ANDROID GAME TUTORIALS

# Giới thiệu

## Giới thiệu game

🡺Mario Bros

## Công cụ sử dụng

### Platform: Android Studio

Để phát triển các ứng dụng mobile chạy trên hệ điều hành Android thì bạn cần một máy tính trên đó có cài đặt Android Studio. Trong bài viết này chúng ta sẽ cùng nhau tìm hiểu về Android Studio, cách cài đặt và sử dụng Android Studio để tạo một ứng dụng đơn giản có tên là Hello World. Bài viết này áp dụng trên hệ điều hành Windows do đó bạn cần một máy tính chạy hệ điều hành Windows.



Android Studio Là Gì

**Android Studio** là một phầm mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập simulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

Việc xây dựng một ứng dụng mobile (hoặc tablet) bao gồm rất nhiều các công đoạn khác nhau. Đầu tiên chúng ta sẽ viết code ứng dụng sử dụng máy tính cá nhân hoặc laptop. Sau đó chúng ta cần build ứng dụng để tạo file cài đặt. Sau khi build xong thì chúng ta cần copy file cài đặt này vào thiét bị mobile (hoặc table) để tiến hành cài đặt ứng dụng và chạy kiểm thử (testing). Bạn thử tưởng tượng nếu với mỗi lần viết một dòng code bạn lại phải build ứng dụng, cài đặt trên điện thoại hoặc tablet và sau đó chạy thử thì sẽ vô cùng tốn thời gian và công sức. Android Studio được phát triển để giải quyết các vấn đề này. Với Android Studio tất cả các công đoạn trên được thực hiện trên cùng một máy tính và các quy trình được tinh gọn tới mức tối giản nhất.

Tiếp theo chúng ta sẽ tìm hiểu cách cài đặt Android Studio trên Windows, Mac OSX và Linux.

### Framework: LibGDX

**Giới thiệu về LibGDX**

LibGDX là một framework phát triển ứng dụng game, được viết bằng ngôn ngữ lập trình Java, một số các thành phần được viết bằng C và C++ để có hiệu năng tốt hơn. Nó cho phép phát triển các ứng dụng desktop và mobile trên cùng một code base. LibGDX chạy được đa nền tảng, hỗ trợ Windows, Linux, Mac OS X, Android, iOS, và trên web browsers với WebGL.

Một số ưu điểm của LibGDX:

1. Đa nền tảng: chúng ta chỉ cần viết code 1 lần nhưng có thể chạy trên được nhiều nền tảng khác nhau. Một ứng dụng của tính năng này đó là phát triển các ứng dụng cho Android. Để phát triển một ứng dụng cho Android, khi chạy thử ứng dụng, chúng ta cần chạy ứng dụng trên Emulator hoặc trên thiết bị thật. Việc này rất mất thời gian do Emulator chạy rất chậm và quá trình cài đặt cũng như chạy ứng dụng trên thiết bị thật cũng không khá hơn. Với LibGDX, chúng ta có thể chạy ứng dụng trên PC, sau đó chỉ cần với vài dòng code, chúng ta có thể chạy ứng dụng này trên Android với hiệu năng tương đương. Điều này giúp chúng ta kiểm thử và tìm lỗi ứng dụng nhanh hơn và hiệu quả hơn.
2. Hiệu năng: Hiệu năng của LibGDX thực sự rất ấn tượng do LibGDX do các thành phần được viết bằng C và C++.
3. Cộng đồng: cộng đồng sử dụng LibGDX rất tuyệt vời với số lượng người dùng lớn. Các lập trình viên luôn đóng góp và giúp đỡ cho cộng đồng. Việc sửa lỗi cũng được cập nhật rất thường xuyên.
4. Tài liệu và ví dụ: rất đầy đủ với Javadoc. LibGDX cũng cung cấp rất nhiều các ví dụ với đầy đủ các chức năng từ đơn giản đến phức tạp.
5. Mã nguồn: mã nguồn mở với thiết kế rất rõ ràng và phù hợp với việc phát triển ứng dụng cho di động. LibGDX cho phép người lập trình khả năng sử dụng các API từ các lớp thấp đến cao, tùy theo yêu cầu của người sử dụng.
6. Tính năng: LibGDX có rất nhiều tính năng như tạo hình, xử lý đồ họa 2D, 3D, xử lý âm thanh, quản lý các thiết bị vào ra, quản lý file hệ thống. Cùng với đó là các công cụ đi kèm rất hữu ích như Texture Packer và Particle Editor.

### Các công cụ khác

Github

1. Cấu trúc ứng dụng chạy trên nền tảng android

Vòng đời của 1 Activity

Các Activity trong hệ thống được quản lý như 1 ngăn xếp activity (activity stack). Khi 1 activity mới bắt đầu nó được đặt lên đầu của ngăn xếp và trở thành Running Activity (activity đang chạy), đồng thời activity trước đó sẽ nằm ngay phía dưới trong ngăn xếp đó, và sẽ không trở nên visible (nhìn thấy) cho đến khi activity ở trên thoát ra khỏi ngăn xếp.

Một Activity gồm 4 trạng thái chính:

* Nếu activity ở phía trên của màn hình (hay ở trên cùng của ngăn xếp), thì nó đang ở trạng thái **active** (hoạt động) / **running** (đang chạy). Ví dụ khi ta cần gọi điện thì activity bấm số đó đang ở trạng thái active.
* Nếu activity không thể tương tác nhưng vẫn nhìn thấy ( khi mà bị che bởi 1 activity khác nhưng người dùng vẫn có thể nhìn thấy nó ở phía sau ) thì activity này đang ở trạng thái **paused** (tạm dừng). Khi ở trạng thái này activity có thể bị xóa bỏ bởi hệ thống khi thiết bị thiếu bộ nhớ. Ví dụ khi có 1 activity khác dạng dialog hiện lên chỉ che đi 1 phần của activity hiện tại thì activity vào trạng thái paused.
* Nếu activity hoàn toàn bị che khuất bởi activity khác thì nó đang ở trạng thái **stopped** (đã dừng). Activity này vẫn giữ được tất cả trạng thái và thông tin, nhưng không còn hiển thị với người dùng và thường xuyên bị xóa bỏ bởi hệ thống khi thiếu bộ nhớ. Ví dụ khi ta tắt màn hình thì khi đó activity vào trạng thái stopped.
* Nếu activity ở trạng thái **paused** (tạm dừng) hay **stopped** (đã dừng), hệ thống có thể xóa bỏ activity đó khỏi bộ nhớ bằng cách yêu cầu nó tự kết thúc hoặc xóa bỏ tiến trình của nó. Khi acitivty đó hiển thị lại với người dùng thì sẽ được khởi tạo lại và khôi phục lại trạng thái trước đó.

Hình ảnh sau đây minh họa cho vòng đời của 1 Activity cùng với các trạng thái của nó:



# Game Framework

## Aspect Ratios(tỉ lệ khung hình)

Để game có thể chạy đúng tỉ lệ trên một màn hình lớp, hay một màn hình có kích thước nhỏ hơn

## ViewPort

ViewPort sẽ được xây dựng dựa vào kích thước của màn hình để tránh trường hợp device có màn hình lớp có thể nhìn được nhiều hơn trong thế giới game => điều đó không công bằng… Vì vậy Viewport sẽ được quy ước 1 đơn vị worldgame sẽ bằng 1 pixel, tỉ lệ đó có thể thay đổi

## TileMap