MỤC LỤC

DANH SÁCH CÁC HÌNH

DANH SÁCH CÁC TỪ VIẾT TẮT

Stt	Từ viết tắt	Mô tả
1	FPGA	Field programmable Gate Array
2	SoC	System on a Chip
3	GPU	Graphics Processing Unit
4		
5		

LỜI CẢM ƠN

Trước hết, em xin chân thành cảm ơn thầy giáo - Ths. Nguyễn Trịnh Đông, giảng viên Khoa Công nghệ thông tin - Trường Đại học Dân Lập Hải Phòng, người đã dành cho em rất nhiều thời gian quý báu, trực tiếp hướng dẫn tận tình giúp đỡ, chỉ bảo em trong suốt quá trình làm đồ án tốt nghiệp.

Em xin chân thành cảm ơn tất cả các thầy cô giáo trong khoa Công nghệ thông tin - Trường Đại học Dân Lập Hải Phòng, chân thành cảm ơn các thầy giáo, cô giáo tham gia giảng dạy và truyền đạt những kiến thức quý báu trong suốt thời gian em học tập tại trường, đã đọc và phản biện đồ án của em giúp em hiểu rõ hơn các vấn đề mình nghiên cứu, để em có thể hoàn thành đồ án này.

Em xin chân thành cảm ơn GS.TS.NGƯT Trần Hữu Nghị, Hiệu trưởng Trường Đại học Dân Lập Hải Phòng, ban giám hiệu nhà trường, khoa Công nghệ thông tin, các phòng ban nhà trường đã tạo điều kiện tốt nhất trong suốt thời gian em học tập và làm tốt nghiệp.

Tuy có nhiều cố gắng trong quá trình học tập và làm đồ án tốt nghiệp nhưng không thể tránh khỏi những thiếu sót nhất định, em rất mong được sự góp ý quý báu của tất cả các thầy cô giáo cũng như tất cả các bạn để đồ án của em ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

Hải Phòng, ngày 5 tháng 07 năm 2014

Sinh viên

Lê Vũ Minh Quang

GIỚI THIỆU

Lập trình trên thiết bị di động đang là xu hướng phát triển ngày nay của ngành truyền thông và công nghệ thông tin. Dưới góc độ kinh tế đây là một ngành đem lại nhiều lợi nhuận cho nền kinh tế. Thứ nhất, nó mở ra một hướng mới cần nhiều lao động kỹ thuật cao, giải quyết nhiều công việc cho người lao động. Thứ 2, ngành này thúc đây nhiều ngành công nghiệp khác phát triển theo như viễn thông, thương mại điện tử, giáo dục và một số ngành dịch vụ khác. Thứ 3, ưu điểm của ngành này là ngành công nghiệp không khói, nguyên liệu chính là tri thức và đặc biệt đem lại lợi ích to lớn cho cộng đồng.

Từ các ưu điểm trên em đã thực hiện đề tài:***"Lập trình game trên thiết bị di động".*** Khóa luận này cho em một hướng đi mới trong việc định hướng nghề nghiệp, cũng như phát triển thêm về kỹ năng lập trình và phát triển hệ thống.

Khóa luận này được trình bày theo cấu trúc sau:

Giới thiệu

Chương 1:*Cơ sở lý thuyết, chương này trình bày các kiến thức cơ bản về thiết bị di động như kiến trúc phần cứng, vi xử lý, bộ nhớ, các thiết bị đo lường và nền tảng của hệ điều hành Android.*

Chương 2: *Lập trình trò chơi trên Android, trong phần này các thành phần liên quan đến một ứng dụng trò chơi được đề cập.*

Chương 3:*Giới thiệu về ứng dụng trò chơi.*

Kết luận

Tài liệu tham khảo

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

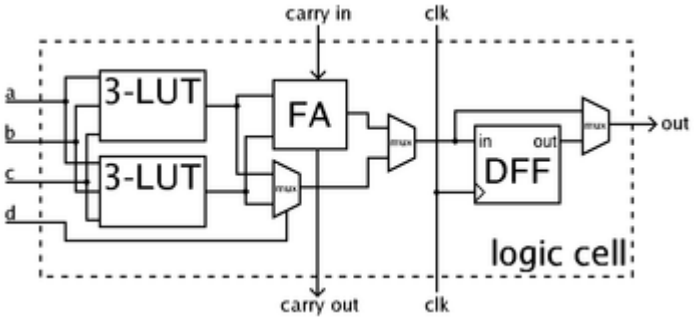
Giới thiệu

Lập trình là một lĩnh vực quan trọng trong ngành công nghệ thông tin, lập trình cho thiết bị di động đòi hỏi một số yêu cầu riêng khác với lập trình cho máy tính. Lập trình cho máy tính nói chung được phát triển ổn định trong một thời gian dài và có ít thay đổi về kiến trúc hệ thống cũng như nền tảng công nghệ. Trong khi đó, thiết bị di động dù rất phổ biến hiện nay nhưng có thời gian phát triển tương đối ngắn, công nghệ đang thay đổi, không có chuẩn thống nhất, phần cứng phụ thuộc nhiều vào các hãng khác nhau, hệ điều hành rất đa dạng do vậy ngôn ngữ lập trình cũng rất đa dạng. Tuy nhiên các sản phẩm về thiết bị di động đều có một số chức năng chính như màn hình đa chạm, CPU đa lõi, bộ nhớ RAM đủ lớn, thẻ nhớ, ổ cứng SSD. Đặc biệt có một số thiết bị mà ở các hệ thống máy tính chưa có như: Gia tốc kế, Con quay hồi chuyển, Định vị vệ tinh, máy đo từ trường, và một số các sensor khác. Từ phân tích trên, trong chương này khóa luận tập trung vào trình bày nền tảng phần cứng của thiết bị di động và một số ngôn ngữ lập trình điển hình cho thiết bị di động.

1.1 KIẾN TRÚC CƠ BẢN CỦA THIẾT BỊ DI ĐỘNG THÔNG MINH

1.1.1. Giới thiệu

Hiện nay có rất nhiều các hãng sản xuất thiết bị phần cứng cho điện thoại thông minh như Apple, Samsung, LG, Nokia, …tuy nhiên nền tảng công nghệ cũng giống nhau đó là sử dụng SoC (System on a chip).Công nghệ này tích hợp nhiều bộ phận khác nhau vào trong một vi mạch tích hợp như: Vi xử lý (CPU), chip xử lý đồ họa (GPU), RAM, ROM, trình điều khiển USB và các vi mạch WIFI cùng nhiều thứ khác nữa.Các hệ thống SoC điều dựa trên nền tảng công nghệ FPGA (**field-programmable gate array**)



Hình 1-1:Kiến trúc cơ bản của FPGA

Hầu hết kiến trúc chung của FPGA bao gồm một mảng các khối logic, dãy cổng vào ra, và các kênh định tuyến,các kênh định tuyến có cùng độ rộng (về mặt vật lý), còn dãy cổng vào ra phù hợp với chiều cao hoặc độ rộng của mảng [WikiFPGA].

Sử dụng FPGA là một giải pháp khác nhằm khắc phục một số hạn chế của ASIC (Application-Specific Integrated Circuit), ASIC là một vi mạch IC được thiết kế dành cho một ứng dụng cụ thể, về vấn đề giá thành và độ phức tạp khi triển khai.FPGA là một loại vi mạch bán dẫn chuyên dụng ASIC, nhưng nếu so sánh FPGA với ASIC thì FPGA không đạt được mức độ tối ưu và có hạn chế trong việc thực hiện những tác vụ đặc biệt phức tạp, tuy nhiên FPGA ưu việt hơn trong khi triển khai như có thể tái lập trình lại, thiết kế đơn giản đẫn đến chi phí giảm, rút ngắn thời gian đưa sản phẩm vào thực tế.

Ngoài ra còn có một số vi mạch bán dẫn lập trình đượcnhư PLA, PAL, CPLD cũng dùng cấu trúc mảng phần tử logic nhưngFPGA ưu việt hơn

- Lập trình của FPGA thực hiện đơn giản hơn
- Cho phép nạp lại chương trình

Các nhà thiết kế sử dụng các ngôn ngữ mô tả phần cứng như VHDL, Verilog, AHDL để thiết kế và lập trình cho FPGA. Các gói phần mềm và thiết bị phụ trợ cho quá trình thiết kế này do các hãng sản xuất FPGA lớn như Xilinx, Altera cung cấp. Ngoài ra cũng có một số hãng khác cung cấp các gói phần mềm kiểu này như Synopsys, Synplify...

1.1.2. Vi xử lý

Vi xử lý cho điện thoại thông minh được nhiều hãng sản xuất và theo nhiều chuẩn khác nhau. Trong phần này khóa luận giới thiệu một số CPU dùng cho điện thoại thông minh.

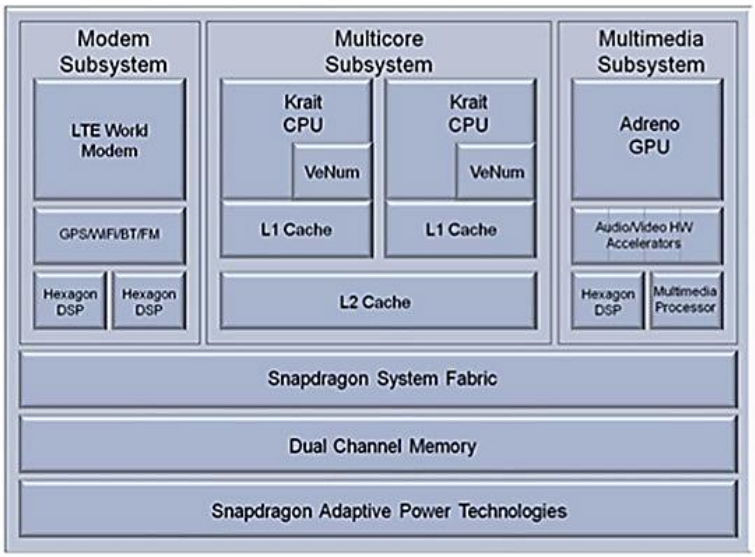
1.1.2.1 ARM

Chip ARM (Advance RISC Machine) của hãng ARM Holdings được nâng cấp qua nhiều phiên bản. Hiện nay có nhiều phiên bản của ARM như ARM 11, Cortex-M, Cortex-R,Cortex-A,… Thực chất ARM Holdings không sản xuất chip mà họ thiết kế ra các loại chip rồi cấp phép cho các công ty khác như *Qualcomm, Texas Instruments, Apple, Samsung,…* ARM cung cấp các thiết kế IP-Core cho loại 32 bit và 64 bit.

***1.1.2.2** SoC Snapdragon*

*SoC Snapdragon*của hãng *Qualcomm*đã lựa chọn một giải pháp khác mặc dù họ cũng sử dụng bản quyền của ARM để sản xuất nhưng họ không sử dụng hoàn toàn thiết kế của ARM. Hãng này dựa vào thiết kế ARM rồi cải tiến thiết kế để phát triển thành các vi xử lý *Scorpion* và *Krait* riêng của họ. Do đó, các vi xử lý của *Qualcomm* có tốc độ xử lý liên quan đến đa phương tiện tốt hơn và tiêu thụ điện hiệu quả hơn so với các bộ vi xử lýban đầu.Các vi xử lý *Scorpion* và *Krait* được đưa vào các *SoC Snapdragon của Qualcomm*. Mã *Snapdragon* có nhiều loại được đánh số S_x ($x \in$ 1, 2,.., n). Số *x* càng lớn thì tốc độ xử lý càng mạnh. Hiện tại có một số sản phẩm dự kiến sẽ

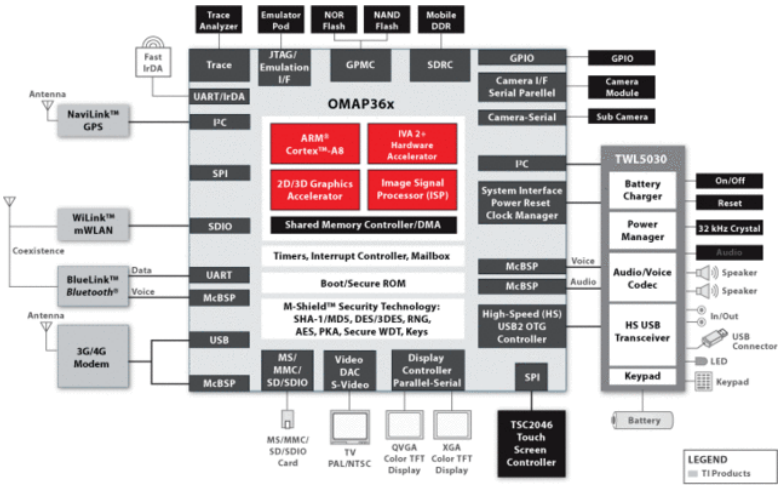
sử dụng *Snapdragon* S4 của *Qualcomm* như *HTC One S, HTC EVO 4G LTE, HTC One XL,…*



Hình 1-2:Kiến trúc Snapdragon S4 sử dụng bộ vi xử lý Krait

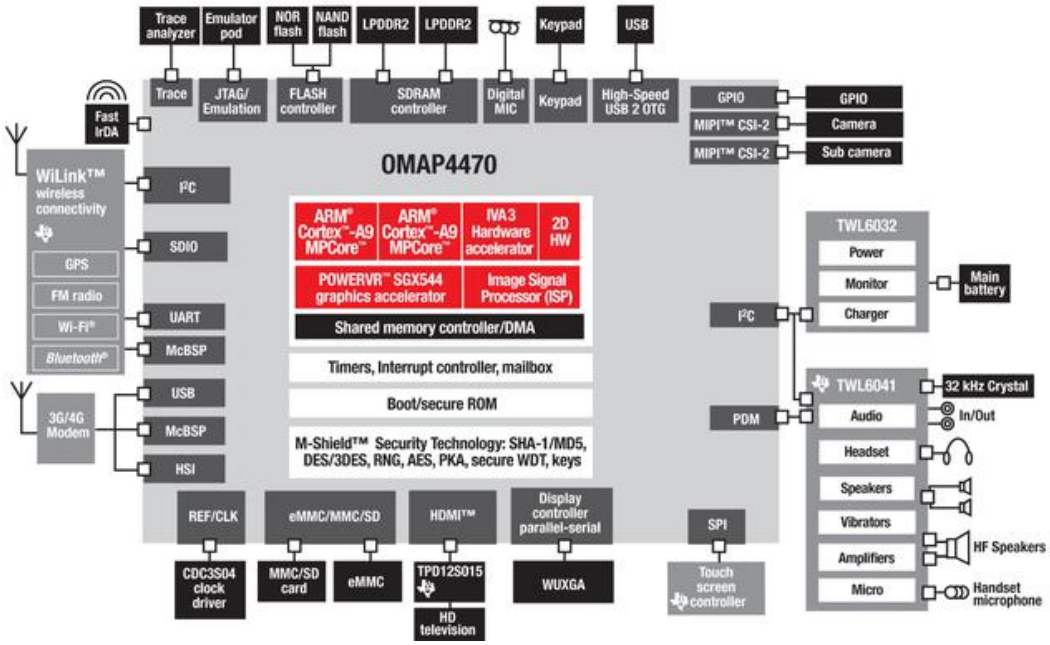
1.1.2.3 *OMAP*

SoC OMAP (Open Media Application Platform) của *Texas Instruments* được sử dụng khá rộng rãi. Trong đó các sản phẩm điện thoại của Motorola đều sử dụng OMAP. OMAP có nhiều dòng sản phẩm như OMAP1, 2, 3, 4,... Nhưng dòng sản phẩm OMAP 3 và 4 được dùng phổ biến hiện nay.



Hình 1-3: Sơ đồ khối SoC OMAP36xx của Texas Instruments

Do mục đích tạo nhiều sản phẩm với các tùy chọn nên *SoC OMAP* không tích hợp bộ phận *Wifi, Bluetooth*và một số thành phần khác, mà để cho từng dòng sản phẩm có sự linh động nhưng lại làm tăng kích thước của vỉ mạch chính.

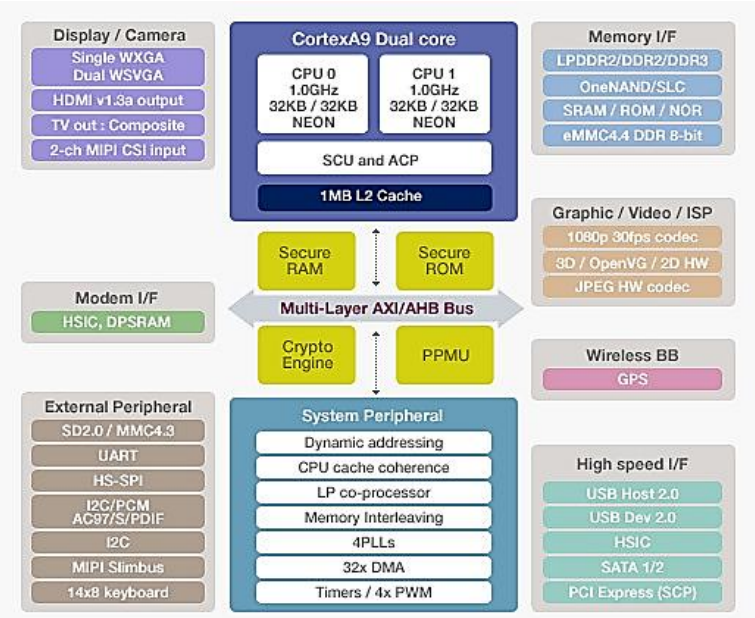


Hình 1-4: Biểu đồ OMAP4470 của Texas Instruments

Tuy nhiên, các SoC OMAP được tích hợp thêm công nghệ tiết kiệm điện *SmartReflex*và tăng cáchai lõi ARM Cortex-M_xđể tiết kiệm điện và tăng thời gian pin.

