

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**



ISO 9001:2015

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

Sinh viên : Nguyễn Xuân Việt
Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn

HẢI PHÒNG - 2019

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

**XÂY DỰNG ỨNG DỤNG GIẢI TRÍ
TRÊN THIẾT BỊ DI ĐỘNG ANDROID**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Nguyễn Xuân Việt
Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn**

HẢI PHÒNG - 2019

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên: Nguyễn Xuân Việt

Mã SV: 1412101130

Lớp: CT1801

Ngành: Công nghệ thông tin

Tên đề tài: Xây dựng ứng dụng giải trí trên thiết bị di động Android

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung:

- Tìm hiểu hệ điều hành android.
- Tìm hiểu môi trường lập trình Android Studio.
- Tìm hiểu các kỹ thuật liên quan đến xây dựng ứng dụng giải trí trên điện thoại android.
- Xây dựng chương trình giải trí chạy trên điện thoại android thật

b. Các yêu cầu cần giải quyết

- Sử dụng công cụ Android Studio để viết chương trình chạy trên điện thoại android
- Xây dựng được chương trình giải trí, đóng gói chương trình thành file *.apk cho phép cài đặt và chạy trên điện thoại android thật.

2. Các số liệu cần thiết để thiết kế, tính toán

- Sử dụng số liệu giả lập

3. Địa điểm thực tập

Trường Đại học Dân lập Hải Phòng

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: Phùng Anh Tuấn

Học hàm, học vị: Thạc sỹ

Cơ quan công tác: Trường Đại học Dân lập Hải Phòng

Nội dung hướng dẫn:

- Tìm hiểu hệ điều hành android.
- Tìm hiểu môi trường lập trình Android Studio.
- Tìm hiểu các kỹ thuật liên quan đến xây dựng ứng dụng giải trí trên điện thoại android.
- Xây dựng chương trình giải trí chạy trên điện thoại android thật.

Người hướng dẫn thứ hai:

Họ và tên:

Học hàm, học vị.....

Cơ quan công tác:

Nội dung hướng dẫn:

.....
.....
.....
.....

Đề tài tốt nghiệp được giao ngày 15 tháng 10 năm 2018

Yêu cầu phải hoàn thành trước ngày 07 tháng 01 năm 2019

Đã nhận nhiệm vụ: Đ.T.T.N
Sinh viên

Đã nhận nhiệm vụ: Đ.T.T.N
Cán bộ hướng dẫn Đ.T.T.N

Hải Phòng, ngàytháng.....năm 2019

Hiệu trưởng

GS.TS.NGŨT Trần Hữu Nghị

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: Phùng Anh Tuấn.

Đơn vị công tác: Trường Đại học Dân lập Hải Phòng.

Họ và tên sinh viên: Nguyễn Xuân Việt Ngành: Công nghệ thông tin

Nội dung hướng dẫn:

- Tìm hiểu hệ điều hành android
- Tìm hiểu môi trường lập trình Android Studio
- Tìm hiểu các kỹ thuật liên quan đến xây dựng ứng dụng giải trí trên điện thoại android.
- Xây dựng chương trình giải trí chạy trên điện thoại android thật.

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

- Có nghiên cứu tài liệu phục vụ cho nội dung đề án.
- Khả năng làm việc độc lập còn nhiều hạn chế.
- Chưa nghiêm túc tuân thủ lịch làm việc với cán bộ hướng dẫn

2. Đánh giá chất lượng của đề án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T. T.N trên các mặt lý luận, thực tiễn, tính toán số liệu...)

- Đã thực hiện được các nội dung đề ra trong đề cương.
- Nội dung đề án có tính thực tế.
- Cần trình bày nội dung đề án logic hơn.

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Đạt

Không đạt

Điểm:.....

Hải Phòng, ngày tháng 01 năm 2019

Giảng viên hướng dẫn

(Ký và ghi rõ họ tên)

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên:

Đơn vị công tác:

Họ và tên sinh viên: Ngành:

Đề tài tốt nghiệp:

.....

1. Phần nhận xét của giảng viên chấm phản biện

.....
.....
.....
.....
.....
.....
.....
.....

2. Những mặt còn hạn chế

.....
.....
.....
.....
.....
.....
.....

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ Không được bảo vệ Điểm:.....

Hải Phòng, ngày tháng 01 năm 2019

Giảng viên chấm phản biện

(Ký và ghi rõ họ tên)

MỤC LỤC

LỜI CẢM ON	1
LỜI MỞ ĐẦU	2
CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID	3
1.1. Giới thiệu hệ điều hành Android	3
1.2. Lịch sử phát triển	3
1.3. Kiến trúc cơ bản của hệ điều hành Android	6
1.3.1. Nhân Linux	6
1.3.2. Thư viện	6
1.3.3. Thực thi	7
1.3.4. Nền tảng Android	7
1.3.5. Tầng ứng dụng.....	8
1.4. Giao diện hệ điều hành android	8
1.5. Quá trình phát triển android	9
1.5.1. Phiên bản Android 1.5	11
1.5.2. Phiên bản Android 1.6	12
1.5.3. Phiên bản Android 2.0 – 2.1	13
1.5.4. Phiên bản Android 2.2	14
1.5.5. Phiên bản Android 2.3	16
1.5.6. Phiên bản Android 3.0	17
1.5.7. Phiên bản Android 4.0	18
1.5.8. Phiên bản Android 4.1	19
1.5.9. Phiên bản Android 4.4	21
1.5.10. Phiên bản Android 5.0 – 5.1.1	22
1.5.11. Phiên bản Android 6.0	23
1.5.12. Phiên bản Android 7.0	25
1.5.13. Phiên bản Android 8.0 – 8.1	26
CHƯƠNG 2: MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG ANDROID STUDIO	27
2.1. Sơ lược về Android Studio	27
2.1.1. Cài đặt Android Studio	27

