index Fetch
Tessellator Vertex Assembler
Geometry Assembler
Rasterizer Interpolators
Shader Instruction Cache
Thread Dispatcher
Hierarchical Z and Stencil
Shader Constant Cache
Memory R/W Cache
SIMD
SIMD
SIMD
SIMD
320 Stream Processing Units
Texture Filter
Texture Filter
Texture Filter
Texture Filter
Vertex/Unfiltered Cache
Shader Export
L1 Texture Cache
Z and Stencil Cache
Unified L2 Texture Cache
Color Cache
Render Backend
Render Backend
Render Backend
Render Backend

# A trip through the Graphics Pipeline 2011: Index

- Part 1: Introduction; the Software stack.
- Part 2: GPU memory architecture and the Command Processor.
- Part 3: 3D pipeline overview, vertex processing.
- Part 4: Texture samplers.
- Part 5: Primitive Assembly, Clip/Cull, Projection, and Viewport transform.
- Part 6: (Triangle) rasterization and setup.
- Part 7: Z/Stencil processing, 3 different ways.
- Part 8: Pixel processing  fork phase.
- Part 9: Pixel processing  join phase.
- Part 10: Geometry Shaders.
- Part 11: Stream-Out.
- Part 12: Tessellation.
- Part 13: Compute Shaders.

https://fgiesen.wordpress.com/2011/07/09/a-trip-through-the-graphics-pipeline-2011-index/

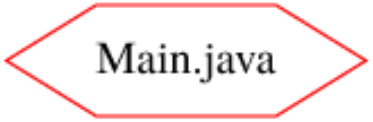
# Plan

— Introductory Videos

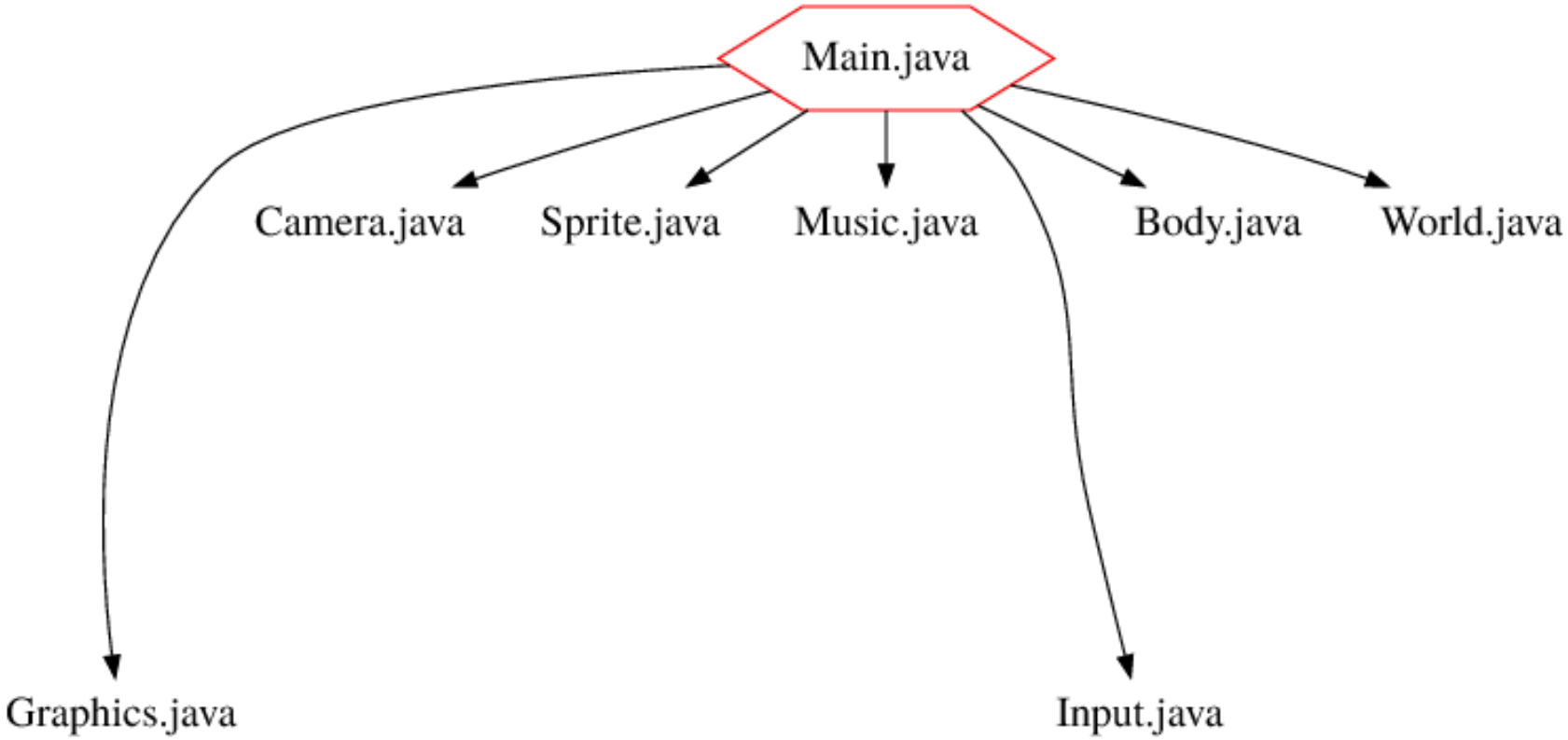
1. Overview of libGdx
   - What is libGdx ?
   - Who uses libGdx ?
   - How can all that be “easy” ?
2. Low level bindings example : openGL
   - Android
   - Desktop
   - GLES
3. High level Api
   - Architecture
   - Tests