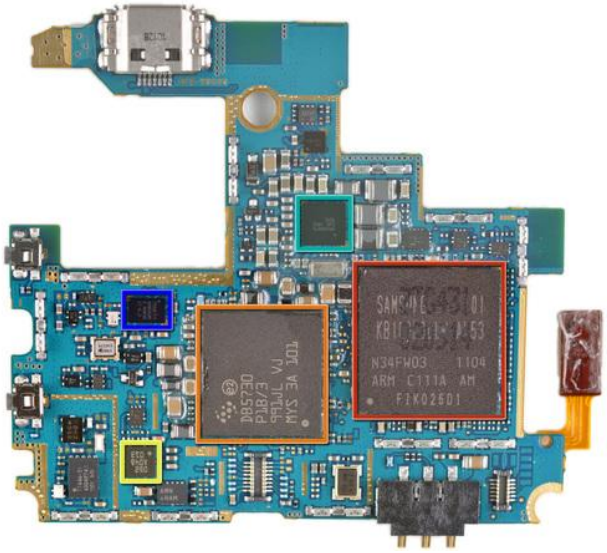
1.1.2.4 d. Exynos

*SoC Exynos*của hãng *Samsung*chủ yếu phục vụ trong các sản phẩm của *Samsung*, ngoài ra hãng Meizu, nhà sản xuất điện thoại Trung Quốc sử dụng cho điên thoại của họ. Mặc dù Samsung sản xuất *SoC* nhưng họ vẫn nhập *Snapdragon* của *Qualcomm*để sản xuất điện thoại vì *Exynos*chưa đáp ứng được một số yêu cầu của sản phẩm mới.



Hình 1-5:Sơ đồ khối Exynos 4210 của Samsung

Dòng sản phẩm *Exynos* 4210 lõi kép được cải thiện mạnh về tốc độ so với phiên bản trước, và còn được tích hợp các chức năng như GPS, HDMI và USB Host.Dưới đây là vỉ mạch sử dụng họ *Exynos*

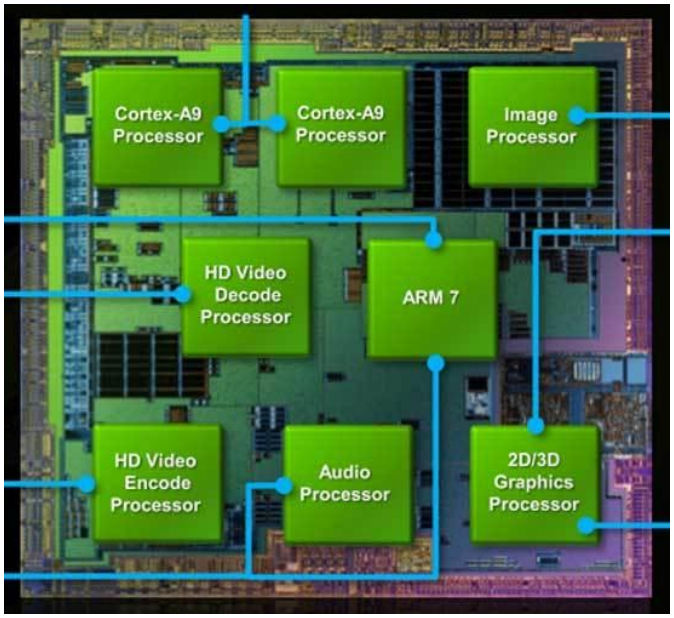


Hình 1-6: Hình minh họa vỉ mạch Samsung Galaxy S 4G

Hiện tại Exynos chưa thực sự chiếm ưu thế trên thị trường nhưng tương lai của Exynos rất hứa hẹn. Samsung đang thử nghiệm các thế hệ Exynos mới được tích hợp các bộ vi xử lý ARM Cortex-A15 lõi kép 2 GHz cùng với vi xử lý đồ họa Mali cải tiến, hỗ trợ 3D, màn hình độ phân giải 2560 x 1600 (WQXGA) cũng như chất lượng *camera* cũng được cải thiện.

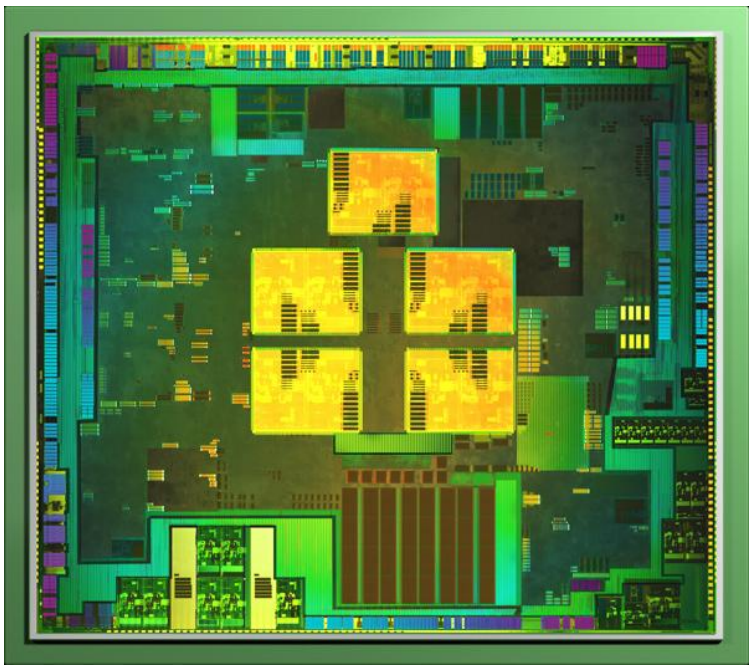
1.1.2.5 e. Tegra

Các loại SoC Tegra của *Nvidia*đang sử dụng trong các thiết bị di động là dòng Tegra 2 hoặc Tegra 3 với tên mã là Kal-El. Các dòng sản phẩm này đều tích hợp vi xử lý đa lõi ARM Cortex-A9 tốc độ từ 1 GHz đến 1.4 GHz; sử dụng GPU tiết kiệm điện GeForce làm chip xử lý đồ họa.



Hình 1-7: Sơ đồ khối của Tegra 2

Tegra 3 được cải tiến và là sản phẩm SoC bốn lõi đầu tiên trên thế giới dành cho thiết bị di động.Tốc độ xử lý của các lõi A9 đã được tăng từ 1.2 GHz lên 1.3 GHz ở cấu hình 4 lõi và GPU cũng được tăng đáng kể. Tegra 3 có thể chạy trên màn hình độ phân giải lên tới 2048 x 1530 pixel nhưng chỉ có thể quản lý hai màn hình đồng thời trong khi các chip mới của *OMAP* và *Exynos* có thể quản lý tới 3 hoặc 4 màn hình cùng lúc.



Hình 1-8: Hình ảnh của Tegra 3. Năm lõi vi xử lý (lõi thứ 5 nằm ở trên cùng)

*Nvidia*dùng lõi Cortex-A9 thứ nămvới ttốc độ 500 MHz để xử lý những tác vụ đơn giản nhằm tiết kiệm năng lượng. Trong tương lai sản phẩm có tên là *Wayne* sẽ

tích hợp các bộ vi xử lý ARM Cortex-A15 trong cấu hình bốn lõi hoặc tám lõi. GPU GeForce cũng sẽ được tăng lên.

1.1.2.6 f. SoC của Apple

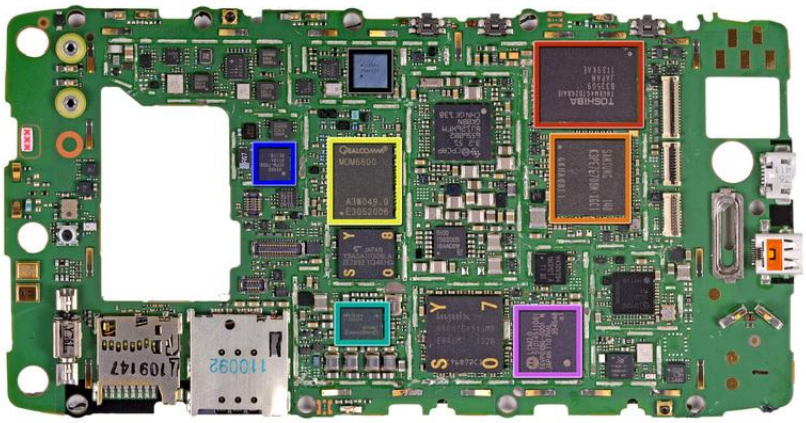
Apple là một hãng nổi tiếng với các sản phẩm tinh tế và chất lượng cao. Họ cũng sản xuất SoC nhưng không chuyển giao cho hãng khác. Apple chỉ dùng cho các sản phẩm của họ. Hiện nay Apple lấy nguồn SoC trực tiếp từ Samsung trang bị cho máy tính bảng và điện thoại Iphone của họ.

Chip A4 là SoC tích hợp vi xử lý ARM Cortex-A8 lõi đơn tốc độ từ 800 MHz đến 1Ghz và GPU PowerVR SGX535. Nó được sản xuất trên quy trình 45nm.

Apple cải thiện Chip A5 với việc tích hợp vi xử lý ARM Cortex-A9 lõi kép và GPU PowerVR SGX543MP2 lõi kép cùng với bộ nhớ RAM 512 MB mạnh hơn A4 dùng RAM 256 MB còn các thành phần khác vẫn tương tự A4 như tốc độ xung nhịp, quy trình 45nm, bộ vi xử lý tín hiệu hình ảnh cũng như công nghệ "earSmart" để khử nhiễu âm thanh.Trong thế hệ kế tiếp SoC A6 sẽ sử dụng quy trình 28nm và tích hợp nhiều lõi với tốc độ vi xử lý cao hơn cùng khả năng tích hợp Cortex-A15.

1.1.3. Bộ nhớ RAM

Bộ nhớ RAM trong thiết bị di động cũng có vai trò tương tự như trong máy tính cá nhân. Công nghệ sản xuất và nguyên lý hoạt động cũng giống nhau do đó trên thiết bị di động có 2 thông số cần quan tâm là: Dung lượng RAM và tần số xung nhịp. RAM được tích hợp trong SoC nhằm giảm thiểu kích thước thiết bị, giảm điện năng tiêu thụ, tốc độ truy cập nhanh hơn.



Hình 1-9: Hình minh họa vỉ mạch Motorola Droid Razr

1.1.4. Bộ nhớ ROM

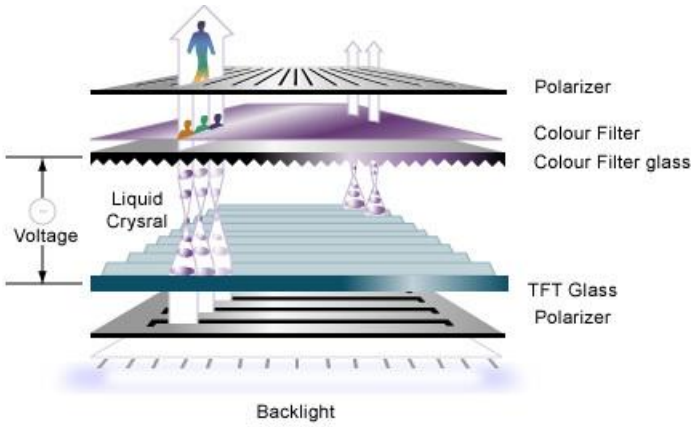
Bộ nhớ ROM trong thiết bị di động có vai trò to lớn hơn so với ROM trong máy tính cá nhân. ROM được tích hợp trong SoC và được phân chia thành nhiều vùng. Vùng chứa thông tin hệ thống, vùng chứa hệ điều hành và chứa các tệp tin hệ thống khác.

1.1.5. Màn hình

Màn hình điện thoại thông minh dựa trên hai công nghệ chính là LCD và LED

1.1.5.1 Màn hình LCD

LCD (Liquid Crytal Display) thường có các lớp sau: lớp bảo vệ bên ngoài, lớp phân cực, lớp tinh thể lỏng và đèn nền.



Hình 1-10: Sơ đồ đơn giản của một tấm nền LCD TFT

Ưu điểm:

- Giá thành rẻ
- Màn hình IPS LCD có khả năng tái tạo màu sắc chính xác.
- Ít bị hiện tượng biến đổi màu
- Có thể đạt độ sáng cao giúp dễ nhìn khi xem ngoài trời

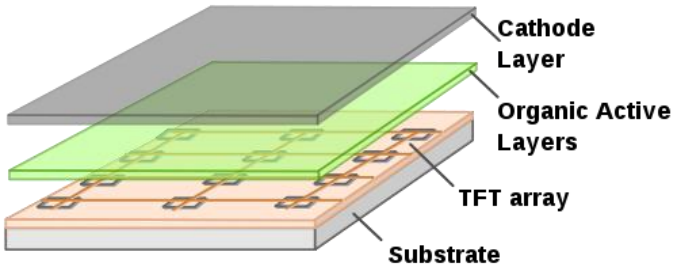
Nhược điểm:

- Do cần có đèn nền nên màn LCD khó đạt được tỉ lệ tương phản cao và màu đen tuyệt đối
- Màn hình TN LCD có góc nhìn kém
- Trong một số trường hợp, màn hình LCD tiêu tốn nhiều điện năng và kích cỡ dày

1.1.5.2 AMOLED

AMOLED (Active-Matrix Organic Light-Emitting Diode) là màn hình dùng các đèn đi-ốt hữu cơkhông cần tới lớp phân cực, tinh thể hay đèn nền như màn hình LCD. Nhờ cơ chế này,

AMOLED có một số ưu điểm so với công nghệ LCD. Nguyên lý hoạt động khá đơn giản: Dòng điện được lớp bóng bán dẫn (transistor) điều khiển đi qua lớp đi-ốt hữu cơ ở trên thì các đi-ốt ở lớp này sẽ phát sáng. Thay đổi dòng điện trên các bóng bán dẫn thì cường độ ánh sáng thay đổi theo từ đó có thể tạo hình ảnh như mong muốn.



Hình 1-11: Sơ đồ tấm hiển thị AMOLED

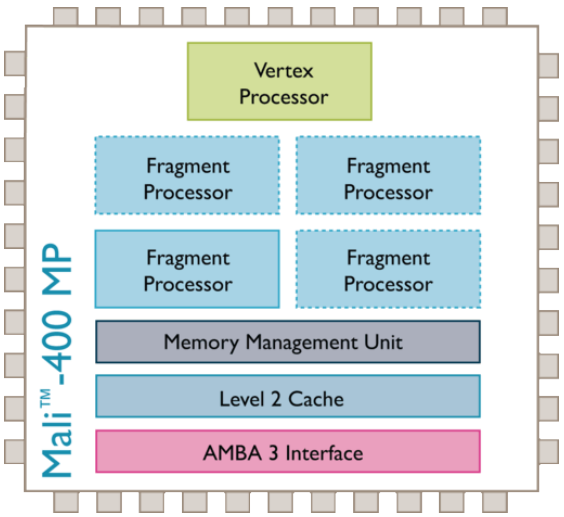
Ưu điểm:

- Kích thước mỏng và linh hoạt
- Màu sắc rực rỡ và độ tương phản cao
- Góc nhìn rộng
- Tiêu thụ điện ít hơn

Nhược điểm:

- Màu sắc rực rỡ nên đôi khi không thật.
- Không bền bằng LCD
- Màn hình loại ma trận PenTile, rẻ hơn nhưng chất lượng kém hơn.

1.1.6. Bộ xử lý đồ họa

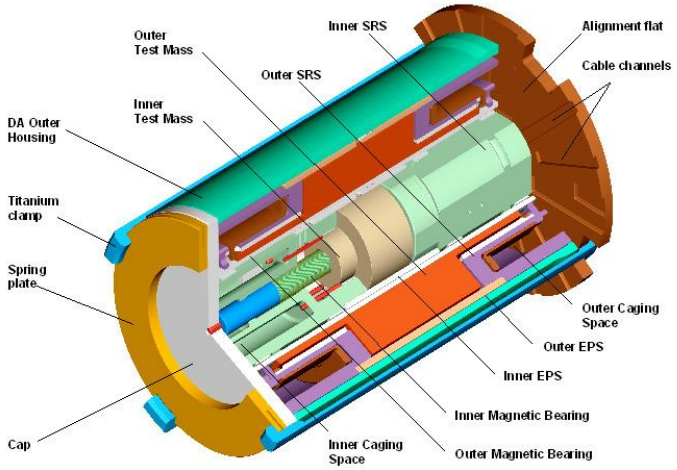


Hình 1-12: Kiến trúc bên trong GPU Mali của ARM

GPU (Graphics Processing Unit) là thuật ngữ chung chỉ chip xử lý độ họa, trong các sản phẩm SoC của các hãng đều được tích hợp các GPUđể đảm nhiệm chức năng xử lý các lệnh đồ họa.GPU xử lý đồ họa 2D/3D với nhiều tần số xung nhịp khác nhau từ 200MHz đến 400sMHz và được tích hợp thành nhiều lõi tuy theo nhu cầu của sản phẩm và có thể xử lý 204 tỷ phép tính/giây. GPU của các hãng đều hỗ trợ OpenGL và DirectX.

1.2 CÁC THIẾT BỊ ĐO LƯỜNG

1.2.1. Gia tốc kế



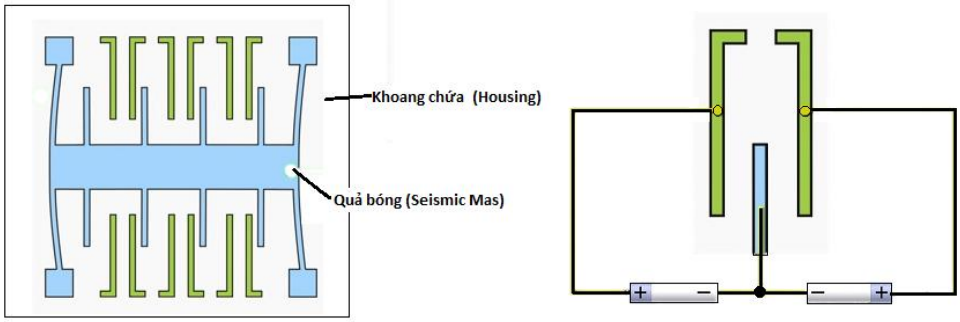
Hình 1-13: Hình minh họa gia tốc kế dùng trong máy bay

Để phát hiện chuyển động của thiết bị di động. Trong thiết bị được lắp một gia tốc kế giúp cho thiết bị nhận diện được chiều xoay của thiết bị để điều chỉnh các phần mềm phù hợp với trạng thái mới của điện thoại.

Gia tốc kế nguyên thủy là một khoang chứa hình trụ có chứa quả bóng gắn lò xo hình []. Khoang chứa này được gắn liền vào vật thể cần đo gia tốc, còn quả bóng di chuyển một chiều bên trong khoang chứa. Khi thiết bị chuyển động quả bóng cũng sẽ di chuyển làm lò xo co dãn. Dựa vào độ co dãn của lò xo thiết bị có thể đo được lực và gia tốc của chuyển động.

