

Games in Android - OpenGL ES - Libgdx (code: ANDROID-GAMES)

Overview

The training concentrates on the technical side of creating games for Android devices. It focuses on **effective graphics support and 2D/3D animation**.

Working with graphics for mobile devices requires not only the knowledge of existing libraries and solutions but also understanding the low-level issues: OpenGL ES 2.0 standard and rendering pipeline. That's why the course involves two complementary and interwoven issues:

1. **Open GL ES 2.0**- standard of supporting graphics in mobile and embedded devices, together with GLSL language (Shading Language) used to write shaders;
2. **Libgdx** - leading framework for creating cross-platform (Android, HTML5, simple desktop application) games in Java.

Libgdx includes turnkey solutions for many problems encountered while writing games: loading data of different formats (level maps compatible with *tiled*, three-dimensional objects), audio support, saving games and preferences, input / output, the creation of the user interactive interface... Libgdx is closely related to OpenGL ES, but while it provides many useful abstractions for working with shaders, geometry and textures, it provides full access to the underlying architecture, therefore allowing a competent programmer to squeeze every last drop of performance from the limited hardware capabilities of modern phones and tablets.

The topics related to OpenGL will be also useful for those who want to **program Apple devices** or **create advanced visual effects in the browser using WebGL** in the future (WebGL is an official standard associated with HTML5, it is the version of GLES 2.0 designed for modern browsers).

During the course participants create the examples of simple games. To save time, the code implementing the logic of the game and the potential graphic/audio files are provided by the trainer.

Duration

2 days

Agenda

1. How OpenGL ES 2.0 works
 - Initialization of OpenGL context from Android application side, OpenGL and threads
 - Shaders: types, function, compilation
 - Introduction to GLSL language: constructs, types, constraints
 - Matrices, vectors, multiplication of vectors
 - Affine transformations and perspective
 - Textures, FrameBuffer objects
 - Basic graphics primitives: triangles, fans, lines
2. Typical 3D techniques for OpenGL ES 2.0
 - Ray-tracing (absolute basics)
 - Per-pixel-lighting
 - Gourad shading

Ask for details

Phone +44 203 608 6289

info@alx.academy

- Bump-mapping
 - Cube-mapping
3. Using LibGDX
- Organising an application, LibGDX architecture
 - 2D graphics: projection, cameras, texture regions, SpriteBatch, managed resources
 - Particle effects, blending
 - Designing game levels with a tile map editor (eg Tiled), loading and displaying game levels
 - Three dimensional graphics, loading and displaying ready-made objects
 - Scene graph, game's GUI (*retained mode* and *immediate mode*)
 - Truetype fonts
 - Tweening, different techniques and curves (on the example of Universal Tween Engine)
 - Sound, input, output, resources

Ask for details

Phone +44 203 608 6289

info@alx.academy

Target audience and prerequisites

The course is designed for people who:

- know Java language quite well
- understand basic concepts of geometry: two and three-dimensional vector, angle, rotation, the coordinate system

The course **is not recommended** to people who:

- already have some experience with OpenGL and Libgdx: for such people we would rather recommend advanced, individual consultations
- are programming beginners - computer games are surprisingly difficult software development field, so for the beginner the course will be too intensive.

Certificates

Course participants receive completion certificates signed by ALX.

Locations

- Warsaw (English) – Jasna 14/16A
- Online (English) – your home, office or wherever you want
- any other location (London, UK, EU) on request

Price

990 EUR

The price includes:

- course materials,
- snacks, coffee, tea and soft drinks,
- course completion certificate,
- one-time consultation with the instructor after course completion.