— Conclusion

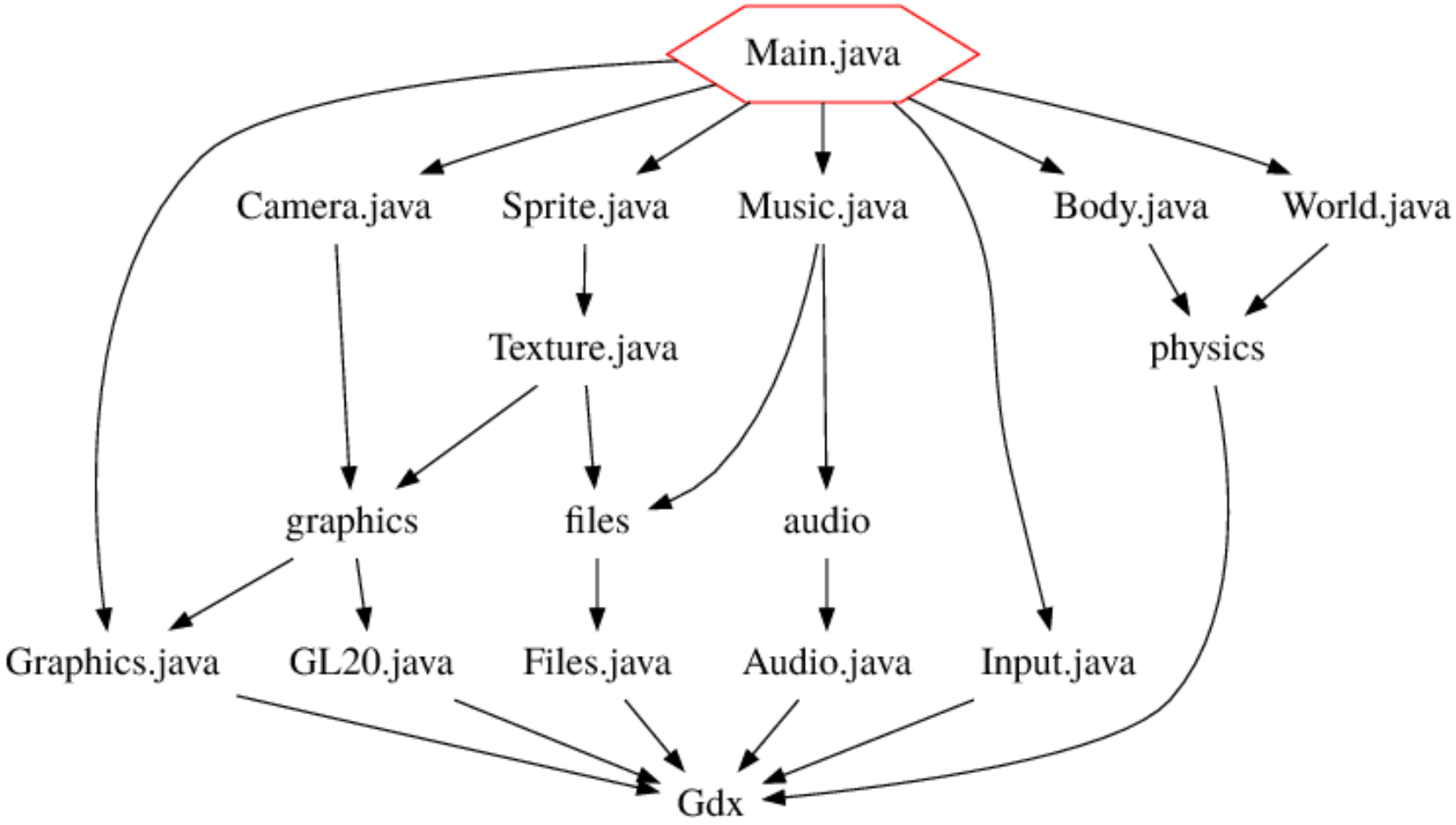
# Architecture



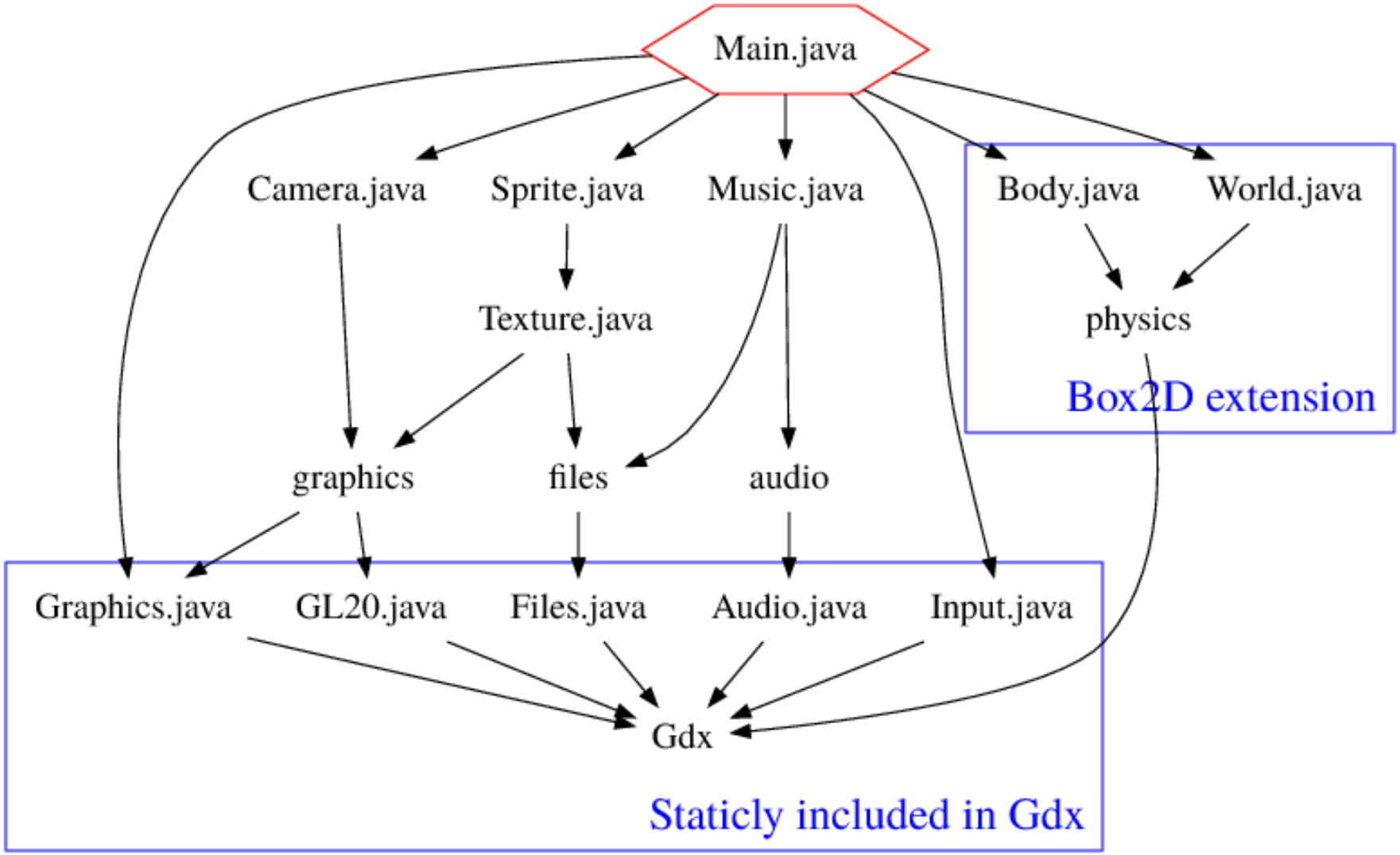
# Architecture



# Architecture