2.1.2. Máy ảo Genymotion	31
2.2. Thành phần trong một dự án Android.....	34
2.3. Thành phần giao diện	35
2.3.1. View Group.....	35
2.3.2. View	37
2.4. Bắt và xử lý sự kiện trên giao diện.....	38
2.5. Thành phần trong ứng dụng Android	40
2.5.1. Vòng đời hoạt động	40
2.5.2. Services	41
2.5.3. BroadcastReceiver	41
2.5.4. Content providers.....	41
2.6. Một số công cụ để phát triển ứng dụng	41
2.6.1. Giới thiệu về Seekbar	41
2.6.2. Giới thiệu về Checkbox.....	44
2.6.3. Giới thiệu về ImageButton	45
2.6.4. Giới thiệu về CountdownTimer.....	46
CHƯƠNG 3: CHƯƠNG TRÌNH THỰC NGHIỆM.....	48
3.1. Phát biểu bài toán.....	48
3.2. Sơ đồ chức năng.....	48
3.3. Kỹ thuật lập trình Cuộc Đua Kỳ Thú	48
3.3.1. Tạo màn hình giao diện trò chơi	49
3.3.2. Tạo menu giao diện chọn màn đua và nhân vật.....	51
3.3.3. Kỹ thuật cho 3 đường đua chạy ngẫu nhiên.	54
3.4. Các giao diện chương trình.....	56
3.4.1. Giao diện chính.....	56
3.4.2. Giao diện menu chính.....	56
3.4.3. Giao diện khi chạy đua	60
KẾT LUẬN	62
TÀI LIỆU THAM KHẢO	63

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến quý thầy cô Trường Đại Học Dân Lập Hải Phòng, những người đã dìu dắt em tận tình, đã truyền đạt cho em những kiến thức và bài học quý báu trong suốt thời gian em theo học tại trường.

Em xin trân trọng gửi lời cảm ơn đến tất cả các thầy cô trong khoa Công Nghệ Thông Tin, đặc biệt là thầy giáo ThS. Phùng Anh Tuấn, thầy đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình làm tốt nghiệp. Với sự chỉ bảo của thầy, em đã có những định hướng tốt trong việc triển khai và thực hiện các yêu cầu trong quá trình làm đồ án tốt nghiệp. Em xin chân thành cảm ơn!

Hải Phòng, ngày.....tháng.....năm 2019

Sinh viên

Nguyễn Xuân Việt

LỜI MỞ ĐẦU

Hiện nay Công nghệ thông tin vô cùng phát triển thì mọi người đều sử dụng máy vi tính hoặc điện thoại di động để làm việc và giải trí. Do đó việc xây dựng các ứng dụng cho điện thoại di động đang là một ngành công nghiệp mới đầy tiềm năng và hứa hẹn nhiều sự phát triển vượt bậc của ngành khoa học kỹ thuật.

Phần mềm, ứng dụng cho điện thoại di động hiện nay rất đa dạng và phong phú trên các hệ điều hành di động. Các hệ điều hành J2ME, Adroid, IOS, Hybrid, Web bases Mobile Application đã rất phát triển trên thị trường truyền thông di động.

Trong vài năm trở lại đây, hệ điều hành Adroid ra đời với sự kế thừa những ưu việt của các hệ điều hành ra đời trước và sự kết hợp của nhiều công nghệ tiên tiến nhất hiện nay. Adroid đã nhanh chóng là đối thủ cạnh tranh mạnh mẽ với các hệ điều hành trước đó và đang là hệ điều hành di động của tương lai và được nhiều người ưa chuộng nhất.

Ngày nay, với sự phát triển nhanh chóng của xã hội, nhu cầu giải trí thông qua điện thoại di động ngày càng phổ biến, vì vậy em đã chọn đề tài “**Xây dựng ứng dụng giải trí trên thiết bị di động android**” với mục đích nghiên cứu, tìm hiểu về ứng dụng trên android để có thể đáp ứng nhu cầu giải trí đó, giúp cho mọi người có thể thư giãn thông qua ứng dụng.

CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID

1.1. Giới thiệu hệ điều hành Android

Android là một hệ điều hành dựa trên nền tảng Linux[1], được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005.

Chính mã nguồn mở của Android cùng với tính không ràng buộc nhiều đã cho phép các nhà phát triển thiết bị di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị.

Nhờ yếu tố mở, dễ dàng tinh chỉnh cùng sự phát triển nhanh chóng đã khiến hệ điều hành này dần trở nên phổ biến, kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng nhưng giờ đây Android đã xuất hiện trên các smart TV, máy chơi game và một số thiết bị điện tử khác.

Android bắt đầu với bản beta đầu tiên vào tháng 11 năm 2007 và phiên bản thương mại đầu tiên, Android 1.0, được phát hành vào tháng 9 năm 2008. Kể từ tháng 4 năm 2009, phiên bản Android được phát triển, đặt tên theo chủ đề bánh kẹo và phát hành theo thứ tự bảng chữ cái: Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, Kitkat, Lollipop, Marshmallow, Nougat, và bây giờ là Oreo.

Kỷ nguyên của Android chính thức bắt đầu vào ngày 22 tháng 10 năm 2008, khi chiếc điện thoại T-Mobile G1 bắt đầu được bán ra tại Mỹ. Vào thời gian đầu, rất nhiều tính năng cơ bản bị thiếu sót như: bàn phím ảo, cảm ứng đa điểm và tính năng mua ứng dụng vẫn chưa xuất hiện. Tuy nhiên, một số tính năng cũng như giao diện đặc sản của hệ điều hành này đã khởi nguồn từ chiếc G1 và trở thành những yếu tố không thể thiếu trên Android sau này.[1]

1.2. Lịch sử phát triển

Tổng công ty Android (Android, Inc.) được thành lập tại Palo Alto, California vào tháng 10 năm 2003 bởi Andy Rubin (đồng sáng lập công ty Danger), Rich Miner (đồng sáng lập Tổng công ty Viễn thông Wildfire), Nick Sears (từng là Phó

giám đốc T-Mobile), và Chris White (trưởng thiết kế và giao diện tại WebTV) để phát triển, theo lời của Rubin, "các thiết bị di động thông minh hơn có thể biết được vị trí và sở thích của người dùng". Dù những người thành lập và nhân viên đều là những người có tiếng tăm, Tổng công ty Android hoạt động một cách âm thầm, chỉ tiết lộ rằng họ đang làm phần mềm dành cho điện thoại di động. Trong năm đó, Rubin hết kinh phí. Steve Perlman, một người bạn thân của Rubin, mang cho ông 10.000 USD tiền mặt nhưng từ chối tham gia vào công ty.