Hình 1-14: Cấu tạo cơ bản của gia tốc kế

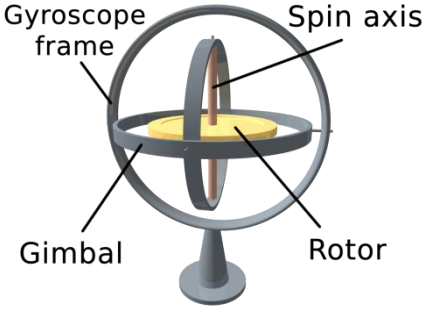
Gia tốc kế trong thiết bị di động là một thiết bị điện tử hoạt động dựa trên nguyên lý của gia tốc kế cơ bản này.



Hình 1-15: Minh họa hoạt động của gia tốc kế điện tử

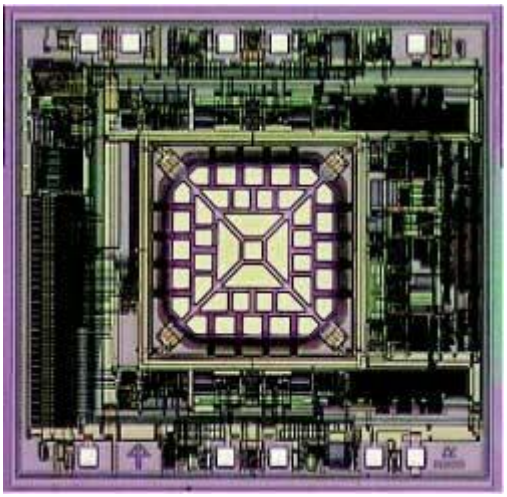
Thiết bị đo lường này đo sự chuyển động của vật bằng cách quan sát mức độ điện áp ở hai cực đầu ra của thiết bị đo.

1.2.2. Con quay hồi chuyển



Hình 1-16: Hình ảnh minh họa con quay hồi chuyển

Con quay hồi chuyển (gyroscope) là một trong những bộ phận cảm ứng quan trọng trong thiết bị di động. Do gia tốc kế chỉ có thể đo được gia tốc tuyến tính của thiết bị nên phải kết hợp với con quay hồi chuyển có thể nhận biết được hướng của thiết bị, hệ thống có thể dễ dàng ghi nhận những chuyển động theo cả phương ngang hoặc phương thẳng đứng. Trong các thiết bị di động không sử dụng con quay hồi chuyển cơ học mà dùng thiết bị gọi là MEMS (MicroElectroMechanical System – Hệ thống vi cơ điện tử) thiết bị này mô phỏng chính xác hoạt động của các thiết bị cơ học trong một con chipvới kích thước vài micromet. Con quay MEMS được sử dụng nhiều trên các thiết bị dùng đến cơ điện tử.



Hình 1-17: Hình ảnh minh họa MEMS

1.2.3. Định vị vệ tinh

Hệ thống định vị toàn cầu - GPS (**Global Positioning System**) được dùng phổ biến trong nhiều lĩnh vực đời sống. Hệ thống này sử dụng 24 vệ tinh bay quanh trái đất với hệ kinh vĩ độ được cố định sẵn.GPS đầu tiên được ứng dụng trong quốc phòng và ngành hàng hải sau đó được chuyển sang ngành vận tải,… Ngày nay hệ thống này được ứng dụng trong hầu hết các lĩnh vực như trắc địa, lập bản đồ, cứu nạn,…

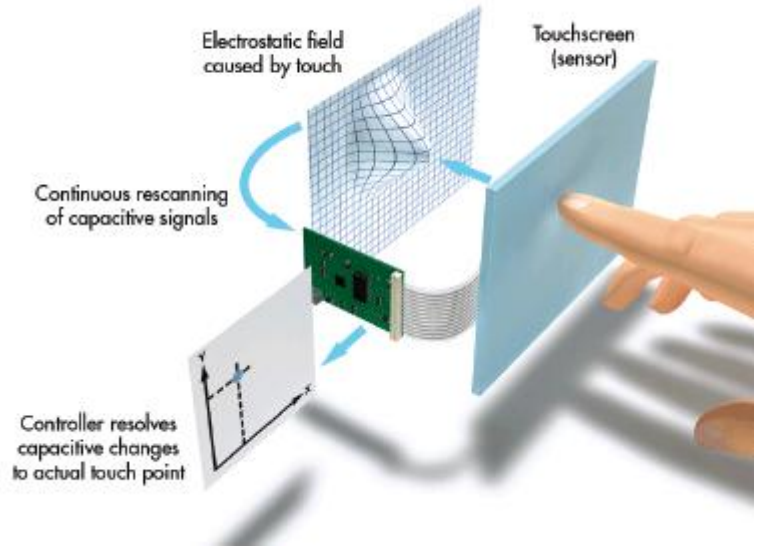
Hình 1-18: Một số tính năng của GPS:

- Độ chính xác định vị cao.
- Phạm vi sử dụng toàn cầu
- Thời gian hoạt động 24h/ngày.
- Giá thành thiết bị rẻ.

1.2.4. Cảm biến điện dung

Màn hình là thiết bị hiển thị thông tin nhưng đối với thiết bị di động nó còn đóng vai trò quan trọng là nhận điều khiển từ người dùng. Bộ phận cảm ứng được bổ sung vào phần dưới màn hình còn được gọi là lớp cảm ứng số hóa. Hầu hết các điện thoại thông minh hiện nay đều trang bị màn hình cảm ứng điện dung.

Lớp cảm ứng điện dung sử dụng công nghệ PCT (projected capacitive touch), tạo ra một "lưới" trên màn hình. Lưới này phát hiện tiếp xúc của người dùng bằng cách giám sát sự thay đổi hiệu điện thế. Dựa trên cường độ điện trường và tính chất đa chiều của trường tĩnh điện các tiếp xúc có thể thực hiện hoặc gián tiếp hoặc trực tiếp trên màn hình.

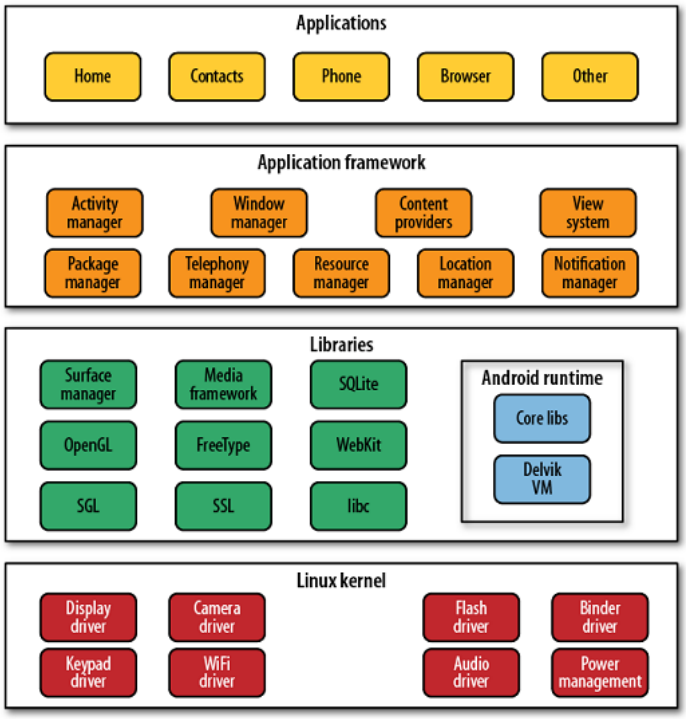


Hình 1-19: Hình minh họa cơ chế hoạt động của màn hình cảm ứng điện dung

Lớp cảm ứng được đặt ngay trên lớp tinh thể lỏng của màn hình LCD, nhưng nằm dưới lớp kính bảo vệ. Trong khi đó màn hình AMOLED được được tích hợp lớp cảm ứng này cùng với lớp đi-ốt phát quang, làm cho trong suốt tiết kiện không gian cùng với chi phí[2].

1.3 KIẾN TRÚC CỦA NỀN TẢNG ANDROID

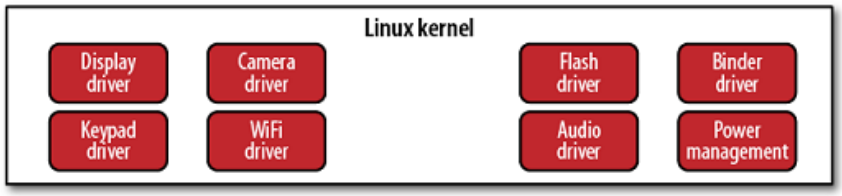
Android là hệ điều hành của hãng Google, với hướng tiếp cận là một hệ điều hành cho thiết bị di động [11]. Android nhanh chóng trở thành hệ điều hành phổ biến nhất trên thế giới. Được xây dựng từ nhân *Linux*, *Android* được mở rộng thêm nhiều chức năng và cung cấp nền tảng cho lập trình ứng dụng. Phần dưới đây sẽ lần lượt mô tả chi tiết từng thành phần trong Android.



Hình 1-20: Kiến trúc cơ bản của hệ điều hành Android

1.3.1. Nhân của hệ điều hành

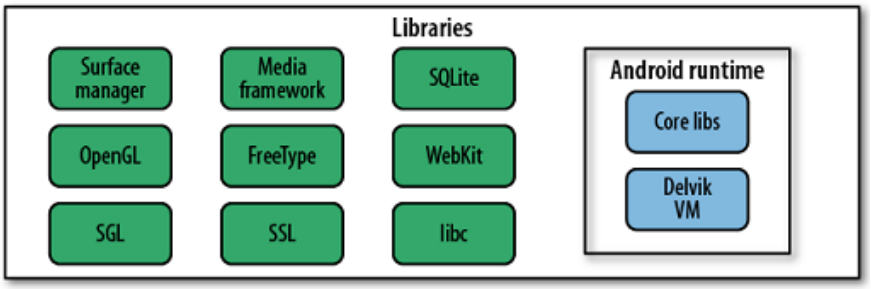
Android là một sản phẩm điển hình của việc hợp tác và phát triển mã nguồn mở. Hệ điều hành Android sử dụng nhân Linux để phát triển. Nhân Linux phát triển đến đâu thì Android lại được cập nhật và phát triển đến đó.



Thành phần cơ bản của Linux có các phần chính là Trình điều khiển thiết bị hiển thị, Camera, bàn phím, chuột, kết nối Wifi, âm thanh, quản lý nguồn điện, ổ USB và thẻ nhớ cùng với các trình kết nối khác.

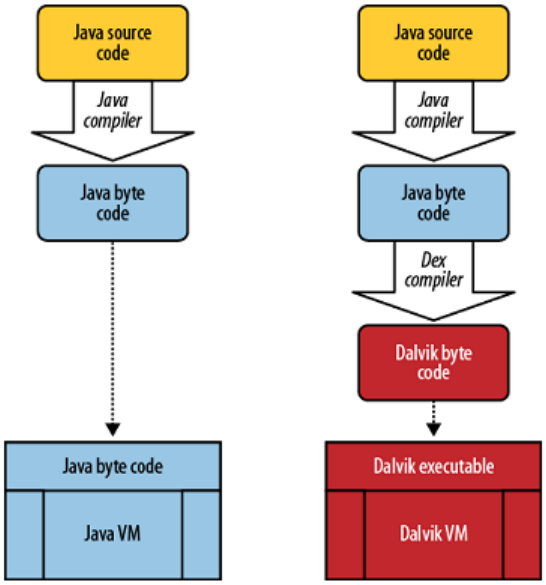
1.3.2. Thư viện

Android cung cấp phong phú các thư viện để hỗ trợ lập trình cũng như một nền tảng phát triển ứng dụng vững trãi. Các thư viện này đã đem lại cho Android sức sống mãnh liệt và trên thực tế có vô số các phần mềm chạy trên nền tảng Android.



- Surface Manager: quản lí giao diện bao gồm lớp giao diện 2D và 3D.
- Thư viện media: dựa tên OpenCORE của PacketVideo, một bộ thư viện hỗ trợ đọc, ghi lại nhiều định dạng âm thanh, video, hình ảnh như: MPEG4, H.264, MP3, AAC, AMR, JPG, và PNG.
- SQLite: hệ thống cơ sở dữ liệu có kích thước nhỏ nhưng mạnh mẽ cho tất cả ứng dụng.
- OpenGL: Thư viện các hàm xử lý đồ họadựa trên OpenGL ES 1.0 API; thư viện gồm phần cứng hỗ trợ 3D và phần mềm hỗ trợ tối ưu 3D.
- FreeType: trình bày kiểu chữ và hình ảnh.
- Thư viện hệ thống C: được rút ra từ thư viện hệ thống C chuẩn (libc).
- Webkit: là bộ thư viện trình duyệt web mới hỗ trợ trình duyệt Android và WebView.
- SGL: hệ thống đồ họa 2D cơ sở.
- SSL: Giao thức bảo mật hỗ trợ sử dụng giao thức mã hóa Secure Sockets Layer trong bảo mật truyền thông Internet

Trong phần này có một phần rất quan trọng đó là *Android Runtime*



Hình 1-21: Sự so sánh Java VM và Dalvik VMs

Mặc dù Android chấp nhận cả hai ngôn ngữ lập trình Java và C, tuy nhiên Java chiếm ưu thế khi lựa chọn ngôn ngữ lập trình cho Android. Google đã xây dựng máy ảo riêng để biên dịch và chạy các ứng dụng trên Android. Đối với ngôn ngữ lập trình C, Google cung cấp framework khác là *libc*, khi lập trình thì dựa vào NDK (Native Development Kit). Ngôn ngữ lập trình cho Android được phát triển trên môi trường Java, nhưng chạy trên máy ảo Dalvik VM. Các thư viện cơ bản của Android cung cấp hầu hết các chức năng có trong thư viện cơ bản của Java cũng như là thư viện riêng của Android.

Máy ảo Dalvik: Dalvik là máy ảo để chạy các ứng dụng trên Android, đã được tối ưu để thực thi đa nhiệm một cách hiệu quả. Nền tảng kỹ thuật này dựa vào lõi Linux để thực hiện đa luồng và quản lý tài nguyên bộ nhớ.

Thư viện Core libs

Hệ điều hành Android hỗ trợ một số các APIs để phát triển ứng dụng.

- ***android.util:*** Gói tiện ích quản lý nhiều lớp ở mức hệ thống như:các lớp (List, Stack…) lớp xử lý chuỗi, lớp xử lý XML,…

- ***android.os:*** Gói hệ điều hành này cung cấp các phương thức cho phép truy cập mức hệ thống như là trao đổi các thông điệp, thông tin chéo, đồng hồ và gỡ lỗi.

- ***android.graphics:***Gói này cung cấp các lớp đồ họa.

- ***android.text:***Cung cấp các phương thức xử lý văn bản

- ***android.database:***Gói này cung cấp các phương thức dùng thao tác với cơ sở dữ liệu.

- ***android.content:***Cung cấp phương thức liên quan lập trình nội dung.

- ***android.view:***Gói khung nhìn người dùng cơ bản nhất. Các chương trình khi tương tác với người dùng đều thông qua các *View*.

- ***android.widget:***Gói này được xây dựng dựa trên gói View. Những thành phần giao diện được tạo sẵn đem vào sử dụng để tạo nên giao diện người dùng một cách nhanh chóng.

- ***com.google.android.maps:***Bộ giao diện ứng dụng dùng để truy cập đến bản đồ sẵn trong Androidtừ các phần mềm ứng dụng.

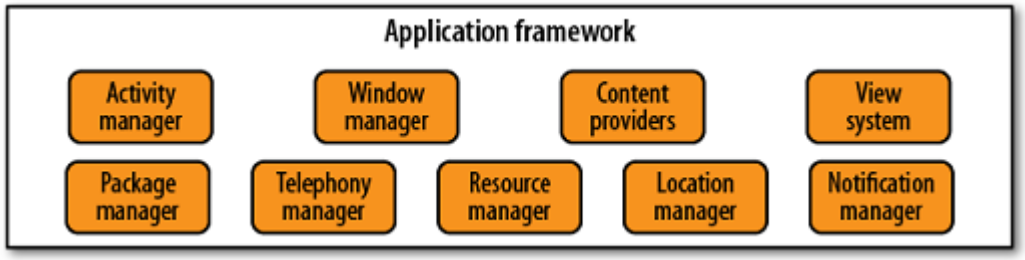
- ***android.app*:**Gói thư viện này cung cấp phương thức truy cập đến dữ liệu của các ứng dụng.

- ***Android.provider:***Gói thư viện này tạo thuận lợi cho các nhà phát triển truy cập đến các *Content Provider* tiêu chuẩncho phép truy cập đến cơ sở dữ liệu chuẩn trong tất cả các bản Android.

- ***Android.telephony:*** Các API cung cấp cách thức kết nối với tầng viễn thông trong thiết bị, cho phép tạo, nhận, theo dõi các cuộc gọi, tình trạng các cuộc gọi và tin nhắn SMS.

- ***android.webkit:*** Gói WebKit cung cấp giao diện lập trình ứng dụng (API)có thể trao đổi nội dung dựa trên nền Web, gồm lớp WebView, trình quản lý cookieđể tạo ra giao diện Web và nhúng trong ứng dụng.

1.3.3. Khung ứng dụng trên Android



Kiến trúc của Android khuyến khích khái niệm Thành phần sử dụng lại, cho phép công bố và chia sẻ các Activity, Service, dữ liệu, với các ứng dụng khác với quyền truy cập được quản lý bởi khai báo.

Cơ chế đó cho phép người lập trình tạo ra một trình quản lý danh bạ hoặc trình quay số điện thoại mà có các thành phần người khác có thể tạo mới giao diện và mở rộng chức năng thay vì tạo lại chúng.

Những dịch vụ sau là những dịch vụ kiến trúc cơ bản nhất của tất cả các ứng dụng, cung cấp một framework cho mọi mọi phần mềm được xây dựng:

- Actitvity Manager: Điều khiển vòng đời của các Activity bao gồm cả quản lý các tầng Activity.

- Views: Được sử dụng để tạo lập các giao diện người dùng cho các Activity

- Notification Mamager: Cung cấp một cơ chế cố định và quy củ cho việc gửi các thông báo đến người dùng.

- Content Provider: Cho phép ứng dụng chia sẻ dữ liệu giữa các ứng dụng.

- Resource Manager: Hỗ trợ các thành phần không thuộc mã nguồn như là chuỗi ký tự, đồ họa được đặt bên ngoài.

1.3.4. Tầng ứng dụng

Đây là lớp trên cùng của kiến trúc nền tảng Android. Android sẽ hoạt động với một bộ các ứng dụng bao gồm ứng dụng thư điện tử, gửi tin nhắn, lịch, bản đồ, trình duyệt web, danh bạ v.v… Tất cả các ứng dụng được viết bằng ngôn ngữ Java. Các ứng dụng này có thể được cung cấp sẵn hoặc được phát triển bởi những lập trình viên.