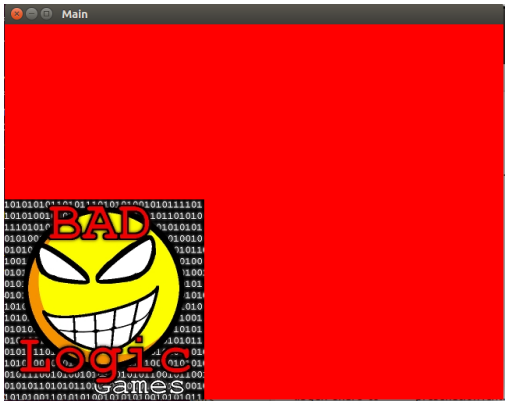


# Architecture

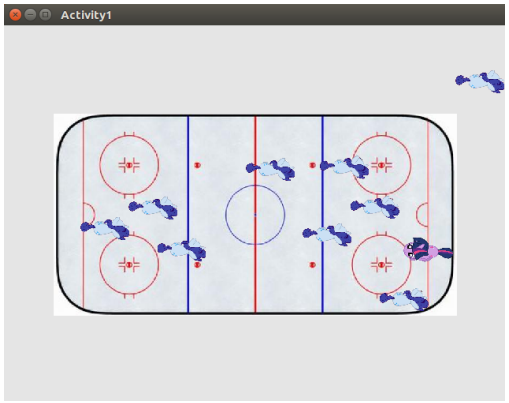


# Personal Tests

