1. Graphics
2. Querying & configuring graphics (monitor, display modes, vsync)

LibGDX memiliki API yang rumit yang memungkinkan kita memantau dan menampilkan permintaan. Perubahan mode tampilan tidak didukung oleh Android dan iOS.

1. Continuous & Non-Continuous Rendering

Secara default di libgdx, thread rendering memanggil metode render() class ApplicationListener secara terus menerus, dengan frekuensi yang bergantung pada perangkat keras (30-50-80 kali per detik) . Jika rendering terus menerus disetel ke false, metode render() akan dipanggil hanya jika hal-hal berikut terjadi:

An input event is triggered

Gdx.graphics.requestRendering() is called

Gdx.app.postRunnable() is called

1. Clearing the Screen

Untuk menghapus layar di Libgdx tidak seperti membersihkan layar dalam aplikasi OpenGL biasa. Satu-satunya perbedaan adalah bagaimana seseorang mengakses konteks OpenGL

1. Taking a Screenshot

byte[] pixels = ScreenUtils.getFrameBufferPixels(0, 0, Gdx.graphics.getBackBufferWidth(), Gdx.graphics.getBackBufferHeight(), true);

Pixmap pixmap = new Pixmap(Gdx.graphics.getBackBufferWidth(), Gdx.graphics.getBackBufferHeight(), Pixmap.Format.RGBA8888);

BufferUtils.copy(pixels, 0, pixmap.getPixels(), pixels.length);

PixmapIO.writePNG(Gdx.files.external("mypixmap.png"), pixmap);

pixmap.dispose();

1. Profiling

Mengupdate panggilan OpenGL aktual yang terjadi saat permainan berjalan seringkali tidak terlalu mudah dilakukan, karena libGDX mencoba untuk menghapus semua hal tingkat rendah. Untuk memungkinkan pengumpulan informasi ini, ada GLProfiler

1. Viewports

Saat berhadapan dengan layar yang berbeda, seringkali perlu diputuskan untuk menentukan strategi tertentu bagaimana ukuran dan rasio aspek layar yang berbeda harus ditangani. Camera and Stage mendukung strategi viewport yang berbeda, misalnya saat melakukan pemotretan melalui Camera.project (vec, viewportX, viewportY, viewportWidth, viewportHeight).

1. OpenGL ES Support

Libgdx didikung dengan OpenGL ES 2.0 dan 3.0. Untuk beberapa platform seperti IOS, GWT/HTML hanya support pada OpenGL ES 2.0. Untuk perangkat android, diperlukan setting tertentu pada android manifest agar aplikasi pada play store tidak terlihat pada perangkat yang tidak mendukung OpenGL .

* Precision Modifiers

OpenGL ES 2.0 membutuhkan spesifikasi precision modifiers untuk atributes, uniforms dan locals. Oleh karena itu dibutuhkan settingan agar sesuai dengan OpenGL ES precision modifier. Desktop OpenGL tidak support untuk hal ini.

OpenGL ES 2.0 Documentation

Link untuk dokumentasi openGL sendiri telah disediakan pada dokumentasi ini.

1. Utility Class
   1. Rendering Shapes

Terdapat link dokumentasi khusus untuk mempelajari pembuatan berbagai bentuk.

* 1. Texture & Texture Regions

Menggunakan Spritebatch, Textureregions, and Sprites yang akan dibahas pada bab berikutnya

* 1. Meshes

Terdapat penjelasan tentang membuat dan merender mesh. Mesh sendiri adalah kumpulan verteks yang mendiskripsikan geometri untuk rendering

* 1. Shaders

Dalam OpenGL 2.0 ES semuanya dirender dengan shader. Shader merupakan program kecil yang berjalan pada GPU dan memproses data untuk dirender. Terdapat verteks shader(hanya mengeksekusi satu verteks) dan fragment shader(mengeksekusi setiap fragment). Selain itu, pada dokumentasi ini juga ada contoh sederhana program shader

* 1. Frame Buffer Objects

Terdapat link dokumentasi khusus untuk mempelajari framebuffer