1.3.5. Các thành phần trong một ứng dụng Android

Một ứng dụng trên môi trường Android có các thành phần sau:

- *Activities (Hoạt động)*
- *Services*
- *Broadcast Receivers*
- *Content Provider*

Một ứng dụng không nhất thiết có đầy đủ các thành phần này.Mọi thông tin về ứng dụng đều được ghi trong tệp tin *AndroidManifest.xml*.

1.3.5.1 Hoạt động (Activity)

Hoạt động là giao diện trong đó người dùng tương tác với ứng dụng khi được kích hoạt. Một ứng dụng có nhiều hoạt động và các hoạt động này gọi lẫn nhau để thực hiện yêu cầu. Mỗi hoạt độngđược dẫn xuất từ lớp *android.app.Activity*.

Mỗi *hoạt động*thể hiện trên một cửa sổ. Thông thường kích thước cửa sổ bằng màn hình, ngoài ra nó có thể có thêm các cửa sổ con khác như là hộp thoại,…Thông tin trong từng cửa sổ được cung cấp bởi hệ thống cấp bậc các *View* (là đối tượng của lớp Views).

Vòng đời của hoạt động

Các *hoạt động* trong hệ thống được quản lý bởi một cấu trúc dữ liệu ngăn xếp. Khi khởi tạo, *hoạt động*được đẩy vào trong ngăn xếp, các trạng thái thực thi và hoạt động trước đó sẽ chuyển sang trạng thái chờ. *Hoạt động* này trở lại trạng thái kích hoạt khi khởi tạo hoàn tất.

Một *hoạt động* có ba trạng thái cơ bản là:

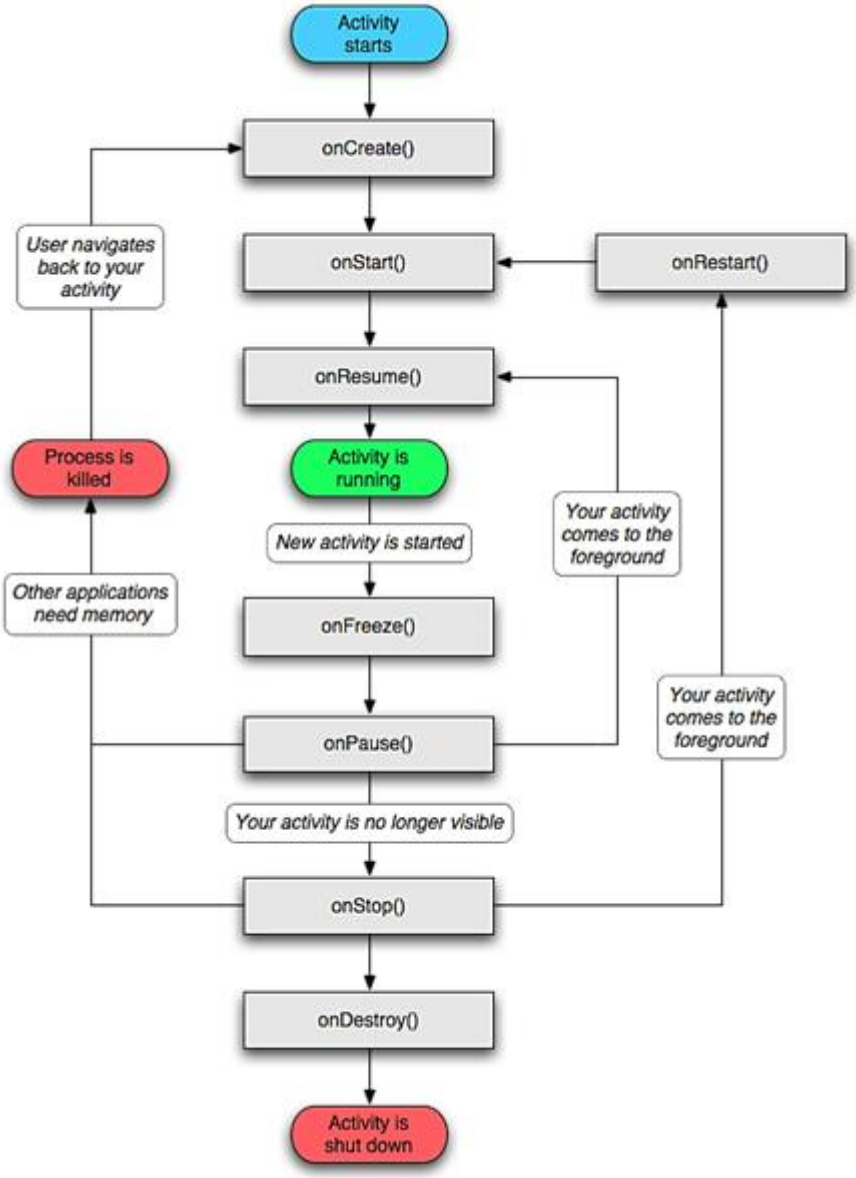
- ***Hoạt động*** (*Active)* hoặc chạy (*running)* khi ứng dụngxuất hiện trên màn hình và nhận tương tác người dùng.

- ***Tạm dừng*** (*Paused)* khi *hoạt động*không còn trọng tâm trên màn hình nhưng vẫn hiện thị trước người dùng.

- ***Dừng****(Stopped)* Khi một *hoạt động* hoàn toàn bị che khuất nhưng *hoạt động* vẫn còn lưu trữ toàn bộ thông tin trạng thái, và thường bị hệ thống đóng lại khi thiếu bộ nhớ.

Khi luân chuyển giữa các trạng thái, ứng dụng sẽ gọi các hàm *callback* ứng với các bước chuyển trong lưu đồ hình [11].

Lưu đồ dưới đây mô tả vòng đời của một *hoạt động* trong Android.



Hình 1-22:Vòng đời của một hoạt động

Chu trình của một *hoạt động*được bắt đầu bằng phương thức *onCreate (Bundle)*và kết thúc bằng phương thức *onDestroy()*. Thời gian hoạt động của một *hoạt động* bắt đầu từ lời gọi phương thức *onStart()*đến khi phương thức onStop() được triệu gọi. Toàn bộ tài nguyên của hệ thống do *hoạt động*sử dụng tiếp tục được lưu giữ cho đến khi bị hệ thống thu hồi.*Hoạt động* được phục hồi khi phương thức *onResume()* được gọi.

1.3.5.2 Service

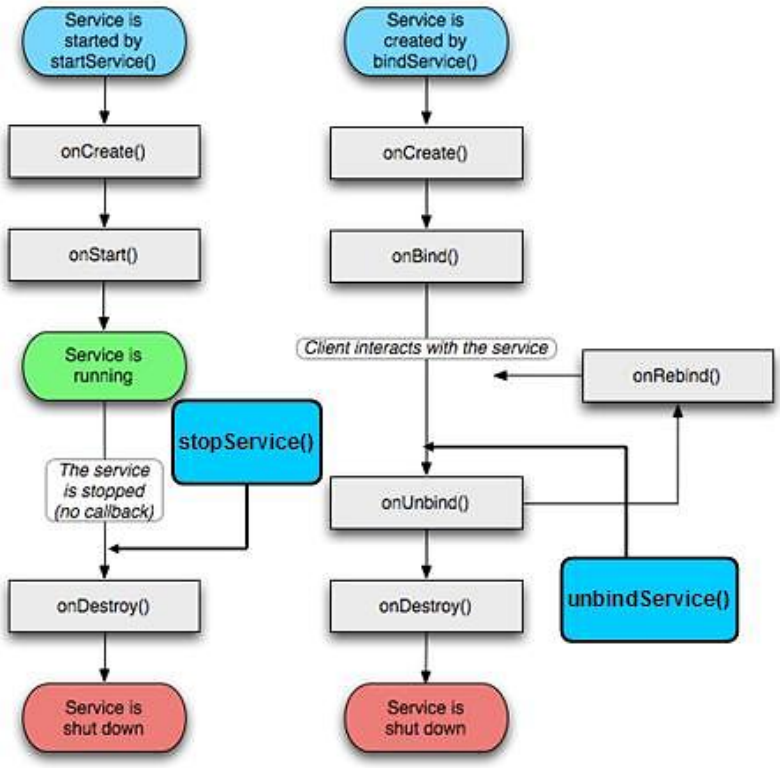
Dịch vụ (service) là các mô đun thực thi ẩn trong hệ thống. Mỗi *dịch vụ* được kế thừa từ lớp cơ sở trong gói android.app.Service.Các *dịch vụ* được kích hoạt hoặc kết nối thông qua việc truyền các tham số cho *dịch vụ*và nó được kích hoạt bằng cách gọi

phương thức *Context.startService()*. Các ứng dụng có thể liên kết tới một dịch vụ bằng cách dùng phương thức *Context.bindService().*

Vòng đời của một dịch vụ

Vòng đời của một *dịch vụ* là quá trình hoạt động của *dịch vụ*từ khi được kích hoạt đến khi bị loại khỏi hệ thống. Khi hệ thống gọi tới phương thức *Context.startService()dịch vụ* được thực thi cho tới khi phương thức*Context.stopService()* được gọi.

Các ứng dụng muốn kết nối tới *dịch vụ*thì phương thức *Context.bindService()*được kích hoạt và nó sẽ nhận được một đối tượng *IBinder* do *dịch vụ* trả về để từ đó gọi các phương thức *Callback*cho phù hợp.*Dịch vụ* được thực thi cho tới khi nào đối tượng *IBinder* còn tồn tại.

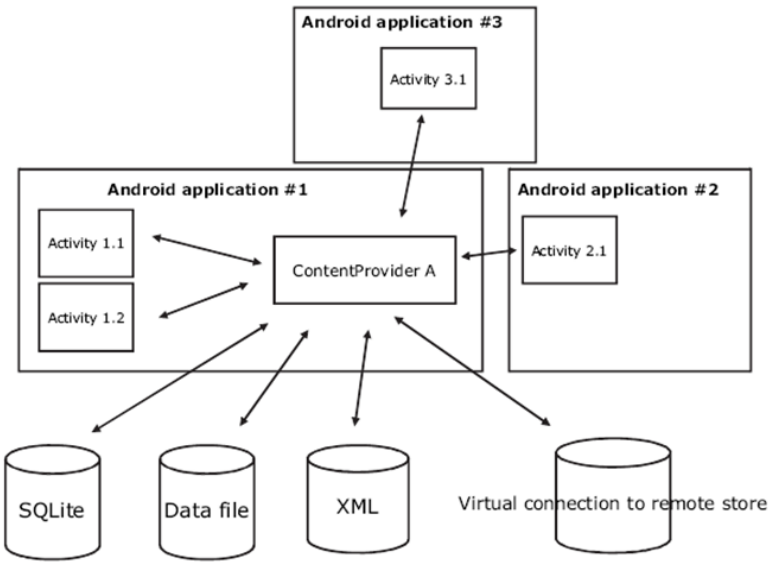


Hình 1-23: Lưu đồ chuyển trạng thái của dịch vụ

1.3.5.3 Bộ nhận thông điệp quảng bá (Broadcast receivers)

Bộ nhận thông điệp quảng bá dùng để nhận và phản hồi lại các thông điệp quảng bá do hệ thống thông báo như múi giờ bị thay đổi, nguồn yếu, và các thông điệp từ các thiệt bị đo lường gửi về.Tất cả hoạt động của bộ nhận thông điệp quảng bá được kế thừa từ lớp *BroadcastReceiver*.

1.3.5.4 Bộ cung cấp nội dung



Hình 1-24: Kiến trúc của bộ cung cấp nội dung trong Android

Bộ cung cấp nội dung *(Content Provider)* có chức năng cung cấp một tập các phương thức cho phép một ứng dụng có thể lưu trữ và trao đổi dữ liệu thông qua trong các tập tin hoặc sử dụng cơ sở dữ liệu SQLite sẵn cóvà được quản lý bởi *Bộ cung cấp nội dung* của nó.*Bộ cung cấp nội dung* là tính năng ưu việt của Android nhằm chia sẻ dữ liệu giữa các ứng dụng một cách dễ dàng.

1.4 CÔNG CỤ VÀ NGÔN NGỮ LẬP TRÌNH

Xuất phát điểm của Android dựa trên nền Linux nên hệ điều hành được hỗ trợ đầy đủ của cộng đồng mã nguồn mở và ngôn ngữ lập trình các ứng dụng là Java và C, C++.

1.4.1. Ngôn ngữ lập trình

Ngôn ngữ dùng để lập trình cho Android là Java và C, Hai ngôn ngữ này phổ biến trong sản xuất phầnmềm. Về cơ bản, hai ngôn ngữ đều có kiến trúc giống nhau, từ bảng chữ cái, các quy định đặt tên, biểu thức, các phép toán, các kiểu dữ liệu cho đến các cấu trúcrẽ nhánh, lặp, kiểu cấu trúc, hướng đối tượng, đa nhiệm và xử lý song song,…do vậy các nhà phát triển có nhiều lựa chọn hơn cho các dự án của họ.

1.4.2. Công cụ cho lập trình

1.4.2.1 Android SDK

Android SDK là sản phẩm của Google nhằm cung cấp cho các lập trình viên một công cụ phát triển các ứng dụng trên hệ điều hành Android của hãng [3]. Về cơ bản, Android SDK gồm một số thành phần sau:

Eclipse và ADT plugin

Android SDK tools

Android Platform tools

Eclipse và ADT-plugin là môi trường để lập trình các ứng dụng cho Android.

Android SDK tools

Android software development Kit bao gồm đầy đủ các bộ công cụ cho phát triển sản phẩm trên hệ điều hành Android. Bao gồm hệ thống gỡ rối, các thư viện, trình mô phỏng dựa trên QEMU, các tài liệu, chương trình mẫu,… Hiện tại, nó hỗ trợ các nền tảng bao gồm các máy chạy Linux, MacOS X trở đi, Windows XP trở lên, và đương nhiên nó hỗ trợ cho các phiên bản của hệ điều hành Android gồm (Android IDE - Java, C++). Một môi trường phát triển chuẩn là Eclipse chạy trình cắm *Android Development Tools (ADT), IntelliJ IDE* và *NetBeans IDE*. Ngoài ra các nhà phát triển còn có thể dùng các trình soạn thảo cho Java và XML để tạo ứng dụng.

Các ứng dụng của Android được đóng gói dưới dạng tệp tin định dạng *.apk*và được lưu trong đường dẫn `/data/app` trong hệ điều hành *Android* các gói .apk chứa tệp tin *.dex* được biên dịch bằng máy ảo *Dalvik*[4].

1.4.2.2 Android NDK

Android NDK (Native Development Kit) là một bộ công cụ để phát triển ứng dụng cho Android nhưng có điều sử dụng ngôn ngữ lập trình C. Các thư viện được viết bằng ngôn ngữ C và một số ngôn ngữ khác có thể được biên dịch sang ARM, MIPS hoặc x86 native code và được cài đặt sử dụng trong *Android Native Development Kit.*Các lớp của *Native* có thể được gọi từ mã Java chạy trong máy ảo Dalvik sử dụng lời gọi thư viện `System.loadLibrary`.

Các ứng dụng được xây dựng xong có thể được biên dịch cài đặt bằng các công cụ truyền thống. Tuy nhiên, theo như bản công bố của Android, NDK không chỉ dùng để phát triển các ứng dụng đơn lẻ mà dùng để xây dựng các ứng dụng có quy mô lớn vì ngôn ngữ C được ưa thích hơn.

1.4.2.3 Eclipse

Là công cụ mã nguồn mở thuộc nhóm các công cụ hỗ trợ trong lập trình từ các ngôn ngữ Java, C/C++, Ngôn ngữ mô hình, ngôn ngữ đặc tả, … được dùng phổ biến hiện nay và đã trở thành những sản phẩm chuẩn mực trong thế giới mã nguồn mở.

+ Cài đặt: Cài đặt Eclipse rất đơn giản và thuận tiện. Ứng dụng Eclipse được xây dựng để khi cài đặt, người dùng chỉ cần download về và giải nén là có thể chạy được. Nếu dùng Eclipse cho một ứng dụng chuyên biệt nào đó thì sản phẩm đó chỉ cần xây dựng dưới dạng một trình cắm (plugin) rồi tích hợp vào Eclipse là xong.

Địa chỉ Internet để lấy Eclipse:

http://www.eclipse.org

Google đã tích hợp sẵn ADT-plugin để thuận tiện cho các nhà phát triển ứng dụng và được để tại địa chỉ:

http://developer.android.com/tools/sdk/eclipse-adt.html

1.4.2.4 Netbean

NetBean cũng là một công cụ xuất thân từ mã nguồn mở chỉ hỗ trợ để lập trình Java. Từ khi hãng Sun bị Oracle thâu tóm nay trở thành sản phẩm đóng của Oracle nhưng miễn phí và mở rộng hỗ trợ cho cả ngôn ngữ lập trình C/C++, PHP, HTML,…

Địa chỉ:

`https://netbeans.org/`

+ Cài đặt:

- Download Netbean và cài đặt xong.

- Vào Menu Tools → Plugins →Hộp thoại Plugins xuất hiện→ Chọn Settings →Chọn nút Add → Nhập địa chỉ vào ô bên dưới:

`http://nbandroid.org/release72/updates/updates.xml`

Đợi cho Netbean cập nhật xong. Chọn *Avainable Plugins→ chọn Android rồi cài đặt.*

1.4.3. Một số Game engine

Với tốc độ phát triển rất nhanh của thị trường trò chơi trên thiết bị di động. Nhiều hãng đã phát triển các game engine của riêng họ. Sau đây có một số game engine phổ biếnAndEngine, Unity3D, CryEngine, Unreal Engine,…

1.5 QUY TRÌNH XÂY DỰNG PHẦN MỀM TRÊN ANDROID

Xây dựng phần mềm nói chung và phần mềm cho thiết bị di động nói riêng phải tuân thủ theo quy trình sản xuất như các sản phẩm thông thường, nhưng cần xét đến khía cạnh đặc thù của từng loại thiết bị. Đối với thiết bị di động, ngoài yếu tố lựa chọn sản phẩm để sản xuất, khi thiết kế chương trình cần tính toán nhiều yếu tố:

+ Tài nguyên phần cứng: Tiêu hao tài nguyên phần cứng cho ứng dụng chạy.

+ Cách thức người dùng tương tác với phần mềm.

+ Các chức năng của thiết bị di động có sẵn như: Máy gia tốc kế, con quay hồi chuyển, cảm ứng từ, cảm ứng ánh sáng,…

Quy trình xây dựng ứng dụng trên Android

Quy trình sản xuất phần mềm trên Android không khác gì với các sản phẩm phần mềm khác.