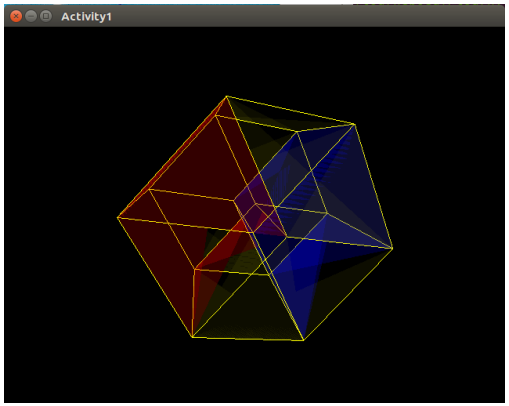
New project
(Auto generated)



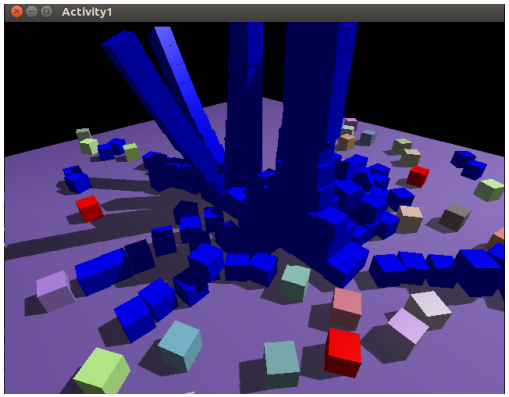

Baby Pony Lost on Ice
(Box2d, sprite)



