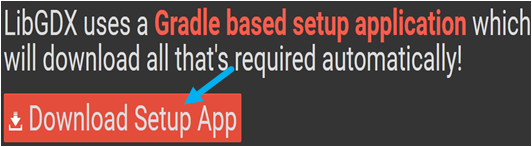
**Lập trình game android với libgdx**

libGDX là một framework được dùng để phát triển game cho nhiều nền tảng: Android, iOS, Desktop, Web bằng [ngôn ngữ lập trình Java](http://giasutinhoc.vn/lap-trinh-java-co-ban/). libGDX được phát triển bởi Mario Zechner. Thật thú vị và hữu ích khi sử dụng engine này để lập trình game cho mobile.

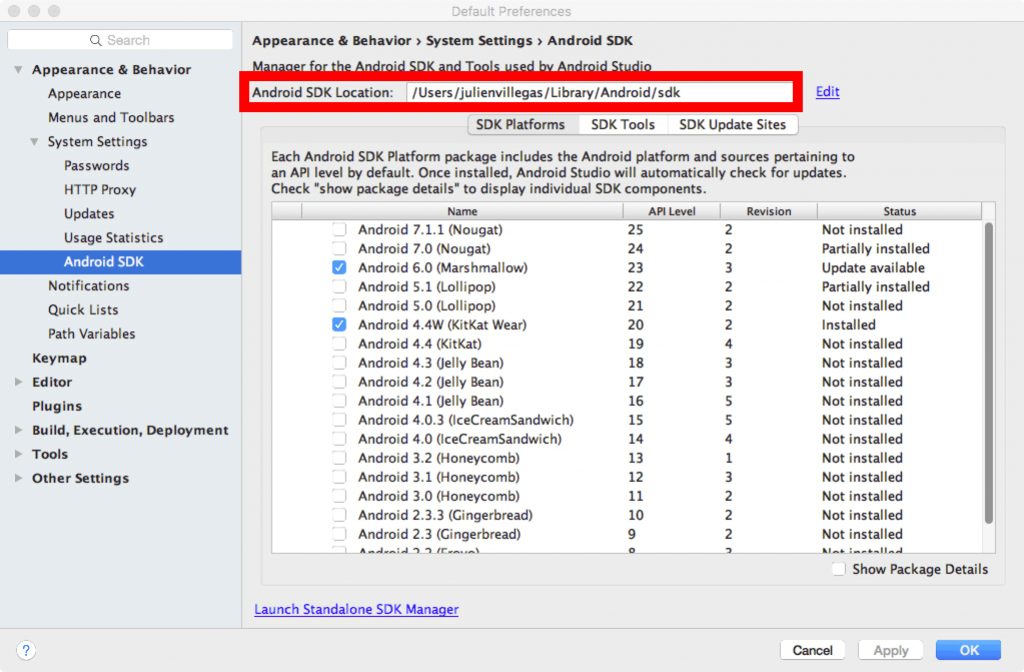
[](http://giasutinhoc.vn/wp-content/uploads/2016/12/lap-trinh-game-android-voi-libgdx-1.png)

**Lập trình game android với libgdx – Tạo project với libGDX**

**Bước 1.** Tải libGDX. Bạn đọc truy cập [libgdx.badlogicgames.com](https://libgdx.badlogicgames.com/download.html) -> chọn **Download Steup App**để tải libGDX về máy của mình.

[](http://giasutinhoc.vn/wp-content/uploads/2016/12/lap-trinh-game-android-voi-libgdx-2.png)

**Bước 2.** Chạy tập tin gdx-setup.jar (Lưu ý máy của bạn phải cài [JDK – Java Development Kit](http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html) và Android SDK. Bạn có thể tải và cài đặt [Android Studio bao gồm Android SDK](https://developer.android.com/studio/index.html)). Quan sát hình bên dưới để biết Android SDK path.