Xác định yêu cầu của sản phẩm được thực hiện bởi nhóm chuyên gia có chuyên môn về nghiên cứu thị trường. Trong bước này, các chức năng, hình hài của sản phẩm phải được xác định rõ.

Lập kế hoạch thực hiện dự án,người quản trị dự án phải tính toán toàn bộ nguồn lực bao gồm thời gian, tài chính, con người tham gia vào dự án. Kết quả của bước này là một bản mô tả tổng thể dự án triển khai trong bao lâu, cần bao nhiêu người trong từng giai đoạn, tiêu hết bao nhiêu tiền trong từng giai đoạn, và điều quan trong nữa là nếu xảy ra rủi ro thì sẽ rơi vào tình huống nào và tổn thất bao nhiêu tiền?

Trong bước tổ chức thực hiện dự án,người quản trị dự án phân bổ công việc theo kế hoạch cho từng nhóm, với một lịch trình cụ thể, giám sát chặt chẽ theo ngày (thậm chí theo giờ) để có phương án xử lý kịp thời. Các pha trong sản xuất phần mềm tuân thủ theo các quy trình chung của ngành. Một số hãng phần mềm lớn họ áp dụng các chuẩn CMMI trong sản xuất sẽ đảm bảo dự án thực thi tốt hơn.

CHƯƠNG 2: LẬP TRÌNH GAME CHO ĐIỆN THOẠI THÔNG MINH

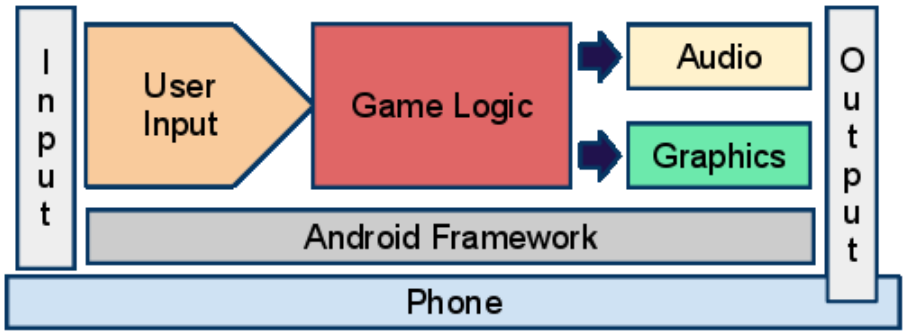
2.1 Giới thiệu

Lập trình ứng dụng trên thiết bị di động là một trong những hướng phát triển chính trong ngành công nghệ thông tin hiện nay. Trong đó, lập trình các trò chơi là chuyên ngành hẹp nhưng đã đem lại nhiều lợi nhuận. Trong chương này khóa luận lựa chọn hướng phát triển ứng dụng game trên hệ điều hành Android làm đại diện cũng như các khía cạnh kỹ thuật của lĩnh vực này.

2.2 KIẾN TRÚC CỦA TRÒ CHƠI TRÊN ANDROID

Các trò chơi được xây dựng trên hệ điều hành Android có kịch bản riêng nhưng đều có quy trình giống nhau được minh họa tronghình dưới đây [5].

2.2.1. Kiến trúc chung



Hình 2-1: Kiến trúc cơ bản của một trò chơi trên Android

2.2.1.1 Input

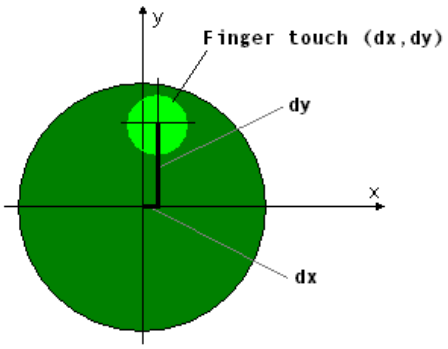
Là phần thông tin đưa vào để tương tác với trò chơi. Các dữ liệu này được đưa vào thông qua các bộ phận như: Màn hình cảm ứng, gia tốc kế, con quay hồi chuyển, cảm ứng từ trường, cảm ứng ánh sáng, camera, micro, và cả bàn phím vật lý,…

2.2.1.2 User Input

Là phần kế tiếp để nhận thông tin đầu vào. Một số trò chơi cần người dùng thêm một số thao tác như tô màu, nhận diện hình, âm thanh,… để trước khi thực hiện trò chơi.

2.2.1.3 Game logic

Game logic là mô đun chịu trách nhiệm về việc thay đổi trạng thái của các nhân vật cũng như các đối tượng trong trò chơi.Ví dụ các nhân vật, địa hình, địa vật, súng, đạn, tia lade,…



Hình 2-2: Hình minh họa ngón tay chạm vào vùng điều khiển trong User Input

Trong hình trên hình tròn xanh lơ biểu diễn ngón tày người dùng chạm vào vùng điều khiển. Mô đun User Input thông báo cho *Game Engine* (*Game Logic*) và đối chiếu vào hệ tọa độ. **dx**,**dy** là khoảng cách tính theo điểm ảnh liên quan tới trung tâm của hình tròn điều khiển. Game engine tính toán tốc độ mới mà nó phải thiết lập cho nhân vật và hướng của nhân vật sẽ di chuyển. Nếu **dx** có giá trị dương có nghĩa là nhân vật di chuyển sang phải và nếu **dy** có giá trị dương thì nhân vật di chuyển về phía trước.

2.2.1.4 Audio

Mô đun này sẽ phát ra âm thanh tương ứng với từng trạng thái hiện tại của nhân vật. Trong hầu hết các trò chơi các nhân vật, đối tượng phát ra âm thanh ở các trạng thái khác nhau của chúng và do các thiết bị chơi trò chơi được giới hạn trong một vài kênh nào đó (nghĩa là có bao nhiêu âm thành được phát cùng một lúc)nó phải quyết định âm thanh để chơi. Ví dụ trong trò chơi khi xuất hiện một nhân vật đối kháng thì một âm thanh kinh dị hoặc đặc biệt được phát ra để gây ra chú ý cho người chơi đồng thời cần một kênh âm thanh riêng cho nhân vật như chuẩn bị vũ khí hoặc lẩn trốn thì âm thanh phát ra phải đanh, gọn,…

2.2.1.5 Graphics

Mô đun này chịu trách nhiệm hiển thị các trạng thái trò chơi trên màn hình. Điều này đơn giản chỉ là vẽ trực tiếp lên miếng toan (canvas) từ một góc nhìn hoặc biểu diễn hình ảnh từ bộ đệm đồ họa, ở đây, các đối tượng đồ họa được xử lý trước rồi cho vào bộ đệm sau đó được đưa ra màn hình dưới góc nhìn tùy chỉnh hoặc của chính OpenGL. Các khung hình FPS (Frame Per Second) là một trong các thông số đánh giá chất lượng hình ảnh được hiển thị, nếu thông số là 30 FPS có nghĩa là có 30 ảnh được hiển thị trong từng giây. Thông số này đánh giá được chất lượng thiết bị là rất tốt.

2.2.1.6 Output

Phần này là kết quả của quá trình xử lý các dữ liệu trong trò chơi, nó thể hiện cả hình ảnh, âm thanh hoặc rung.

2.2.2. Kỹ thuật âm thanh

Âm thanh trong trò chơi tăng cường cảm xúc của người chơi. Nó cung cấp một kết nối cảm xúc mà không thể nào đạt được bằng cách khác. Các nhà sản xuất điện ảnh đã tiên phong trong lĩnh vực này và đã đầu tư rất nhiều tài chính để tạo ra các nhạc nền phù hợp cho phim. Đối với các trò chơi, âm nhạc có thể không có quy mô lớn như trong những bộ phim nhưng âm nhạc rất cần thiết trong trò chơi. Trong các *game engine* cung cấp các tính năng cho phép bổ sung âm thanh vào trong các trò chơi.

Âm thanh sử dụng trong trò chơi được các nhạc sỹ và nghệ sỹ chuyên nghiệp thiết kế và xây dựng. có hai loại âm thanh trong chương trình trò chơi.

+ Âm nhạc thường sử dụng như nhạc nền trong trò chơi.

+ Hiệu ứng âm thanh cùng xảy ra với các sự kiện bên trong trò chơi.

Âm thanh nói chung trong các trò chơi là các chuẩn âm thanh đã có sẵn trên thế giới như: mp3, wav, wma,… tuy nhiên 2 chuẩn nén quan trong thường được dùng là OGGs, MP3s.

Âm nhạc

Âm nhạc rất quan trọng để thiết lập thái độ người chơi. Không có quy tắc nhất định cho việc sử dụng âm nhạc trong chương trình trò chơi. Nhạc vui, buồn, trầm, bổng đều tác động đến tâm lý của người chơi. Do đó mỗi kịch bản trong trò chơi được thiết kế một tập các tệp âm thanh riêng biệt. Một đặc tính chung trong trò chơi, các bậc kế tiếp càng ngày càng khó, nhịp độ tăng lên khi thách thức tăng. Các nhân vật đối kháng cũng hoạt động nhanh và tinh xảo hơn gây nhiều khó khăn cho người chơi. Đôi khi người chơi không muốn nghe nhạc từ trò chơi, ngược lại họ muốn nghe bản nhạc khác do chính họ chọn từ trong máy. Các thiết bị di động phải có những chế độ cho phép nghe nhạc trong khi các ứng dụng khác vẫn hoạt động.

Hiệu ứng âm thanh

Âm nhạc được phát trong suốt chương trình trò chơi hoạt động còn các hiệu ứng âm thanh chỉ xảy ra tương ứng với các sự kiện trong trò chơi, ví dụ tiếng súng nổ, tiếng va chạm, … tương ứng với các sự kiện khi bắn nhau hoặc khi các vật thể va chạm nhau.

Các hàm API liên quan đến lập trình với âm thanh

Khi lập trình với âm thanh, hệ thống cần đưa các thư viện xử lý âm thanh vào chương trình.

```
importandroid.media.AudioManager;
import android.media.SoundPool;
```

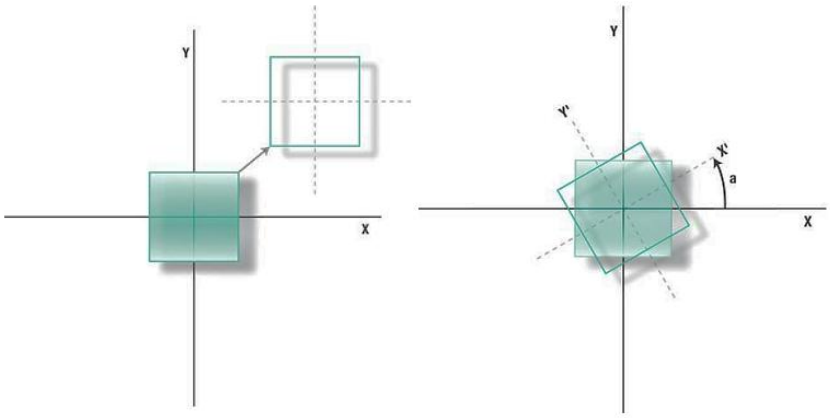
Trong các thư viện này có hàng trăm phương thức và hằng số giúp cho lập trình viên lập trình rất hiệu quả trong việc xử lý âm thanh.

2.2.3. Kỹ thuật đồ họa

Xử lý đồ họa trong trò chơi của android cũng không nằm ngoài nguyên lý xử lý đồ họa thông thường.Tuy nhiên đối với thiết bị di động đòi hỏi việc tối ưu tài nguyên và các thuật toán có khác so với máy tính các nhân và laptop. Hiện nay Android tích hợp thư viện xử lý đồ họa OpenGL ES (OpenGL for Embeded System) vào hệ thống, và các ứng dụng trò chơi của Android đều sử dụng thư viện này. Phiên bản mới nhất là OpenGL ES 3.0 được tích hợp vào Android ver.18 trở đi.

2.2.3.1 Kỹ thuật 2D

Kỹ thuật này đều dùng các thuật toán xử lý đồ họa cơ bản như phép dịch chuyển, phép quay, phép có dãn,…[8].



Hình 2-3: Hình minh họa phép dịch chuyển, phép quay

Các hàm trong OpenGL ES dùng trong Android.

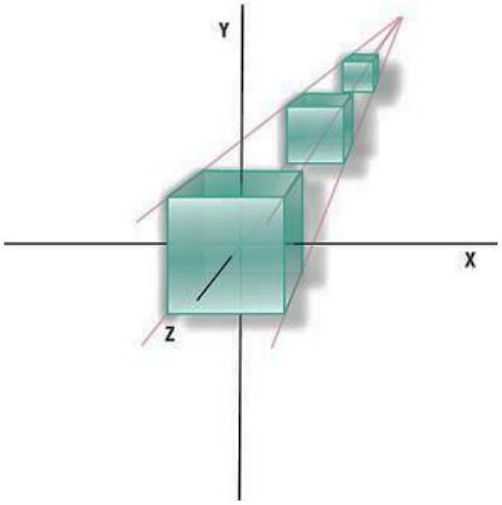
```
import android.graphics

importandroid.graphics.drawable

importandroid.graphics.drawable.shapes

importandroid.graphics.pdf
```

Các hàm trong các thư viện này được sử dụng theo từng yêu cầu bài toán và được tra cứu trong [7,8].

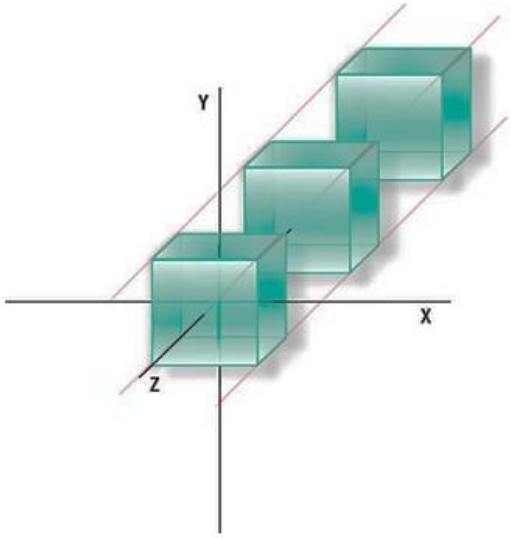
2.2.3.2 Kỹ thuật 3D

Kỹ thuật 3D là kỹ thuật tiên tiến trong lĩnh vực xử lý đồ họa, các thuật toán để xây dựng các ứng dụng này đòi hỏi nhiều tài nguyên của hệ thống. Do vậy đối với các

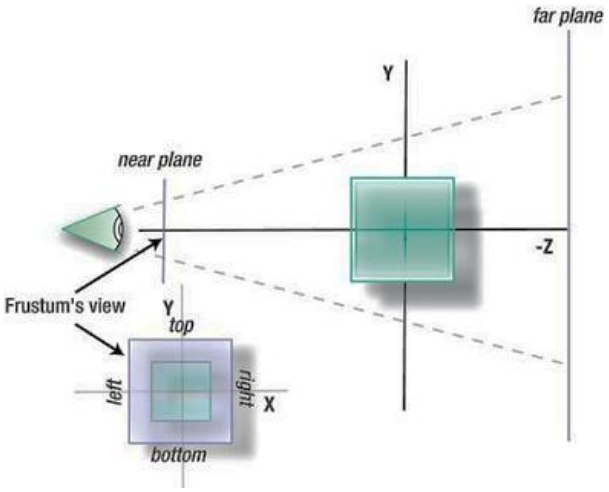
thiết bị di động đôi khi sử dụng kỹ thuật 2D giả lập cho 3D (hay đổ họa 3D – *xyz có chiều z =0*). Các nhà thiết kế phải quy đổi từ hệ tọa độ 3D sang tọa độ của màn hình.



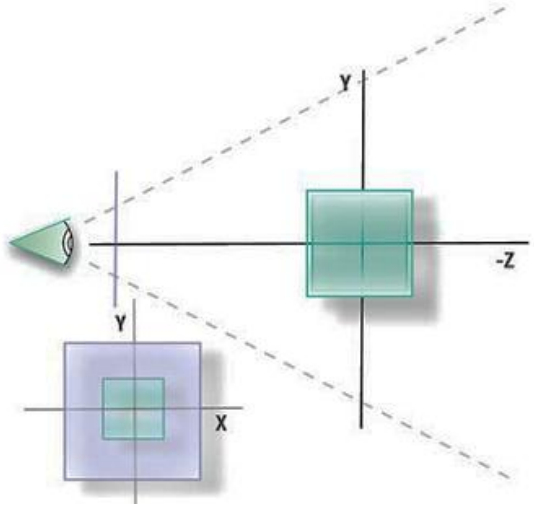
Hình 2-4: Minh họa phép chiếu phối cảnh 3D trên 2D



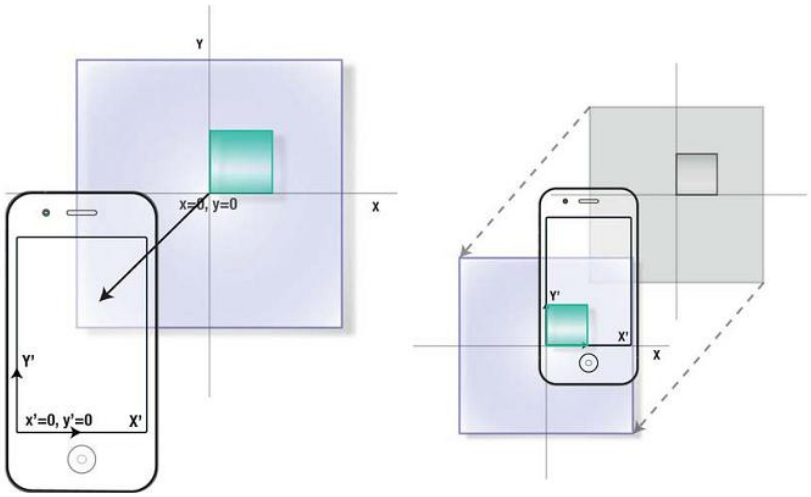
Hình 2-5: Minh họa phép chiếu song song



Hình 2-6: Minh họa góc nhìn hẹp



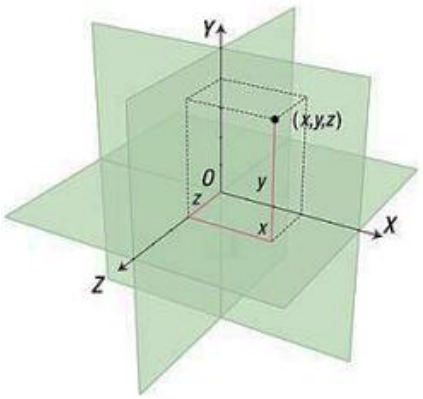
Hình 2-7: Minh họa góc nhìn rộng



Hình 2-8: Minh họa phép chiếu hình ảnh vào thiết bị, hình bên trái chuyển gốc tọa độ vào thiết bị và hình bên phải là dịch chuyển thiết bị về phía hình ảnh.