Google mua lại Tổng công ty Android vào ngày 17 tháng 8 năm 2005, biến nó thành một bộ phận trực thuộc Google. Những nhân viên của chủ chốt của Tổng công ty Android, gồm Rubin, Miner và White, vẫn tiếp tục ở lại công ty làm việc sau thương vụ này. Vào thời điểm đó không có nhiều thông tin về Tổng công ty, nhưng nhiều người đồn đoán rằng Google dự tính tham gia thị trường điện thoại di động sau bước đi này. Tại Google, nhóm do Rubin đứng đầu đã phát triển một nền tảng thiết bị di động phát triển trên nền nhân Linux. Google quảng bá nền tảng này cho các nhà sản xuất điện thoại và các nhà mạng với lời hứa sẽ cung cấp một hệ thống uyển chuyển và có khả năng nâng cấp. Google đã liên hệ với hàng loạt hãng phần cứng cũng như đối tác phần mềm, bắn tin cho các nhà mạng rằng họ sẵn sàng hợp tác với các cấp độ khác nhau.

Ngày càng nhiều suy đoán rằng Google sẽ tham gia thị trường điện thoại di động xuất hiện trong tháng 12 năm 2006. Tin tức của BBC và Nhật báo phổ Wall chú thích rằng Google muốn đưa công nghệ tìm kiếm và các ứng dụng của họ vào điện thoại di động và họ đang nỗ lực làm việc để thực hiện điều này. Các phương tiện truyền thông truyền thống lẫn online cũng viết về tin đồn rằng Google đang phát triển một thiết bị cầm tay mang thương hiệu Google. Một vài tờ báo còn nói rằng trong khi Google vẫn đang thực hiện những bản mô tả kỹ thuật chi tiết, họ đã trình diễn sản phẩm mẫu cho các nhà sản xuất điện thoại di động và nhà mạng. Tháng 9 năm 2007, InformationWeek đăng tải một nghiên cứu của Evaluateserve cho biết Google đã nộp một số đơn xin cấp bằng sáng chế trong lĩnh vực điện thoại di động.

Ngày 5 tháng 11 năm 2007, Liên minh thiết bị cầm tay mở (Open Handset Alliance), một hiệp hội bao gồm nhiều công ty trong đó có Texas Instruments, Tập

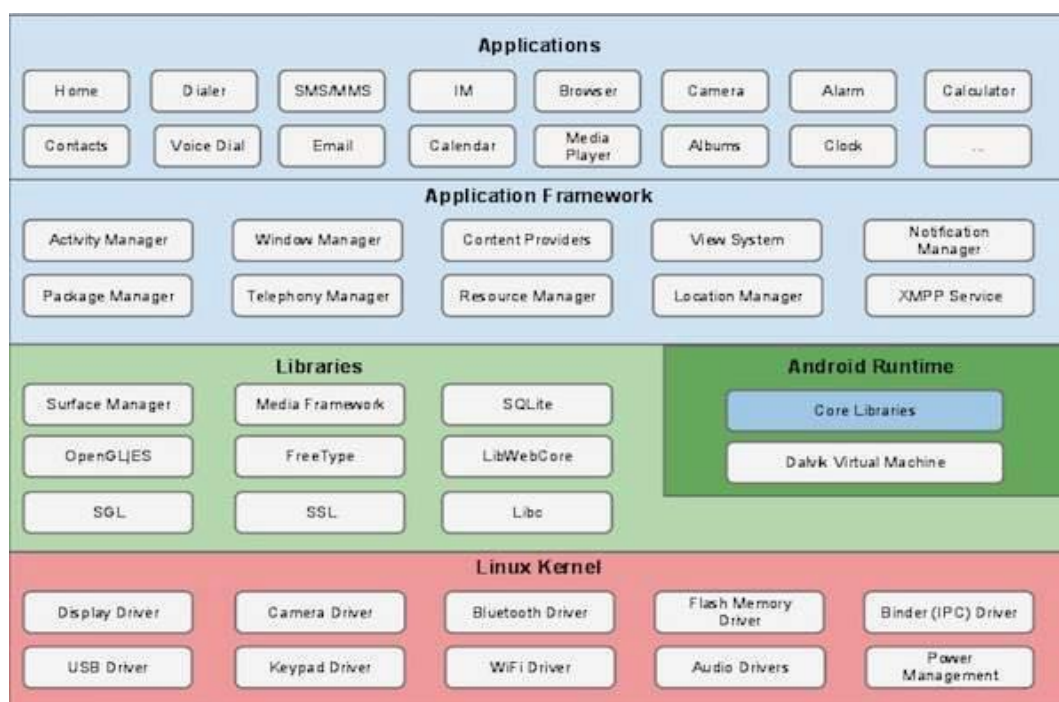
đoàn Broadcom, Google, HTC, Intel, LG, Tập đoàn Marvell Technology, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel và T-Mobile được thành lập với mục đích phát triển các tiêu chuẩn mở cho thiết bị di động. Cùng ngày, Android cũng được ra mắt với vai trò là sản phẩm đầu tiên của Liên minh, một nền tảng thiết bị di động được xây dựng trên nhân Linux phiên bản 2.6. Chiếc điện thoại chạy Android đầu tiên được bán ra là HTC Dream, phát hành ngày 22 tháng 10 năm 2008. Biểu trưng của hệ điều hành Android mới là một con rôbốt màu xanh lá cây do hãng thiết kế Irina Blok tại California vẽ.

Từ năm 2008, Android đã trải qua nhiều lần cập nhật để dần dần cải tiến hệ điều hành, bổ sung các tính năng mới và sửa các lỗi trong những lần phát hành trước. Mỗi bản nâng cấp được đặt tên lần lượt theo thứ tự bảng chữ cái, theo tên của một món ăn tráng miệng; ví dụ như phiên bản 1.5 Cupcake (bánh bông lan nhỏ có kem) tiếp nối bằng phiên bản 1.6 Donut (bánh vòng). Phiên bản mới nhất hiện nay là 5.0 Lollipop. Vào năm 2010, Google ra mắt loạt thiết bị Nexus-một dòng sản phẩm bao gồm điện thoại thông minh và máy tính bảng chạy hệ điều hành Android, do các đối tác phần cứng sản xuất. HTC đã hợp tác với Google trong chiếc điện thoại thông minh Nexus đầu tiên, Nexus One. Kể từ đó nhiều thiết bị mới hơn đã gia nhập vào dòng sản phẩm này, như điện thoại Nexus 4 và máy tính bảng Nexus 10, lần lượt do LG và Samsung sản xuất. Google xem điện thoại và máy tính bảng Nexus là những thiết bị Android chủ lực của mình, với những tính năng phần cứng và phần mềm mới nhất của Android.