[](http://giasutinhoc.vn/wp-content/uploads/2016/12/lap-trinh-game-android-voi-libgdx-4.png)

**Các thiết lập cần thiết**

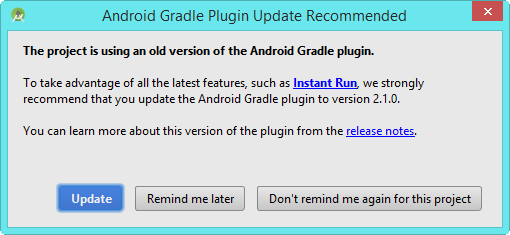
1. Name: Tên project
2. Package: Tên gói (thư mục con trong project)
3. Game class: Tên lớp
4. Destination: Thư mục chứa libGDX project sau khi tạo
5. Android SDK: Thư viện lập trình Android
6. Sub Projects: Lựa chọn nền tảng mà game hỗ trợ

Sau khi thiết lập những yêu cầu trên -> chọn **Generate ->**chọn**Yes**(libGDX tự động tải những thư viện cần thiết vì vậy yêu cầu máy tính phải kết nối internet).

[](http://giasutinhoc.vn/wp-content/uploads/2016/12/lap-trinh-game-android-voi-libgdx-3.png)

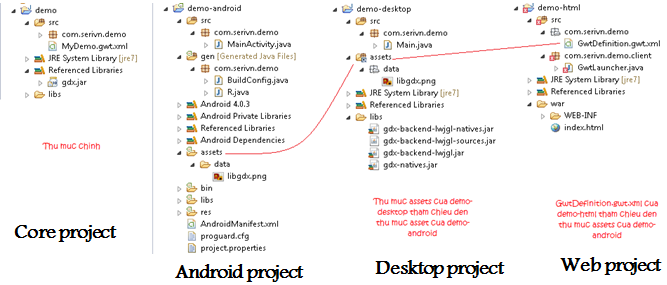
**Bước 3.** Import libGDX project vào Android Studio (Máy tính phải cài đặt Android Studio) theo trình thự sau:

Khởi động Android Studio **-> File ->** **New -> Import Project ->** chỉ định thư mục chứa libGDX project (mà bạn đã thực hiện ở bước 2) **->** chọn **OK** (Trong quá trình import cần kết nối mạng để tải các tập tin liên quan). Sau khi import nếu xuất hiện thông báo sau, vui lòng chọn **Update**

[](http://giasutinhoc.vn/wp-content/uploads/2016/12/lap-trinh-game-android-voi-libgdx-5.png)

**Lập trình game android với libgdx – Cấu trúc libGDX project**

Hình bên dưới thể hiện cấu trúc, ý nghĩa và sự liên kết giữa các project bên trong.

[](http://giasutinhoc.vn/wp-content/uploads/2016/12/lap-trinh-game-android-voi-libgdx-6.png)

Theo hình trên, chúng ta có 4 proeject. **Core project** là nơi lập trình viên viết code. **Android project** tương ứng cho nền tảng Android. Tài nguyên game sẽ chứa trong project này. **Desktop project** tương ứng cho nền tảng PC. **Web project** tương ứng cho nền tảng Web. Và nếu trong lúc tạo libGDX ở bước trên, chúng ta chọn Ios thì sẽ có thêm **iOS project** tương ứng cho nền tảng iOS.

**Lập trình game android với libgdx – Hệ thống trạng thái game trong libGDX**

Mỗi một trạng thái trong game tương ứng với một phương thức trong Java.