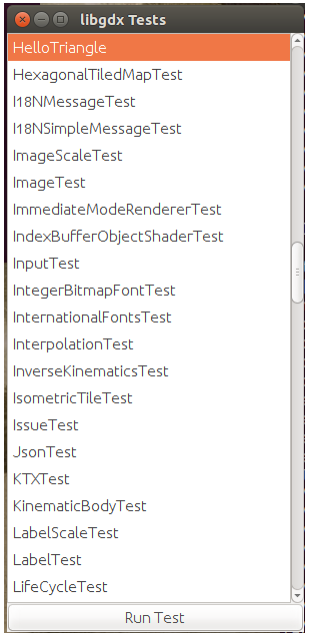

Hypercube
(3D Model)



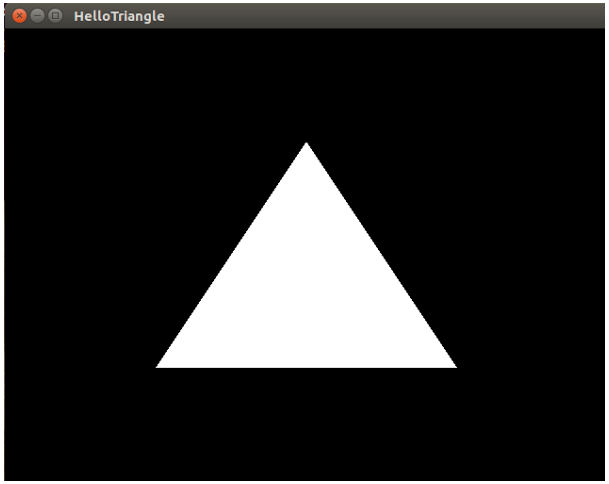

Tower destroy
(bullet)