2.2.3.3 Chuyển từ tọa độ 2D sang 3D

Khi các đối tượng được dịch chuyển từ hệ tọa độ 2D sang 3D, hệ thống cần thực hiện một số thao tác trên đối tượng bằng cách mở rộng chiều thứ 3 trên hệ trục tọa độ 3D.



Hình 2-9: Hình minh họa hệ trục tọa độ Đề-Các 3 chiều

Trên thực tế,cùng một loại dữ liệu mang ra hiển thị, có nhiều cách biểu diễn chúng trên màn hình thiết bị di động.

- Không gian đối tượng, biểu diễn quan hệ giữa các đối tượng với nhau.
- Camera, hoặc mắt, khoảng không, vị trí từ điểm nhìn người dùng.
- Phép chiếu, hoặc phép cắt,hoặc khoảng trống màn hình 2D tính từ ảnh cuối cùng được hiển thị.
- Không gian tuyến tính, được dùng nhiều trong hiệu ứng cao cấp như đường bình đồ, các hình nhấp nhô trong bản đồ,...
  Ví dụ: chuyển đối tượng từ 2D sang 3Dtham khảo [8, p49].

```
float vertices[] =
   {
         -1.0f, -1.0f,
          1.0f, -1.0f,
         -1.0f,  1.0f,
          1.0f,  1.0f
   };
byte maxColor=(byte)255;
byte colors[] =
   {
      maxColor,  maxColor,         0,  maxColor,
             0,  maxColor,  maxColor,  maxColor,
             0,         0,         0,  maxColor,
      maxColor,         0,  maxColor,  maxColor
   };
```

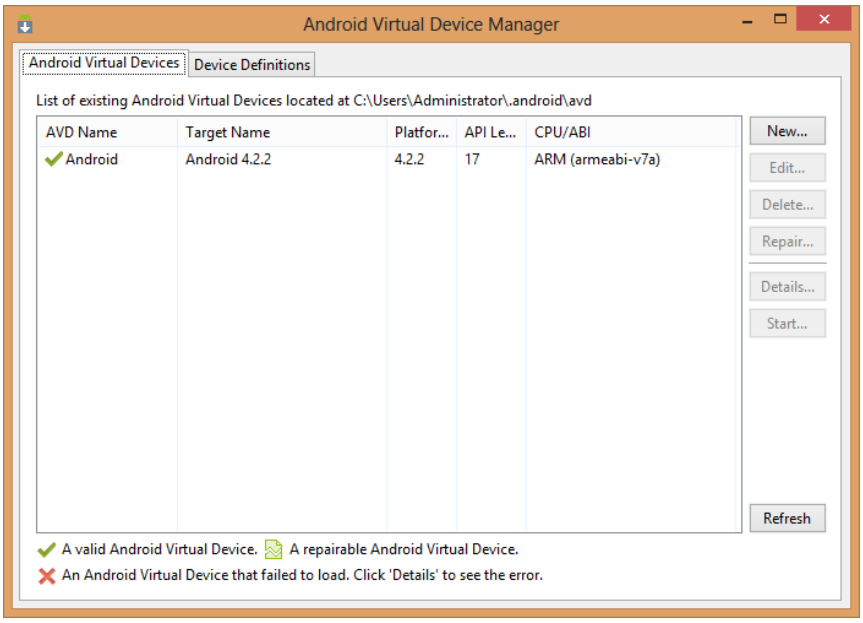
```
byte indices[] =
   {
      0, 3, 1,
      0, 2, 3
   };
```

Mở rộng chiều thứ 3 cho đối tượng, tức là thêm truc trục z vào mảng mô tả đối tượng.

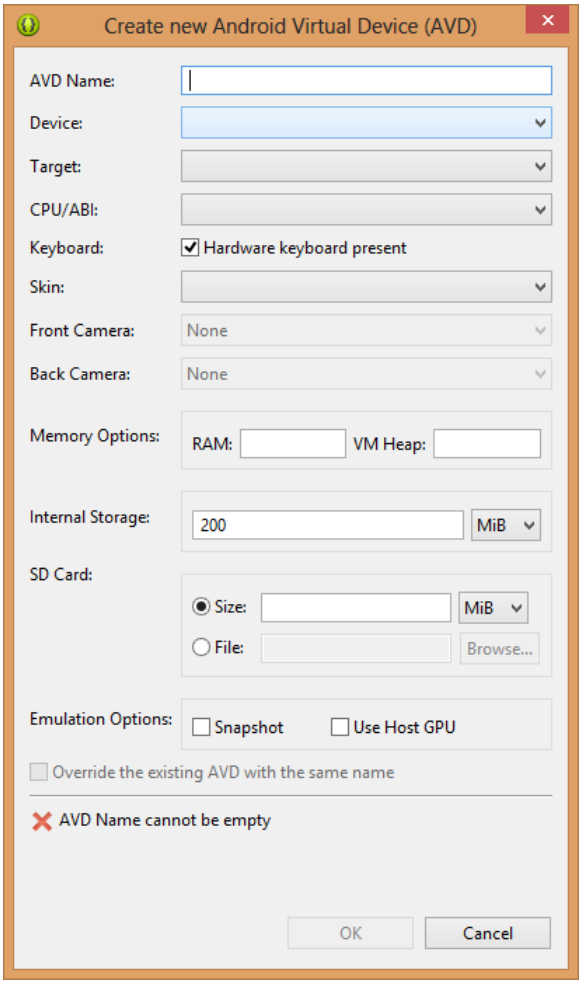
```
float vertices[] =
{
      -1.0f,    1.0f,    1.0f,
       1.0f,    1.0f,    1.0f,
       1.0f,   -1.0f,    1.0f,
      -1.0f,   -1.0f,    1.0f,
      -1.0f,    1.0f,   -1.0f,
       1.0f,    1.0f,   -1.0f,
       1.0f,   -1.0f,   -1.0f,
      -1.0f,   -1.0f,   -1.0f
};
byte maxColor=(byte)255;
byte colors[] =
{
        maxColor,   maxColor,   0,          maxColor,
        0,          maxColor,   maxColor,   maxColor,
        0,          0,          0,          maxColor,
        maxColor,   0,          maxColor,   maxColor,
        maxColor,   0,          0,          maxColor,
        0,          maxColor,   0,          maxColor,
        0,          0,          maxColor,   maxColor,
        0,          0,          0,          maxColor
};
```

2.2.4. Hệ thống mô phỏng

Trong Android SDK có hệ thống mô phỏng giả lập phần cứng gọi là *Android Virtual Device Manager* hệ thống này cho phép giả lập các phần cứng để tương thích với mục đích phát triển phần mềm.



Hình 2-10: Giao diện Android Virtual Device Manager



Hình 2-11: Tạo thiết bị ảo trong Android Virtual Device Manager

Các thông số trong thiết bị ảo được thiết lập nhằm tao ra nền tảng phần cứng phù hợp với yêu cầu phát triển hệ thống. Thông tin cụ thể trong tài liệu [10].

2.2.5. Kỹ thuật xử lý va chạm trong game

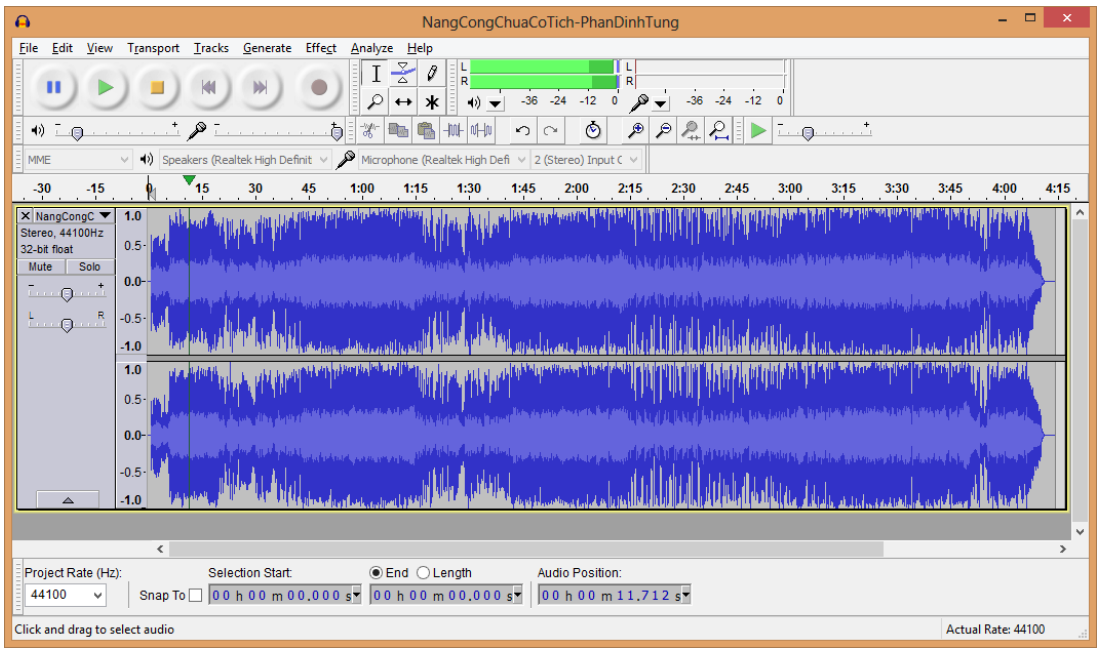
Xử lý va chạm các đối tượng trong game là một phần không thể thiếu đối với các hệ thống tương tác với người dùng. Đặc biệt đối với các trò chơi, xử lý va chạm còn được nâng lên như là một nghệ thuật. Các trò chơi trong thiết bị thông minh có khả năng tương tác với người dùng rất cao. Nhờ có các cảm biến, các nhân vật trong trò chơi có thể phản ứng lại tác động của môi trường như, độ nghiêng, lắc, va đập, quay, … Điều này cho thấy các trò chơi trong thiết bị di động rất thông minh.

Xử lý va chạm được lập trình thành các thư viện đặt trong các *game engine*.Các tình huống va chạm của các vật thể được phân thành từng nhóm, nhóm các đối tượng 2D như các hình vẽ, ảnh, đường thẳng, mặt phẳng,…; nhóm các đối tượng 3D như các vật thể, mặt cong đồi, núi, tòa nhà,… đều được cụ thể hóa thành các phương thức, từ đó các lập trình viên chỉ việc gọi ra sử dụng và gắn các vật thể muốn xử lý vào các phương thức đó.

Ví dụ: Trong *game engine**AndEngine*** các nhà phát triển quy ước mọi vật thể đều được đặt trong đối tượng Box2D từ đó họ xây dựng mọi phương thức xử lý cho đối tượng Box2D này. Như vậy dù đặt vật thể nào vào trong Box2D cũng đều được xử lý một cách đơn giản.

2.3 CÔNG CỤ XỬ LÝ ÂM THANH

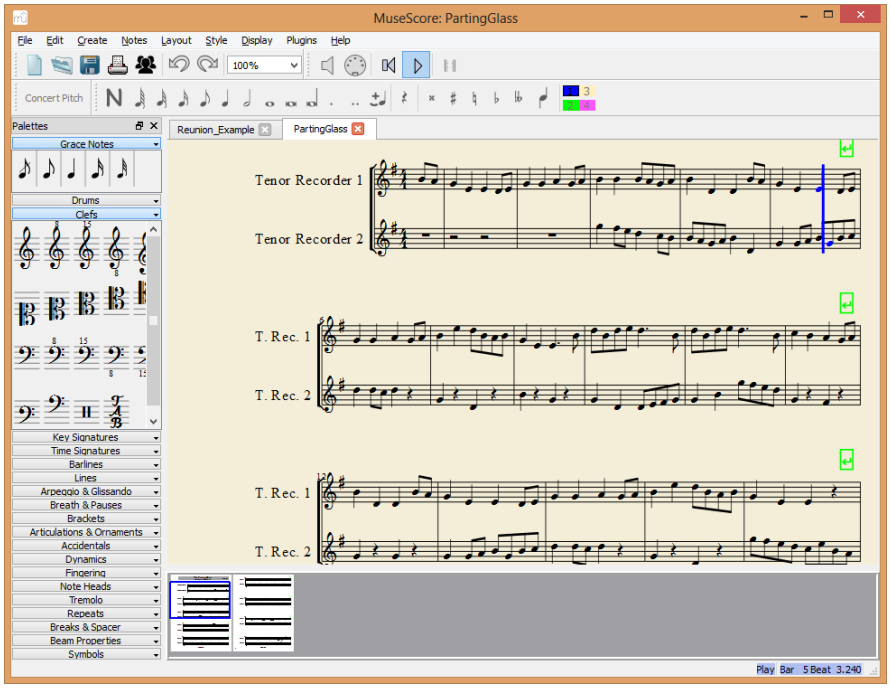
Các loại định dạng âm thanh sử dụng trong trò chơi trên thiết bị di động cũng là các loại định dạng dùng trong các thiết bị khác. Do đó không có sự cách biệt nào liên quan đến âm thanh trên thiết bị di động. Khi xây dựng chương trình trò chơi cho các thiết bị di động. Các nhà phát triển thường dụng một số công cụ mã nguồn mở điển hình là phần mềm Audacity.



Hình 2-12: Giao diện chương trình Audacity

Phần mềm này cho phép biên tập âm thanh tùy yêu cầu của chương trình trò chơi. Phân tách âm thanh và điều chỉnh biên độ, ghi âm, …

Phần mềm MuseScore cho phép biên tập âm thanh theo dạng bản nhạc rất đa dạng và có thể chuyển thành các tệp âm thanh sử dụng trong các chương trình trò chơi.

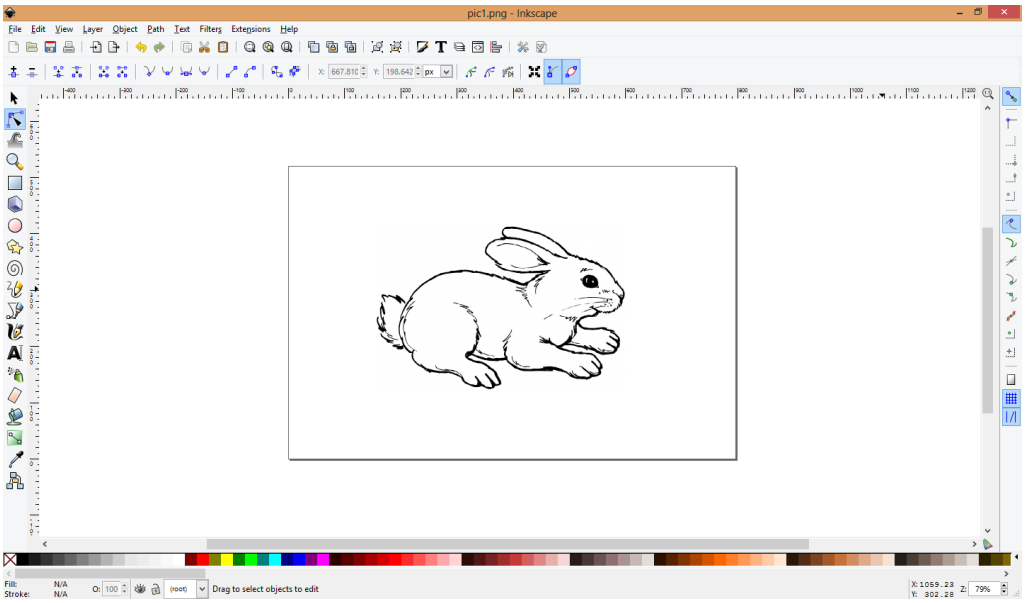


Hình 2-13: Giao diện chương trình MuseScore

2.4 CÔNG CỤ XỬ LÝ HÌNH ẢNH

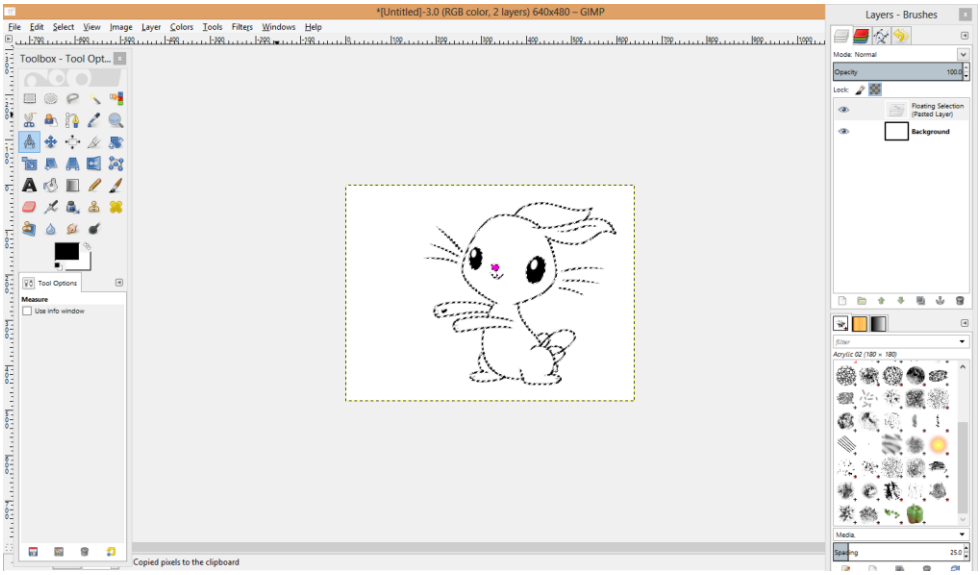
Các công cụ biên tập hình và ảnh rất quan trọng trong quá trình xây dựng các ứng dụng trò chơi. Tùy vào mục đích, chương trình biên tập ảnh và đồ họa được chọn.

Đối với đồ họa vécto, chương trình *Inkscape*được sử dụng biên tập hình, ảnh và cho chất lượng tốt.