Năm 2014, Google công bố Android Wear, hệ điều hành dành cho các thiết bị đeo được.[1]

1.3. Kiến trúc cơ bản của hệ điều hành Android

- Android gồm 5 phần chính sau được chia trong 4 lớp như trong hình 1.3.1 sau đây:



Hình 1.3.1 Kiến trúc hệ điều hành android

1.3.1. Nhân Linux

Android dựa trên Linux phiên bản 2.6 cho hệ thống dịch vụ cốt lõi như security, memory management, process management, network stack, and driver model. Kernel Linux hoạt động như một lớp trừu tượng hóa giữa phần cứng và phần còn lại của phần mềm stack.

1.3.2. Thư viện

Android bao gồm một tập hợp các thư viện C/C++ được sử dụng bởi nhiều thành phần khác nhau trong hệ thống Android. Điều này được thể hiện thông qua nền tảng ứng dụng Android. Một số các thư viện cơ bản được liệt kê dưới đây:

- Hệ thống thư viện C: một BSD có nguồn gốc từ hệ thống thư viện tiêu chuẩn C (libc), điều chỉnh để nhúng vào các thiết bị dựa trên Linux.
- Thư viện Media: dựa trên PacketVideo's OpenCORE; các thư viện hỗ trợ phát lại và ghi âm của âm thanh phổ biến và các định dạng video, cũng như

các tập tin hình ảnh tĩnh, bao gồm cả MPEG4, H.264, MP3, AAC, AMR, JPG, PNG.

- Bề mặt quản lý : Quản lý việc truy xuất vào hệ thống hiển thị.
- LibWebCore: một công cụ trình duyệt web hiện đại mà quyền hạn cả hai trình duyệt web Android và xem web nhúng.
- SGL: Đồ họa 2D cơ bản của máy.
- Thư viện 3D: một thực hiện dựa vào OpenGL ES 1.0 APIs; các thư viện sử dụng phần cứng tăng tốc 3D (nếu có), tối ưu hóa cao rasterizer phần mềm 3D.
- FreeType: vẽ phông chữ bitmap và vector.

1.3.3. Thực thi

Android bao gồm một tập hợp các thư viện cơ bản mà cung cấp hầu hết các chức năng có sẵn trong các thư viện lõi của ngôn ngữ lập trình Java. Tất cả các ứng dụng Android đều chạy trong tiến trình riêng. Máy ảo Dalvik đã được viết để cho một thiết bị có thể chạy nhiều máy ảo hiệu quả. Các VM Dalvik thực thi các tập tin thực thi Dalvik (dex). Định dạng được tối ưu hóa cho bộ nhớ tối thiểu. VM là dựa trên register - based, và chạy các lớp đã được biên dịch bởi một trình biên dịch Java để chuyển đổi thành các định dạng dex. Các VM Dalvik dựa vào nhân Linux cho các chức năng cơ bản như luồng và quản lý bộ nhớ thấp.

1.3.4. Nền tảng Android

Bằng cách cung cấp một nền tảng phát triển mở, Android cung cấp cho các nhà phát triển khả năng xây dựng các ứng dụng cực kỳ phong phú và sáng tạo. Nhà phát triển được tự do tận dụng các thiết bị phần cứng, thông tin địa điểm truy cập, các dịch vụ chạy nền, thiết lập hệ thống báo động, thêm các thông báo để các thanh trạng thái, và nhiều, nhiều hơn nữa. Nhà phát triển có thể truy cập vào các API cùng một khuôn khổ được sử dụng bởi các ứng dụng lõi. Các kiến trúc ứng dụng được thiết kế để đơn giản hóa việc sử dụng lại các thành phần; bất kỳ ứng dụng có thể xuất bản khả năng của và ứng dụng nào khác sau đó có thể sử dụng những khả năng (có thể hạn chế bảo mật được thực thi bởi khuôn khổ). Cơ chế này cho phép các thành phần tương tự sẽ được thay thế bởi người sử dụng.

Cơ bản tất cả các ứng dụng là một bộ các dịch vụ và các hệ thống, bao gồm:

- Một tập hợp rất nhiều các View có khả năng kế thừa lẫn nhau dùng để thiết kế phân giao diện ứng dụng như: gridview, tableview, linearlayout
- Content Provider cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác (chẳng hạn như Contacts) hoặc là chia sẻ dữ liệu giữa các ứng dụng đó.
- Resource Manager cung cấp truy xuất tới các tài nguyên không phải là mã nguồn, chẳng hạn như: localized strings, graphics, and layout files.
- Notification Manager cho phép tất cả các ứng dụng hiển thị các custom alerts trong status bar. Activity Manager được dùng để quản lý chu trình sống của ứng dụng và điều hướng các activity.

1.3.5. Tầng ứng dụng

Tầng ứng dụng (Application) là tầng giao tiếp với người dùng với các thiết bị Android như Danh bạ, tin nhắn, trò chơi, tiện ích tính toán, trình duyệt... Mọi ứng dụng viết đều nằm trên tầng này.

1.4. Giao diện hệ điều hành android

Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo giãn và thu lại để xử lý các đối tượng trên màn hình. Sự phản ứng với tác động của người dùng diễn ra gần như ngay lập tức, nhằm tạo ra giao diện cảm ứng mượt mà, thường dùng tính năng rung của thiết bị để tạo phản hồi rung cho người dùng. Những thiết bị phần cứng bên trong như gia tốc kế, con quay hồi chuyển và cảm biến khoảng cách được một số ứng dụng sử dụng để phản hồi một số hành động khác của người dùng, ví dụ như điều chỉnh màn hình từ chế độ hiển thị dọc sang chế độ hiển thị ngang tùy theo vị trí của thiết bị, hoặc cho phép người dùng lái xe đua bằng xoay thiết bị, giống như đang điều khiển vô-lăng.