**public class MyGame implements ApplicationListener {**

**public void create () {**

**}**

**public void render () {**

**}**

**public void resize (int width, int height) {**

**}**

**public void pause () {**

**}**

**public void resume () {**

**}**

**public void dispose () {**

**}**

**}**

**Phương thức create()**: Phương thức này được gọi một lần khi application được tạo và dùng để khởi tạo trong game.

**Phương thức resize()**: Thực hiện thay đổi kích thước game theo thiết bị và được gọi sau phương thức create().

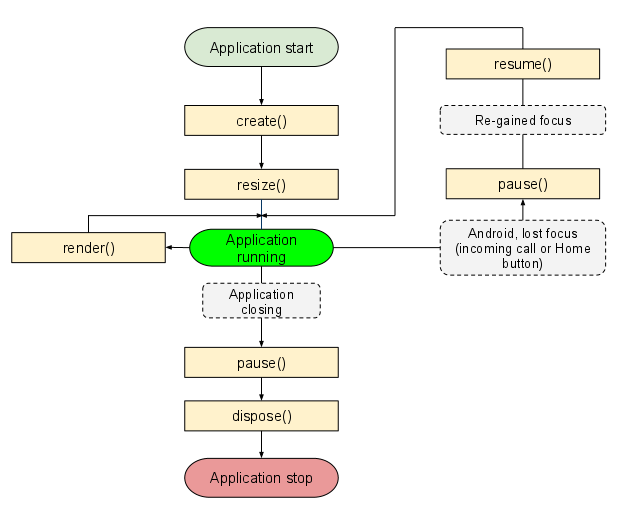
**Phương thức render()**: Có vai trò xử lý vòng lặp game (game loop)

**Phương thức pause()**: Đối với thiết bị di động, phương thức này được khi người dùng dùng nhấn phím Home hoặc khi có cuộc gọi đến. Đối với PC phương thức này được gọi trước phương thức dispose() khi người chơi thoát game.

**Phương thức resume()**: Phương thức này chỉ được gọi trên mobile, khi game chuyển từ trạng thái pause.

**Phương thức dispose()**: Được gọi khi kết thúc game.

Hình bên dưới thể hiện life cycle

[](http://giasutinhoc.vn/wp-content/uploads/2016/12/lap-trinh-game-android-voi-libgdx-7.png)

**Lập trình game android với libgdx – Vẽ hình trong libGDX**

**Khai báo các đối tượng SpriteBatch và Texture**

**private SpriteBatch sp;**

**private Texture texture;**

**Tạo đối tượng tại phương thức create()**

**sp = new SpriteBatch();**

**texture = new Texture(Gdx.files.internal("Your picture file name"));**

**Vẽ hình tại phương thức render()**

**sp.draw(texture, x, y);**

**Ví dụ về xử lý vẽ hình trong libGDX**

**/\*\***

**\* Created by giasutinhoc.vn**

**\*/**

**public class ImageExample extends ApplicationAdapter{**

**SpriteBatch sb;**

**Texture texture;**

**float y ;**

**@Override**

**public void create() {**

**sb = new SpriteBatch();**

**texture = new Texture(Gdx.files.internal("jet.png"));**

**y = 0;**

**}**

**@Override**

**public void render() {**

**Gdx.gl.glClearColor(1, 1, 1, 1);**

**Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);**

**sp.begin();**

**sp.draw(texture, 10, 100);**

**sp.end();**

**}**

**@Override**

**public void dispose() {**

**texture.dispose();**

**}**

**}**

**Lập trình game android với libgdx – Vẽ chuỗi trong libGDX**

**Khai báo các đối tượng SpriteBatch và BitmapFont**

**private SpriteBatch sp;**

**private BitmapFont bf;**

**Tạo đối tượng tại phương thức create()**

**sp = new SpriteBatch();**

**bf = new BitmapFont();**

**Vẽ chuỗi tại phương thức render()**

**sp.begin();**

**bf.draw(sp, "Your text", x, y);**

**sp.end();**

**Một số thiết lập thường gặp khi vẽ chuỗi trong libGDX**

1/ Thiết lập màu chữ (color)

**bf.setColor(Color.ColorName);**

2/ Thiết lập cỡ chữ (size)

**bf.getData().setScale(nf,mf);**

**// Với n và m là những con số**

Ví dụ về thiết lập cỡ chữ trong libGDX

**bf.getData().setScale(2f,2f);**

3/ Thiết lập màu nền (Background color)

**Gdx.gl.glClearColor(Red/255, Green/255, Blue/255, 1);**

**Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);**

Ví dụ thiết lập màu nền trắng trong libGDX. Lưu ý màu trắng thì Red = 255, Green = 255, Blue = 255 => Red/255 = 1, Green/255 = 1 và Blue/255 = 1.