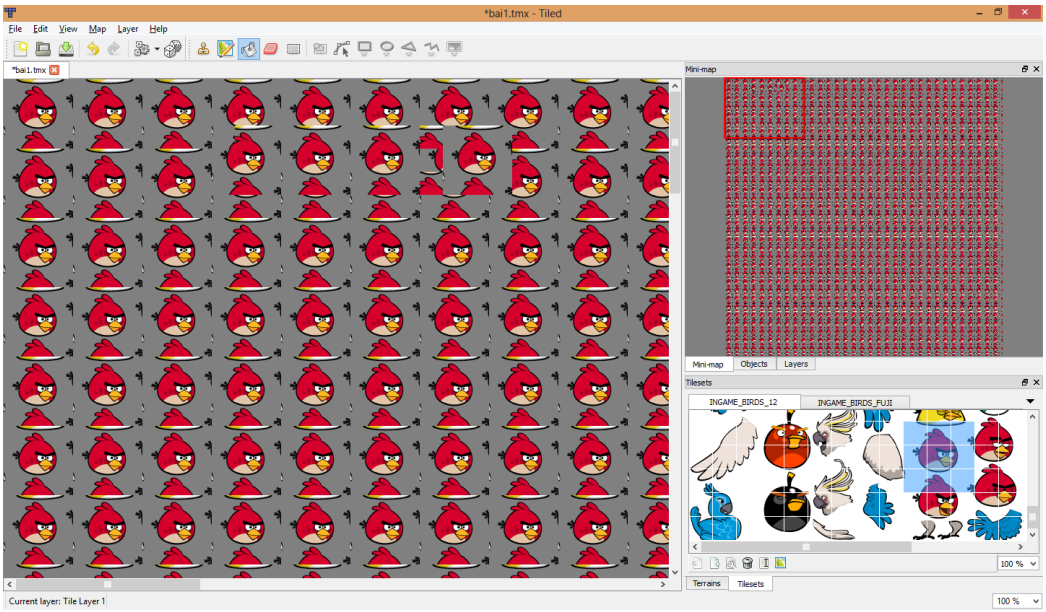
Hình 2-14: Giao diện chương trình InkSpcace

Khi cần ghi các hình ảnh động, sử dụng chương trình *GIMP*,ngoài ra còn một số chương trình trợ giúp tạo các hoạt hình 3D như: 3Ds Max, Poser, và Maya (đây là các phần mềm thương mại).



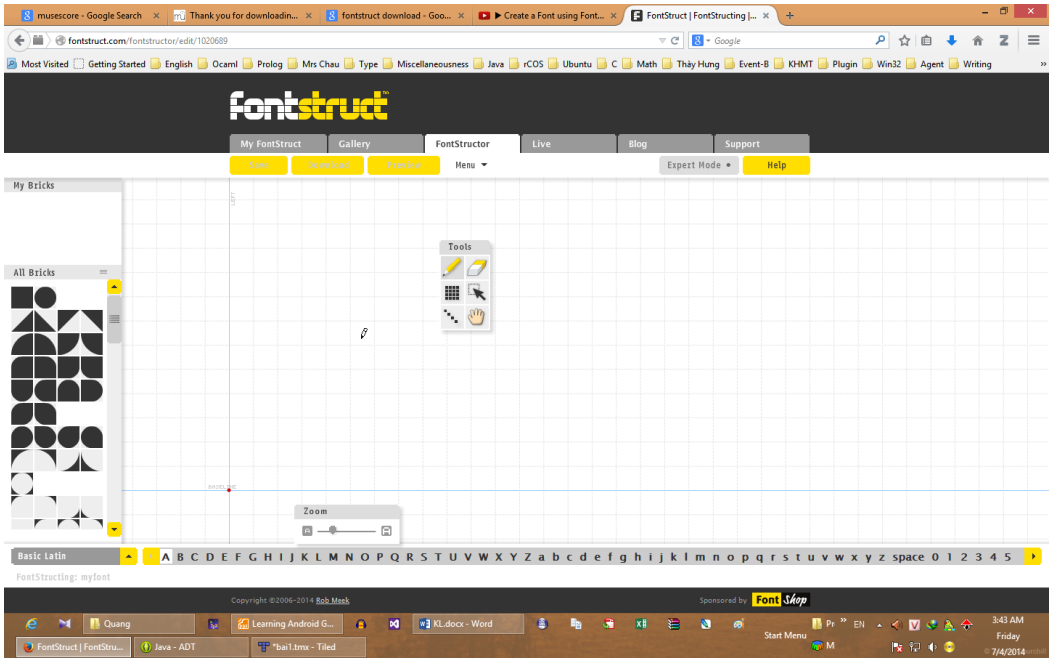
Hình 2-15: Giao diện chương trình GIMP

Chương trình tạo môi trường cho các trò chơi *Tiled*. Ứng này thực sự hữu ích khi phần mềm trò chơi cần môi trường cho các nhân vật trong trò chơi hoạt động.



Hình 2-16: Giao diện chương trình tạo nền cho các trò chơi.

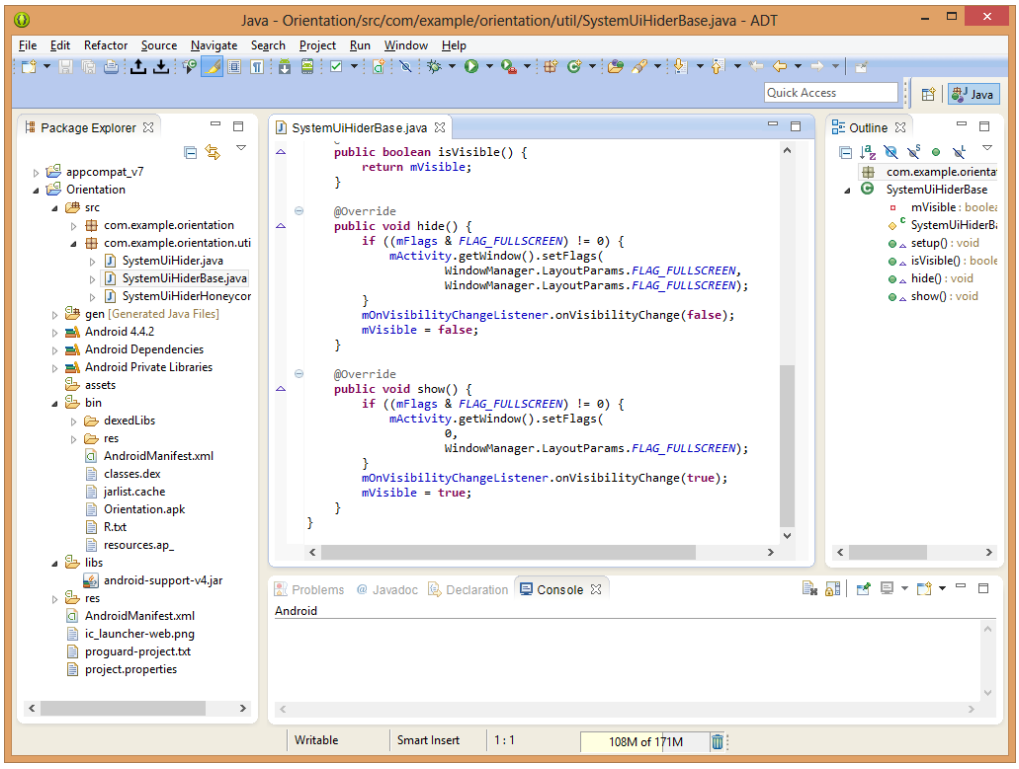
Chương trình sửa phông chữ, *FontStruct* cho phép các nhà phát triển tạo ra những font mới cho ứng dụng của họ.



Hình 2-17: Giao diện chương trình Fontstruct online

2.5 CÔNG CỤ PHÁT TRIỂN PHẦN MỀM

Công cụ phát triển phần mềm trên Android nói chung là khá phổ biến: Eclipse, Netbean, … Tuy nhiên trong khóa luận này đề cấp đến công cụ chuẩn do Google phát hành đó là bộ công cụ *Eclipse* được tích hợp ADT (*Android Development Tool*) cùng với Android SDK Manager do Google phát hành.



Hình 2-18: Giao diện Eclipse

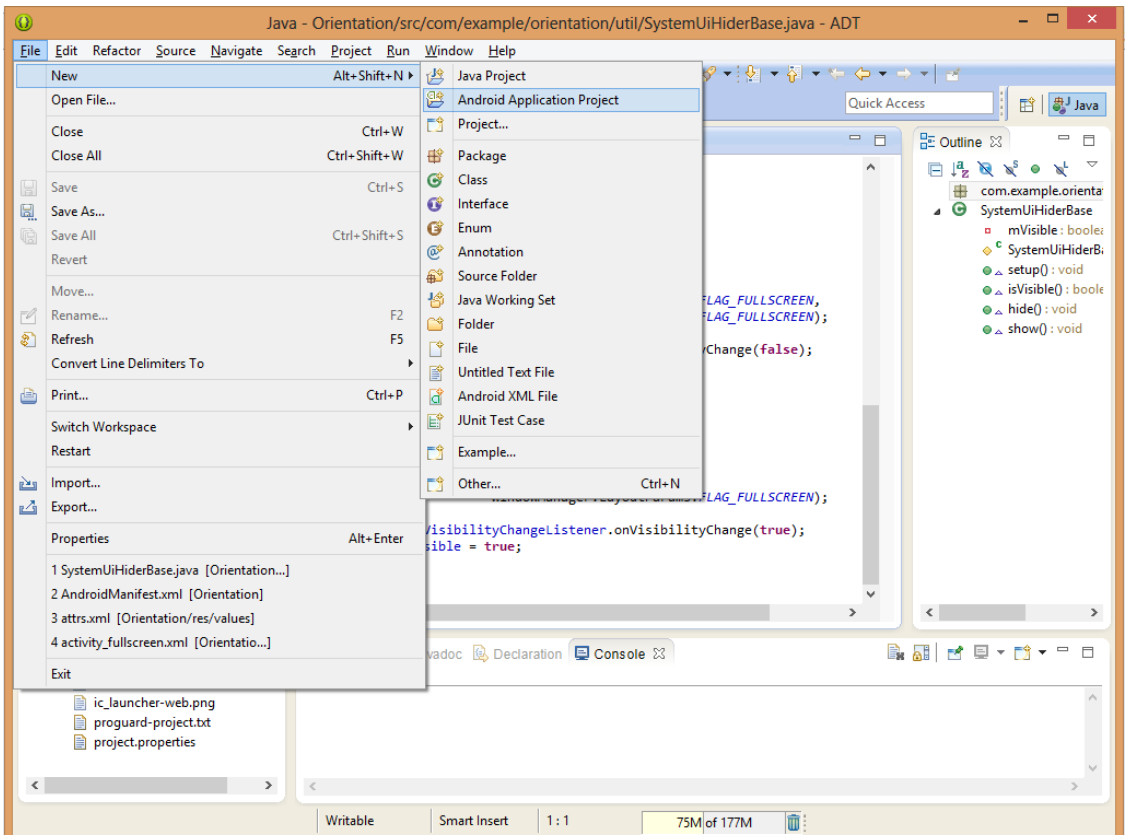
Trong phần này khóa luận mô tả một số bước tạo một dự án trò chơi trên thiết bị di động.

Bước 1: Cài đặt

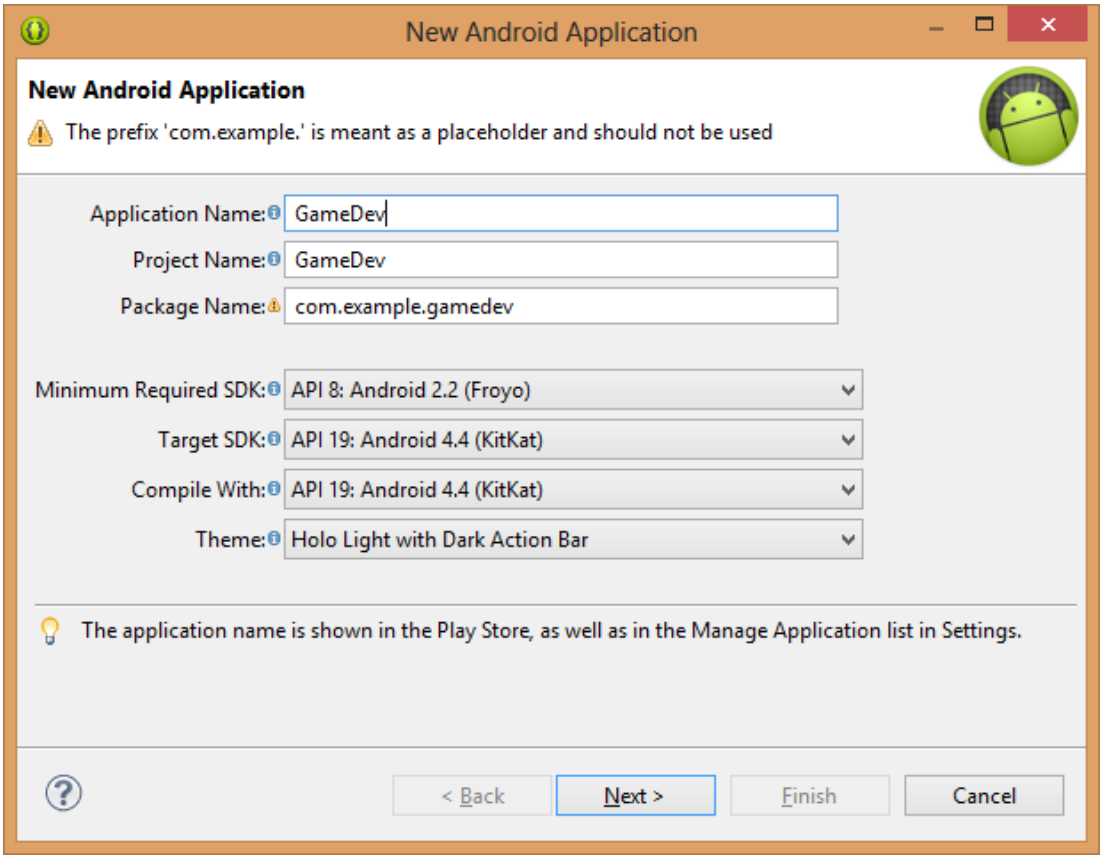
Google cung cấp công cụ phát triển phần mềm cho Android. Các nhà phát triển vào địa chỉ sau để lấy về và giải nén là có thể dùng được ngay.