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop (bàn làm việc) trên máy tính để bàn. Màn hình chính Android thường gồm nhiều biểu tượng (*icon*) và tiện ích (*widget*); biểu tượng ứng dụng sẽ mở ứng dụng tương ứng, còn

tiện ích hiển thị những nội dung sống động, cập nhật tự động như dự báo thời tiết, hộp thư của người dùng, hoặc những mẫu tin thời sự ngay trên màn hình chính. Màn hình chính có thể gồm nhiều trang xem được bằng cách vuốt ra trước hoặc sau, mặc dù giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích. Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi "chủ đề" của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như Windows Phone chẳng hạn. Phần lớn những nhà sản xuất, và một số nhà mạng, thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh.

Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Thanh trạng thái này có thể "kéo" xuống để xem màn hình thông báo gồm thông tin quan trọng hoặc cập nhật của các ứng dụng, như email hay tin nhắn SMS mới nhận, mà không làm gián đoạn hoặc khiến người dùng cảm thấy bất tiện. Trong các phiên bản đời đầu, người dùng có thể nhấn vào thông báo để mở ra ứng dụng tương ứng, về sau này các thông tin cập nhật được bổ sung thêm tính năng, như có khả năng lập tức gọi ngược lại khi có cuộc gọi nhỡ mà không cần phải mở ứng dụng gọi điện ra. Thông báo sẽ luôn nằm đó cho đến khi người dùng đã đọc hoặc xóa nó đi.

1.5. Quá trình phát triển android

Dường như ở thời điểm hiện tại, nếu phải trả lời câu hỏi rằng đâu là nền tảng xuất hiện nhiều nhất trên các thiết bị từ di động đến Desktop, thì gần như toàn bộ câu trả lời đều sẽ hướng tới một cái tên duy nhất – Android. Sự đa dạng của Android là một điều không thể bàn cãi, khi nó không chỉ dừng lại ở sự phổ cập cho smartphone hay tablet, mà giờ đây, từ Smart TV, tới xe hơi, hay đồng hồ thông minh... chúng ta đều thấy sự có mặt của nền tảng này để nó có thể giao tiếp một cách trực tiếp với người dùng của mình.

Dù vậy, để đạt được một thị phần ấn tượng như vậy, Android đã phải trải qua các thăng trầm trong suốt lịch sử phát triển của mình từ những ngày đầu tiên còn áp ủ như một dự án nền tảng di động từ những năm 2003. Và với sự ra mắt của phiên

bản Android 7.0 Nougat trong thời gian gần đây, chúng ta có thể nhìn qua các bước phát triển của Android thông qua từng phiên bản phát triển tính năng từ Android 1.0 đến nay.

- Android và thời điểm trước khi phân phối ra phiên bản thương mại chính thức như hình 1.5.1 dưới đây:



Hình 1.5.1. Thời điểm ban đầu của android

Vào tháng 10 năm 2003, Android Inc, khi đó được biết đến như là một công ty công nghệ được thành lập bởi Andy Rubin, Rich Miner, Nick Sears và Chris White. Theo Andy Rubin cho biết, mục tiêu của Android Inc lúc bấy giờ không giống như những gì ở thời điểm hiện tại khi mà các nhà sáng lập lúc đó hướng tới việc tạo ra một nền tảng cao cấp nhưng cho các máy ảnh kỹ thuật số, và biến chúng thành một thiết bị thông minh hơn trong việc nhận ra vị trí hiện tại của người dùng.

Tuy nhiên, chỉ trong một thời gian ngắn không lâu sau đó, các mục tiêu này nhanh chóng nhận ra được trở ngại về thị phần khi mà thị trường máy ảnh không chiếm một số lượng đủ lớn để biến nền tảng này thành một công cụ tiềm năng, và cũng chính vì lý do đó, mà Android Inc ngay lập tức hướng sự chú ý của họ vào thị trường béo bở hơn ở mảng smartphone đang lên như điều gặp gió ngay cả khi mà Symbian và Windows Mobile là những nền tảng lớn đã thống trị từ lâu khi nhắc đến điều này.

Ngay cả khi có sự định hướng rõ ràng như vậy, thì với bản thân không hề có một sự vững chãi trong khả năng phát triển, cũng như các vấn đề liên quan đến các nhà tài trợ và nguồn vốn rót vào đã đẩy đến thương vụ mua lại hoàn toàn Android Inc với không khách hàng không ai khác chính là Google vào thời điểm 17/8/2005 với phần lớn đội ngũ phát triển ban đầu của công ty này vẫn được giữ nguyên để tiếp tục công việc trước đó của họ. Sau đó, Android hoạt động như một đội trong số các đội khác nhau của Google dưới sự dẫn dắt của Andy Rubin bắt đầu phát triển một nền tảng mới với phần nhân từ Linux giống như điều họ làm đến thời điểm hiện tại. Và biểu tượng chú robot màu xanh tượng trưng cho Android được thiết kế bởi Irina Blok, và nó gắn liền với nền tảng này từ phiên bản thử nghiệm đầu tiên phát hành vào 5/11/2007.

1.5.1. Phiên bản Android 1.5



Hình 1.5.1.1 Phiên bản Android 1.5

Android 1.5 Cupcake là phiên bản nâng cấp tiếp theo trong hệ sinh thái Android được ra mắt vào ngày 27/4/2009, và đây đồng thời là phiên bản gây sự chú ý nhiều nhất khi nó gần như là bản Android đầu tiên mang tên mã của một món bánh tráng miệng, một trong những đặc trưng cơ bản làm nên thương hiệu của hệ sinh thái dành cho thiết bị di động này. Kể từ đó, các phiên bản Android khác sau

này đều được có một tên mã được xây dựng dựa trên các quy luật được đặt ra bởi trụ sở chính của Google đặt tại Mountain View, California.