**Gdx.gl.glClearColor(1, 1, 1, 1)**

**Ví dụ về xử lý vẽ chuỗi trong libGDX**

**/\*\***

**\* Created by giasutinhoc.vn**

**\*/**

**public class StringExample extends ApplicationAdapter {**

**SpriteBatch sb;**

**BitmapFont bf;**

**String msg = "Game android voi libGDX";**

**@Override**

**public void create() {**

**sb = new SpriteBatch();**

**bf = new BitmapFont();**

**}**

**@Override**

**public void render() {**

**Gdx.gl.glClearColor(1, 1, 1, 1);**

**Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);**

**sb.begin();**

**bf.setColor(Color.BLUE);**

**bf.getData().setScale(3f);**

**bf.draw(spriteBatch, msg, 100, 100);**

**sb.end();**

**}**

**@Override**

**public void dispose() {**

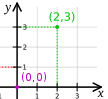
**super.dispose();**

**}**

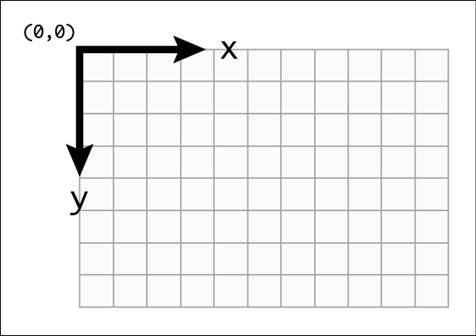
**}**

**Lập trình game android với libgdx – Hệ toạ độ trong libGDX**

Mặc định góc toạ độ là góc dưới bên trái. X sẽ tăng theo chiều từ trái qua phải, Y sẽ tăng theo chiều từ dưới lên.

[](http://giasutinhoc.vn/wp-content/uploads/2016/12/lap-trinh-game-android-voi-libgdx-8.png)

**Xử lý thay đổi góc toạ độ trong libGDX**

[](http://giasutinhoc.vn/wp-content/uploads/2016/12/lap-trinh-game-android-voi-libgdx-9.jpg)

1/ Khai báo đối tượng **OrthographicCamera**

**private OrthographicCamera** **camera;**

2/ Tạo đối tượng **OrthographicCamera**và thiết lập chiều rộng, chiều cao của thiết bị tại phương thức **create()**

**// Chiều rộng thiết bị**

**int w = Gdx.graphics.getWidth();**

**//** **Chiều cao thiết bị**

**int h = Gdx.graphics.getHeight();**

**// Tạo đối tượng**

**camera = new OrthographicCamera(w,h);**

**camera.setToOrtho(true);**

3/ Chỉ định đối tượng **OrthographicCamera**tại phương thức **render()**

**batch.setProjectionMatrix(camera.combined);**

**Lập trình game android với libgdx – Vẽ Shape trong libGDX**

1/ Khai báo đối tượng **ShapeRenderer**

**private ShapeRenderer sr;**

2/ Tạo đối tượng **ShapeRenderer**tại phương thức **create()**

**sr = new ShapeRenderer();**

3/ Vẽ shape tại phương thức **render()**

3.1/ Thiết lập shape type

**sr.begin(ShapeRenderer.ShapeType.TypeName);**

3.2/ Thiết lập màu

**sr.setColor(Red/255, Green/255, Blue/255, 1);**

3.3/ Vẽ

**sr.MeThodName();**

Tham khảo các [phương thức vẽ shape trong libGDX](https://libgdx.badlogicgames.com/nightlies/docs/api/com/badlogic/gdx/graphics/glutils/ShapeRenderer.html)

**Ví dụ về xử lý vẽ shape trong libGDX**

**/\*\***

**\* Created by giasutinhoc.vn**

**\*/**

**public class ShapeExample extends ApplicationAdapter{**

**ShapeRenderer sr;**

**@Override**

**public void create() {**

**sr = new ShapeRenderer();**

**}**

**@Override**

**public void render() {**

**sr.begin(ShapeRenderer.ShapeType.Line);**

**//Color**

**sr.setColor(1, 1, 1, 1);**

**//Draw rectange**

**sr.rect(10, 10, 200, 200);**

**//Draw Circle**

**sr.circle(300, 300, 20);**

**//Draw line**

**sr.setAutoShapeType(true);**

**sr.set(ShapeRenderer.ShapeType.Line);**

**sr.line(100, 310, 300, 310);**

**sr.end();**

**}**

**@Override**

**public void dispose() {**

**sr.dispose();**

**}**

**}**