`http://developer.android.com/sdk/index.html`

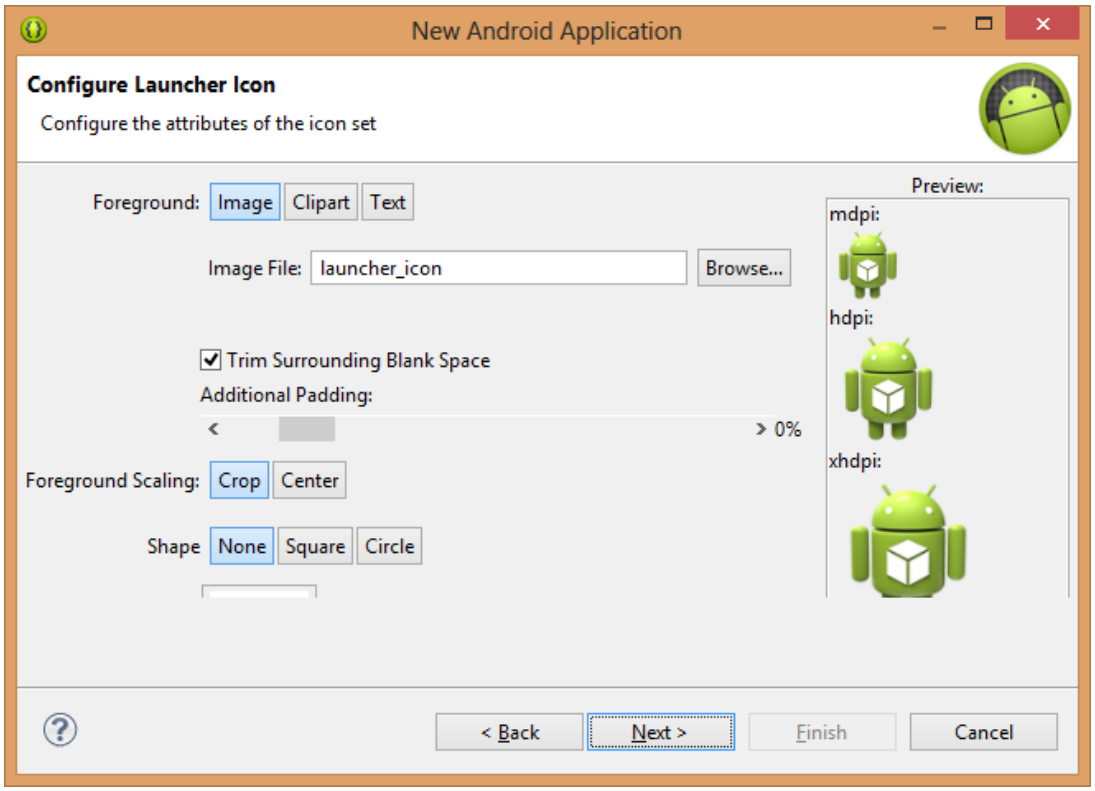
Bước 2: Tạo dự án



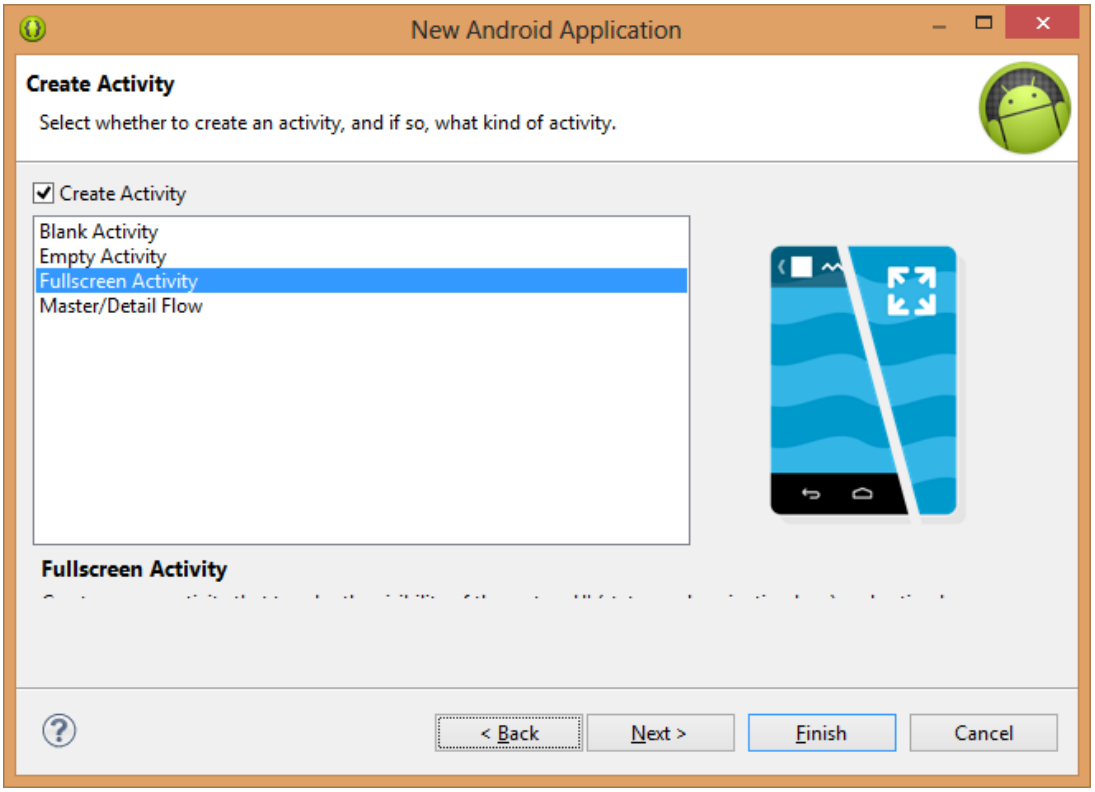
Hình 2-19: Chọn Menu để tạo dự án



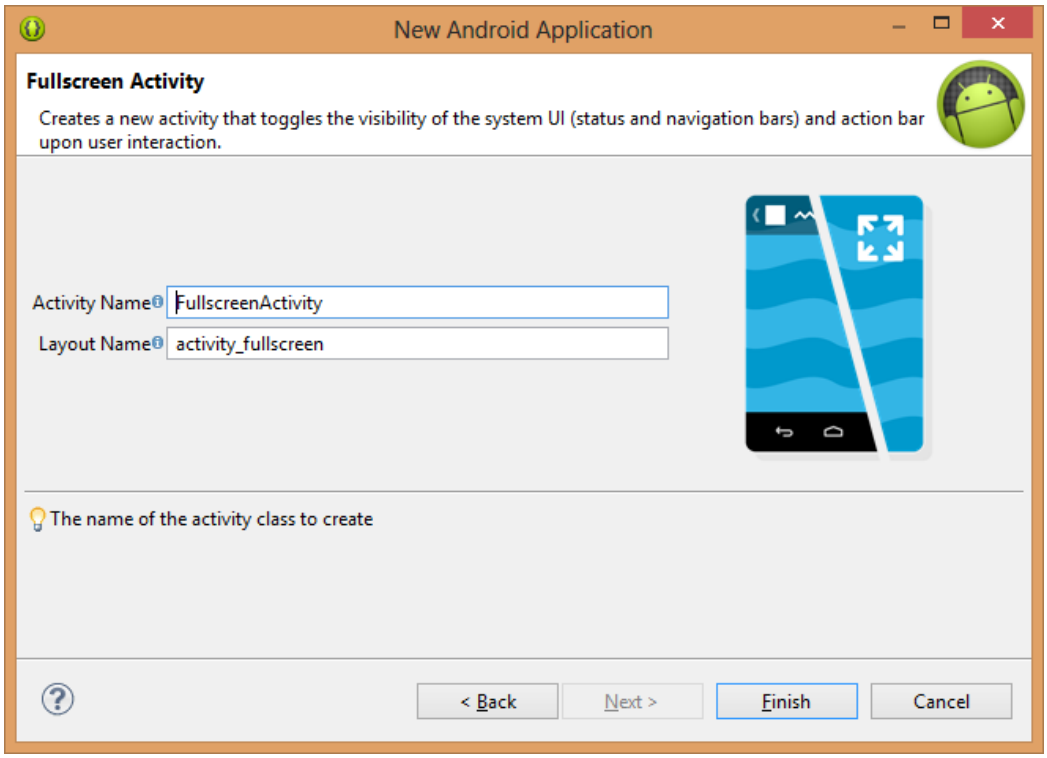
Hình 2-20: Nhập thông tin cho dự án



Hình 2-21: Thiết lập thêm các thông số cho ứng dụng



Hình 2-22: Chọn chế độ hiển thị



Hình 2-23: Hoàn thành tạo dự án

CHƯƠNG 3: TRIỂN KHAI ỨNG DỤNG

Trong chương này khóa luận minh họa một ứng dụng trò chơi bắn máy bay trên hệ điều hành Android.

3.1 CHUẨN BỊ TÀI NGUYÊN CHO ỨNG DỤNG

3.1.1. Ý tưởng của trò chơi

Trò chơi xảy ra trong bối cảnh có một máy bay bảo vệ bầu trời, nếu có máy bay của kẻ thù bay vào vùng trời thì máy bay sẽ bắn, nếu trúng mục tiêu thì máy bay kia sẽ bị cháy (biến mất).

Các chức năng điều khiển gồm: bốn phím mũi tên điều khiển máy bay di chuyển. Phím cách trống (SpaceBar) dùng để bắn. Chương trình có nhạc nền trong suốt quá trình trò chơi hoạt động. Khi đạn được bắn ra thì có âm thanh của đạn bắn.

3.1.2. Đồ họa

Trong trò chơi cần có một số hình ảnh, trong đó:

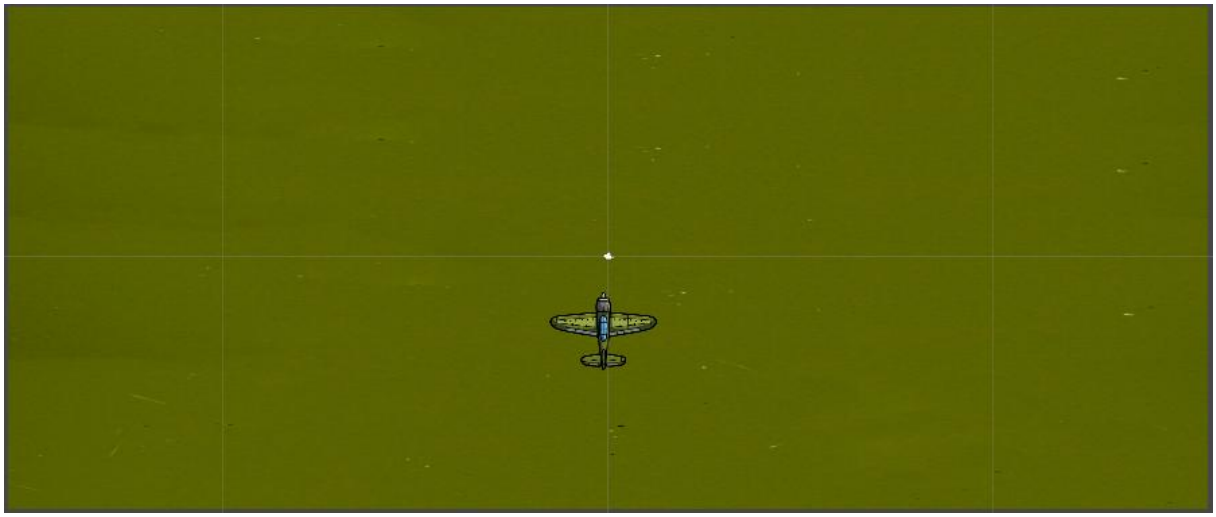
- Hình ảnh máy bay canh giữ bầu trời
- Hình ảnh máy bay địch
- Hình ảnh viên đạn
- Hình ảnh bầu trời

Hình đạn

Hình máy bay địch

Hình vụ nổ

Hình ảnh máy bay canh giữ bầu trời



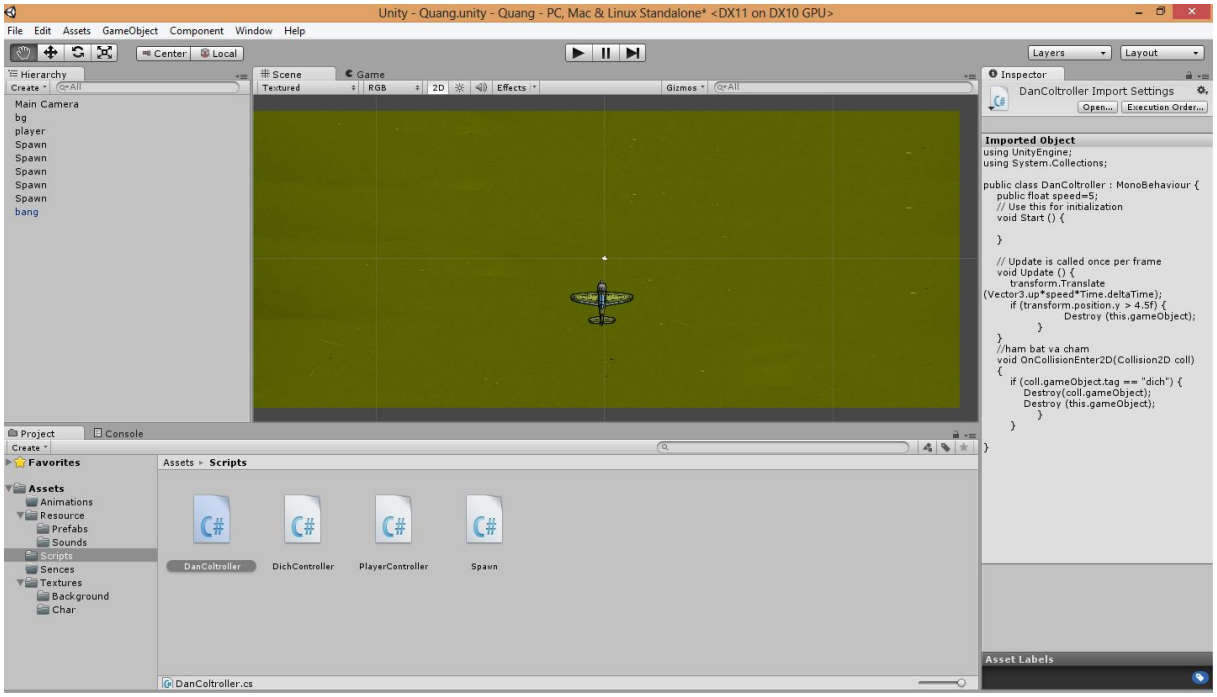
Hình 3-1: Hình ảnh máy bay trên bầu trời

3.1.3. Âm thanh

Âm thanh là một phần không thể thiếu được trong các trò chơi, âm thanh đóng vai trò cảnh báo các tình huống, tạo cảm xúc cho người chơi. Trong ứng dụng này có hai âm thanh. Âm thanh nền mô tả hoạt động của chương trình. Âm thanh thứ hai mô tả tình huống máy bay bắn đạn.

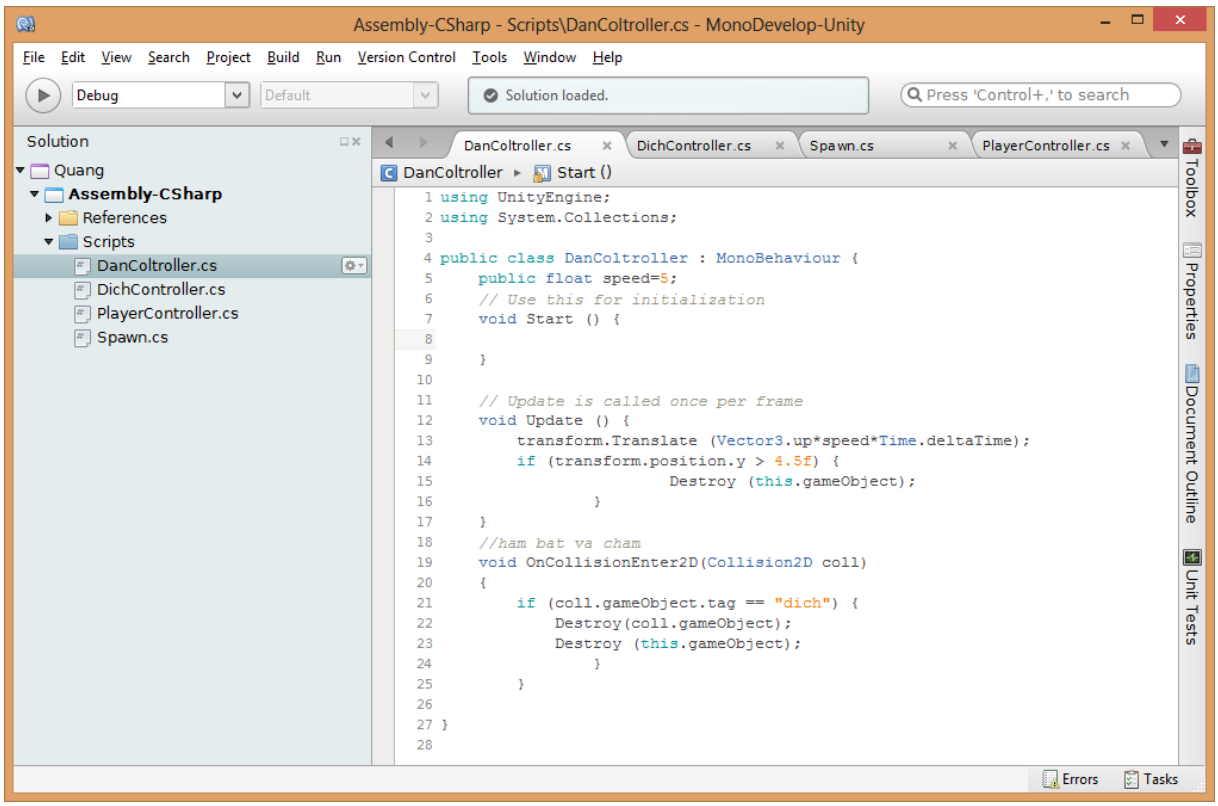
3.2 THỰC NGHIỆM

Hiện nay, để xây dựng chương trình trò chơi trên thiết bị di động, các hãng dùng phần mềm *Game Engine* như Unity3D, AndEngine,… Trong thực nghiệm này khóa luận dùng Game Engine Unity3D để xây dựng chương trình.



Hình 3-2: Màn hình làm việc cảu Unity3D

Unity3D tích hợp công cụ lập trình tên là MonoDevelop cho phép lập trình viên lập trình bằng ngôn ngữ thuộc họ Visual Studio như C#, C/C++,… sử dụng các thư viện trong *.NET* và *Mono*(một Framework như .NET, cho phép chạy trên Linux, MacOS X, Windows).

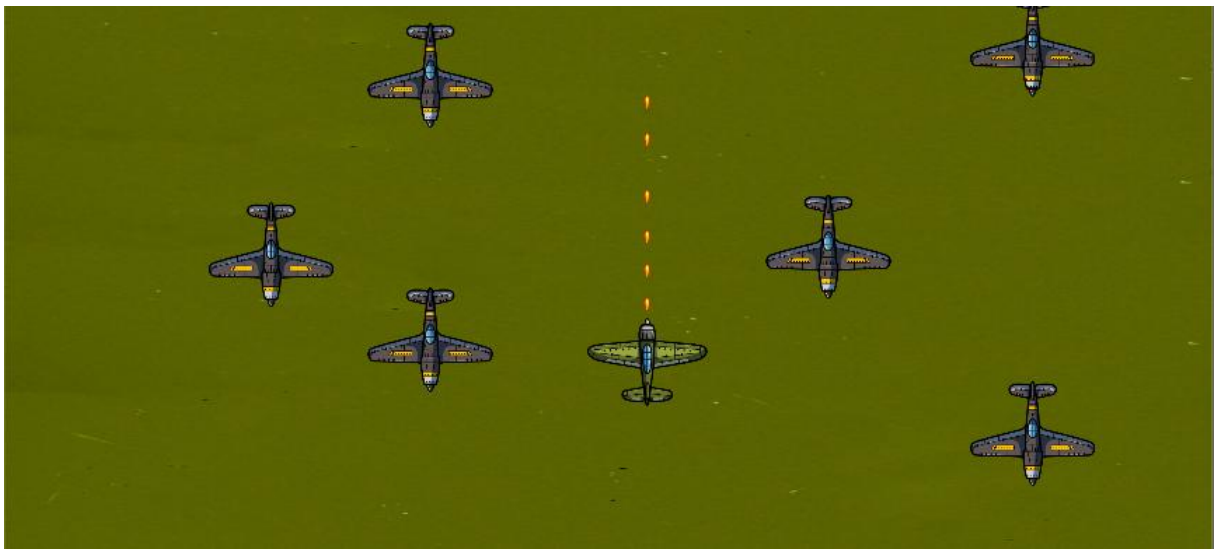


Hình 3-3: Màn hình làm việc của MonoDevelop

Muốn lập trình cho các sự kiện trong trò chơi, MonoDevelop cho phép tạo các *component*cho các sự kiện trong trò chơi. Lập trình viên lập trình như trong Visual Studio.

Sau đây là một số hình ảnh của trò chơi:

Hình 3-4: Hình ảnh khi máy bay địch tấn công



Hình 3-5: Máy bay bắn đạn

Hình 3-6: Hình ảnh vừa rẽ sang trái vừa bắn

KẾT LUẬN

Khóa luận tốt nghiệp đã trình bày về đề tài lập trình trò chơi trên thiết bị di động. Đây là chủ đề mới, là một trong những hướng phát triển đem lại nhiều thành quả trong lĩnh vực công nghệ thông tin. Khóa luận đã đạt được một số kết quả như sau:

- *Trình bày được kiến trúc cơ bản và nguyên lý hoạt động của các bộ phận trên thiết bị di động.*
- *Trình bày kỹ thuật áp dụng trong lập trình trên thiết bị di động, từ kiến trúc cơ bản của một chương trình trên nền hệ điều hành Android đến các công cụ hỗ trợ để lập trình trò chơi.*
- *Phần cuối là chương trình thử nghiệm với trò chơi bắn máy bay.*

TÀI LIỆU THAM KHẢO

[1] http://www.neowin.net/news/guide-to-smartphone-hardware-17-processors

[2] http://vnreview.vn

[3] http://www.ibm.com/developerworks/vn/library/os-android-devel/

[4] http://en.wikipedia.org/wiki/Android_software_development#Android_SDK

[5] http://obviam.net/index.php/table-of-contents/

[6] http://developer.android.com/reference/android/graphics/package-summary.html

[7] Prateek Mehta, *Learn OpenGL ES For Mobile Game and Graphics Development*, Apress, 2013, 209p.

[8] Mike Smithwick, Mayank Verma, *Pro OpenGL ES for Android*,Apress, 2012, 309p.

[9] Rick Rogers, *Learning Android Game Programming*, Addison-Wesley, 2012, 476p.

[10] Mario Zechner, Robert Green, *Beginning Android 4 Games Development*, Apress, 2011, 685p.

[11] Marko Gargenta, *Learning Android*, Oreilly Pub, 2011, 268 p.