Với sự ra mắt của Android 1.5 Cupcake, thì đã có rất nhiều những tính năng mới mẻ khác được đội ngũ phát triển của Google mang vào cho nền tảng của họ, mà không thể nào bỏ qua hệ thống widget được xem như là một phiên bản thu nhỏ của ứng dụng để người dùng có thể tương tác và xử lý ngay trên chính màn hình chính của thiết bị mà không cần phải mở chúng lên. Bên cạnh đó, Android 1.5 Cupcake cũng được bổ sung thêm khả năng xử lý các video định dạng MPEG-4 khá phổ biến tại thời điểm bấy giờ, khả năng tự động kết nối thông qua chuẩn Bluetooth, phát âm thanh chuẩn Stereo, cũng như cho phép người dùng sử dụng hình ảnh để thêm vào trong các số liên lạc yêu thích của mình. [2]

1.5.2. Phiên bản Android 1.6



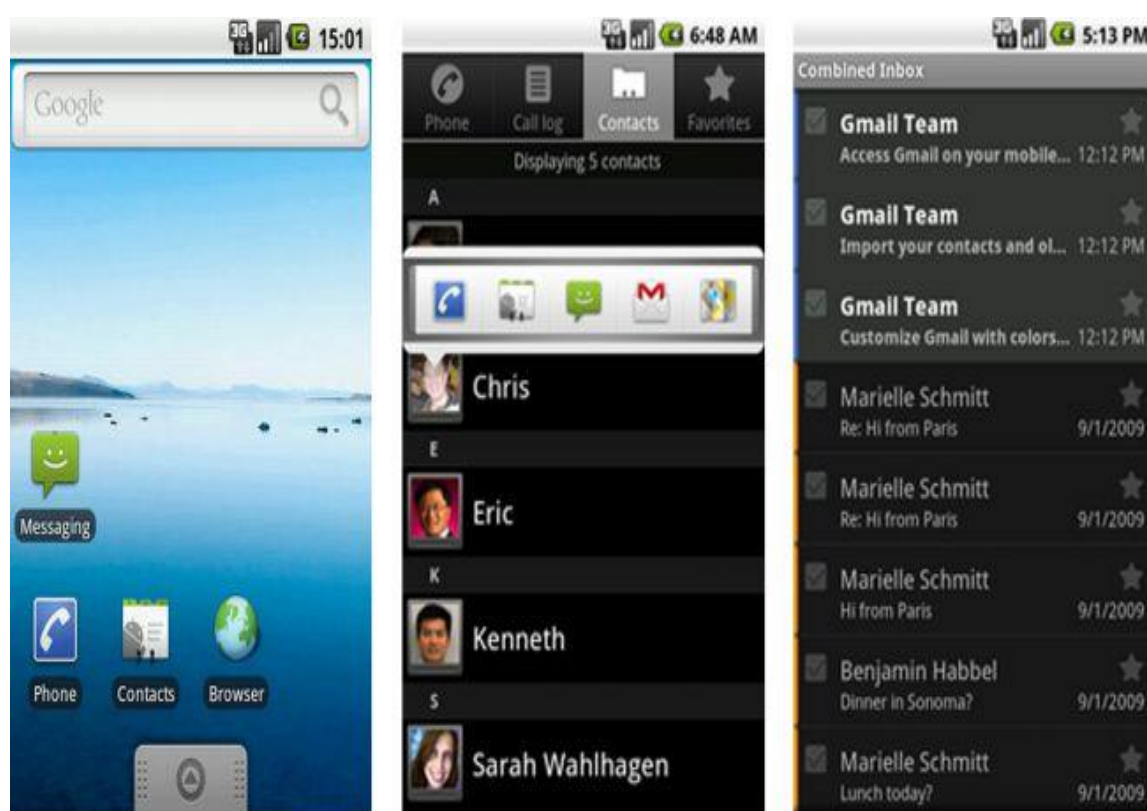
Hình 1.5.2.1. Phiên bản Android 1.6

Chỉ trong vòng chưa đầy 5 tháng sau khi phiên bản Android 1.5 được đưa đến tay người dùng, thì vào ngày 15/9/2009, Google một lần nữa tung ra bản nâng cấp Android 1.6 Donut với khả năng hỗ trợ hiển thị nội dung một cách tốt hơn các

thiết bị với màn hình độ phân giải tiêu chuẩn WVGA (780 x 480 pixels), cũng như các ứng dụng về Gallery và Camera được tích hợp sâu hơn vào trong nền tảng nhằm mang lại một trải nghiệm tốt hơn trong việc chụp và xem chúng với tốc độ nhanh hơn và dễ dàng hơn trước.

Khả năng tìm kiếm bằng kí tự và giọng nói cũng có sự cải tiến trong hệ thống Bookmarks, History, Contacts và nội dung từ Internet, cũng như cho phép đội ngũ phát triển ứng dụng từ nhiều bên có thể thêm các nội dung của họ vào trong các kết quả tìm kiếm được trả về. Điều đó đồng nghĩa với việc Android 1.6 Donut mang đến một cảm giác tốt hơn trong sự gắn kết giữa các bên phát triển nên hệ sinh thái này, ngay cả khi đó là từ Google hay một bên thứ 3 nào khác[2].

1.5.3. Phiên bản Android 2.0 – 2.1



Hình 1.5.3.1. Phiên bản Android 2.0 – 2.1

Ngày 26/10/2009 đón nhận phiên bản phân phối thứ 3 của hệ điều hành Android được ra mắt trong cùng một năm với tên mã Android 2.0 Éclair. Và so với các phiên bản từng được ra mắt trước đây, Android 2.0 Éclair thực sự là một bước nhảy vô cùng lớn của nền tảng này trong việc bổ sung hàng loạt những tính năng vô cùng quan trọng, mà điển hình là khả năng hỗ trợ hệ thống thư điện tử Microsoft