# Official Tests



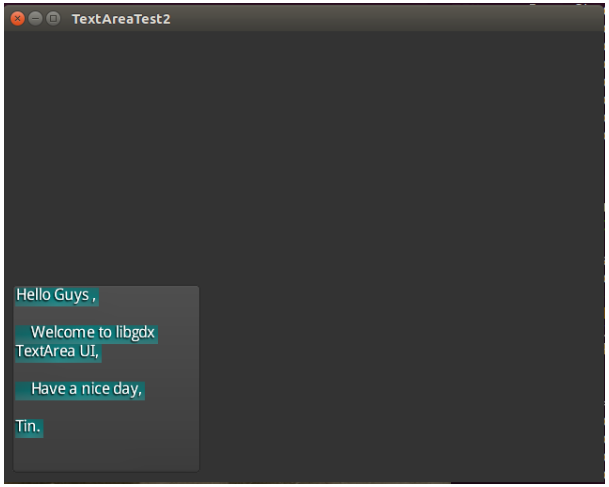

## Triangle




## Animation


## Material

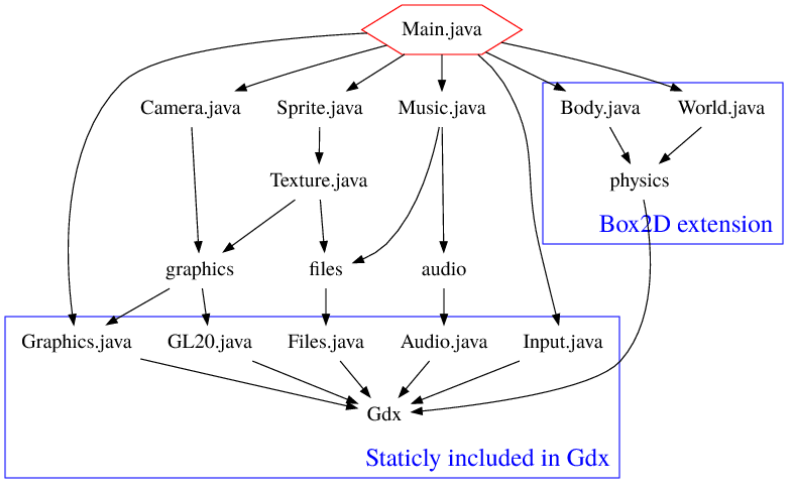

## Text Area



# Conclusion

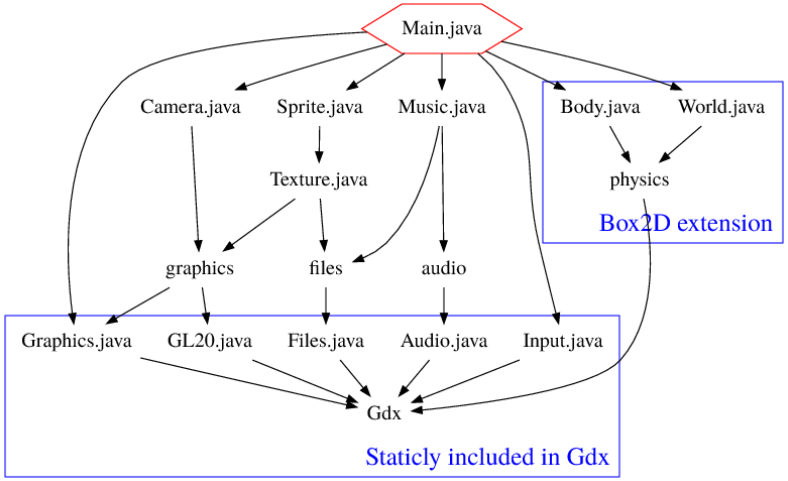
"LibGdx is big framework accessible with little difficutly"

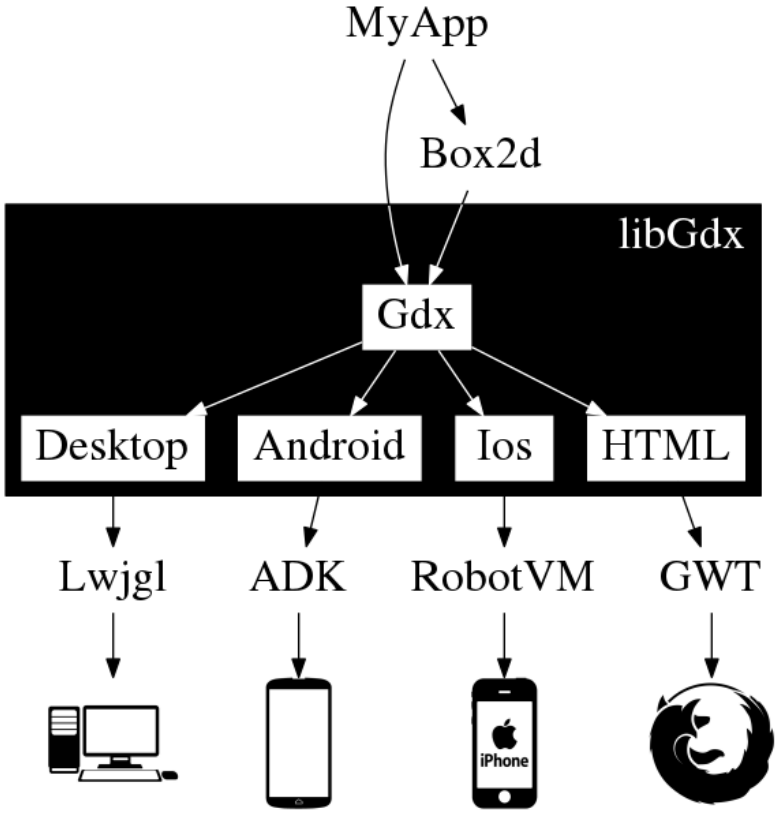
# Conclusion




"LibGdx is big framework accessible with little difficutly"

# Conclusion




"LibGdx is big framework accessible with little difficutly"

# More

- Gradle
- Google Web Toolkit, JavaScript, Dart
- Extensions (box2D, bullet, freetype, visUI (skin), video, ai, pay)
- 3D models : Blender
- Java standard lib (io, files)
- Preferences saving
- Events, Application life cycle.
- File types : Midi, mp3, ogg
- OpenGameArt.com
- Game engines : Unreal 4

# Question


Any Questions ?