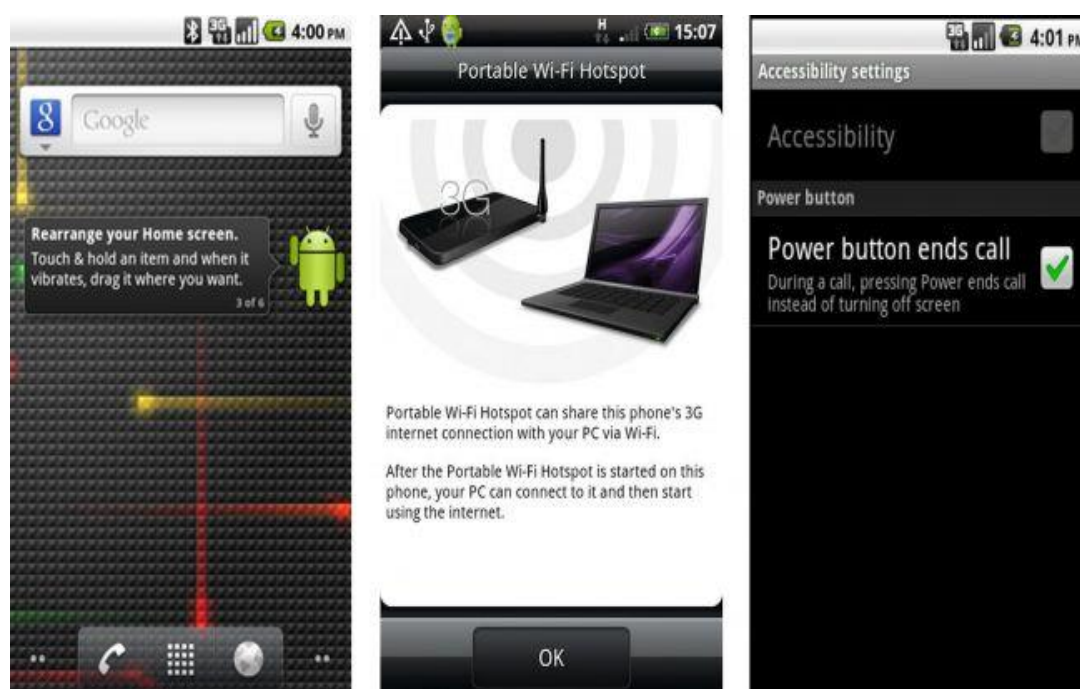
Exchange, bên cạnh việc nâng cấp các thuật toán tìm kiếm để người dùng có thể tìm ra được chính xác thư điện tử mà họ cần giữa nhiều tài khoản khác nhau được hiển thị chung trong cùng một hộp thư đến.

Các tính năng nhắn tin kí tự cũng được cải thiện với việc người dùng giờ đây cũng có thể tìm kiếm được các nội dung được lưu trong SMS và MMS, hay tốc độ nhập dữ liệu được nâng cao hơn thông qua sự cải tiến của hệ thống bàn phím ảo trên màn hình cảm ứng.

Hệ thống camera cũng mang đến nhiều sự hỗ trợ, mà quan trọng là việc cho phép tích hợp thêm sự có mặt của đèn flash, hay chế độ phóng to kỹ thuật số, các hiệu ứng màu và chế độ chụp Macro.

Sau đó, các phiên bản Android 2.0.1 được ra mắt vào 3/9/2009 và Android 2.1 ra mắt vào 12/1/2010 là những phiên bản hướng đến sự cải tiến các tính năng vốn có trên Android 2.0 Éclair và sửa các lỗi hiện có trên nền tảng này theo các báo cáo từ phía người dùng của mình[2].

1.5.4. Phiên bản Android 2.2



Hình 1.5.4.1. Phiên bản Android 2.2

Android 2.2 Froyo chính thức được cập bến các thiết bị di động vào ngày 20/5/2010. Và không giống như nhiều người lầm tưởng, Froyo không hề là một đồ chơi cho trẻ em, mà thay vào đó, Froyo là cụm từ viết tắt của “Frozen Yoghurt”, có

nghĩa là sửa chữa đã trong trường hợp nhiều người thắc mắc là tại sao mà Google không tuân theo quy luật đặt tên cho Android mà họ đã đề ra trước đây.

Phiên bản nâng cấp này hướng đến sự cải thiện lớn trong tốc độ và hiệu năng của toàn bộ hệ thống các thiết bị mà nền tảng này được cài đặt. Các tính năng trong sự liên kết dữ liệu thông qua cổng USB và chế độ Wi-Fi Hotspot cũng chính thức được bổ sung lên các thiết bị di động để biến đây trở thành một trong những bản Android nhận được nhiều sự đón nhận từ phía người dùng của hãng ở thời điểm lúc bấy giờ.

Với Android 2.2 Froyo, các nhà phát triển cũng mang đến cho người dùng sự hạn chế khả năng tải về của người dùng nếu như họ đang sử dụng các kết nối thiết bị di động được cung cấp bởi các nhà mạng, và đó là một trong những tính năng vô cùng quan trọng trong việc giảm thiểu cước di động hàng tháng nếu chẳng may họ đang kết nối 3G và thiết bị đang tải bản cập nhật ứng dụng nào đó. Cũng như bên cạnh đó, các vấn đề bảo mật cũng được quan tâm đến nhiều hơn để mang đến một hệ thống mật khẩu bao gồm cả kí tự và số để mở khóa các nội dung lưu trữ trên thiết bị.

Kho ứng dụng Android Market lúc bấy giờ cũng cải thiện khả năng tương tác với các nhóm ứng dụng được cài đặt, cũng như cải thiện khả năng tự động nâng cấp lên các phiên bản mới hơn trong trường hợp nó sẵn sàng để người dùng tải về, cũng như tự động hỗ trợ các thiết bị có mật độ điểm ảnh cao nhằm mang đến một chất lượng hình ảnh tốt hơn trong trải nghiệm.

Với những tính năng của mình, Android Froyo là một trong số các phiên bản vẫn còn được sử dụng nhiều đến thời điểm ngày nay trên các thiết bị có cấu hình thấp, ngay cả khi thị phần của nó thực sự thấp khi chỉ chiếm đến 0.1%, và Google cũng có những động thái chấm dứt sự hỗ trợ phiên bản này trong một thời gian dài trước đó để bắt buộc người dùng lên các phiên bản cao hơn, nhưng nó không thực sự hiệu quả.[2]

1.5.5. Phiên bản Android 2.3



Hình 1.5.5.1. Phiên bản Android 2.3

Android 2.3 Gingerbread được biết đến như một trong số phiên bản thành công nhất của hệ sinh thái Android được phát hành chính thức vào thời điểm 6/12/2010, với một sự nâng cấp mạnh mẽ trong giao diện người dùng mà chú trọng nhất là tính dễ dàng trong các thao tác sử dụng và tốc độ tương tác với khả năng giảm thiểu tối đa thao tác cảm ứng cần thực hiện. Bàn phím ảo trên màn hình cũng được làm lại để mang đến một cảm quan tốt hơn trong cách thức nhập dữ liệu của người dùng và độ chính xác cũng được cải thiện tốt hơn.

Công nghệ kết nối gần (Near Field Communication – NFC) cũng lần đầu tiên được giới thiệu khả năng hỗ trợ kể từ phiên bản Android 2.3 Gingerbread trở đi, cùng với trình quản lý tải về Download Manager cho phép người dùng có thể tiến trình của các tập tin từ mạng Internet hay Email giờ đây dễ dàng hơn bao giờ hết.

Sự bổ sung khả năng hỗ trợ các dòng cảm biến la bàn số, phong vũ biểu và nhiều loại khác mở ra một tương lai nâng cấp về tính năng tốt hơn trên các ứng dụng phục vụ cho nền tảng, mà trong đó, chủ yếu nhất vẫn là theo dõi vị trí của người dùng, khả năng di chuyển và các hoạt động của họ đối với thiết bị di động đang sử dụng. [2]

1.5.6. Phiên bản Android 3.0



Hình 1.5.6.1. Phiên bản Android 3.0

Android 3.0 Honeycomb là một phiên bản đặc biệt của nền tảng Android được phát hành vào 22/10/2011 khi Google dường như chỉ tập trung vào sự hỗ trợ các thiết bị tablet với phiên bản này, và chỉ trong vòng 2 ngày sau khi nó được ra mắt, thì Motorola Xoom, thiết bị tablet đầu tiên được cài đặt phiên bản này cũng chính thức có mặt trên thị trường.

Mặc dù Android 3.0 Honeycomb là phiên bản chỉ có thể hỗ trợ cho các dòng tablet, nhưng bù lại, hệ thống giao diện người dùng của nó đã đặt ra những bước phát triển đầu tiên cho toàn bộ hệ sinh thái Android sau này. Các thanh thông báo hệ thống giờ đây mang khả năng hiển thị các thông báo cho người dùng, trong khi các nút điều hướng cảm ứng lại nằm ở phía cạnh dưới màn hình. Vào thời điểm đó, các nút điều hướng nay, bao gồm Back, Home, Recent và Applications có thể tìm thấy trên bất kì một thiết bị Android nào, cho phép các nhà sản xuất thiết bị có thể hoàn toàn loại bỏ các phím vật lý trước đây phục vụ cho các tính năng tương tự để mở rộng phần màn hình của họ cho các phím bấm cảm ứng trên.

Một trong những cải tiến khác được nhắc đến trong mặt giao diện người dùng của Android 3.0 Honeycomb là việc hỗ trợ duyệt web đa thẻ, với chế độ

Incognito cho phép người dùng hoàn toàn ẩn danh trước các địa chỉ mà mình truy cập đến và không hề lưu lại lịch sử nhằm bảo vệ tối đa các quyền riêng tư, hay một chế độ xem cao cấp khác dành cho danh bạ và email.

Cũng kể từ Android 3.0 Honeycomb, nó hoàn toàn mở ra khả năng hỗ trợ tốt hơn trên các thiết bị phần cứng cao cấp, khi mà dòng vi xử lý đơn nhân được đưa ra trong thời gian gần đây nhằm đưa smartphone dần thay thế hiệu quả hơn các máy tính truyền thống với hiệu năng vượt trội hơn trong khả năng xử lý các thuật toán về mặt đa nhiệm.

Android 3.1 là phiên bản nâng cấp được giới thiệu không lâu sau đó vào ngày 10/5/2011 với việc hỗ trợ các kết nối với các thiết bị ngoại vi tốt hơn trên cổng giao tiếp USB, trình phát âm thanh chuẩn định dạng FLAC, và khả năng thay đổi kích thước các widget trên màn hình chính để người dùng có thể tiện lợi hơn trong các thao tác sắp xếp của mình. Và sau đó, bản cập nhật khác mới hơn Android 3.2 mang khả năng hỗ trợ tốt hơn các tính năng tương tác với hệ thống Google TV.[2]

1.5.7. Phiên bản Android 4.0



Hình 1.5.7.1 Phiên bản Android 4.0

Android 4.0 Ice Cream Sandwich được phát hành chính thức lần đầu tiên vào ngày 18/10/2011 mang đến sự đồng nhất trong các thiết bị với khả năng tương thích cao cho bất kỳ smartphone hay tablet nào có thể sử dụng Android 2.3 trở về sau. Giao diện người dùng Holo cũng nhận được nhiều sự cải tiến đáng kể, mang đến

cho người dùng một kiểu chữ mới Roboto được sử dụng như kiểu chữ mặc định cho toàn bộ hệ sinh thái Android trong hàng loạt các phiên bản sau đó.

Các thư mục cũng được làm lại nhằm mang đến một sự dễ dàng hơn trong việc tạo mới trên các thao tác kéo thả, các ứng dụng giờ đây cũng có thể nhận được sự truy xuất nhanh ngay từ màn hình khóa, ứng dụng camera được cải thiện trong tốc độ khởi động và xử lý hình ảnh tốt hơn trên cuộc đua độ phân giải với khả năng hỗ trợ chuẩn Full HD 1080p, và trình duyệt web mặc định có thể đồng bộ hóa với các bookmark trên Chrome không chỉ là phiên bản di động, mà còn là đối với phiên bản trên máy tính được đăng nhập sẵn tài khoản Google của người dùng.

Tính năng Face Unlock cũng được xây dựng và ứng dụng để người dùng có thể áp dụng khả năng mở khóa bằng khuôn mặt trên thiết bị, bên cạnh các tính năng bảo mật sẵn có.[2]

1.5.8. Phiên bản Android 4.1



Hình 1.5.8.1. Phiên bản Android 4.1

Ngày 9/7/2012, phiên bản Android 4.1 Jelly Bean chính thức được phân phối đến tay người dùng của mình với sự tập trung trong công việc phát triển phiên bản này hướng đến những trải nghiệm tốt hơn trong giao diện người dùng về cả hiệu

năng thiết bị lẫn tính năng mà nó cung cấp. Trong đó, sự chú ý lớn nhất mà người ta có thể tìm thấy trên Android 4.1 Jelly Bean dường như là sự lãng phí quá mức của Google trong các thiết lập về đồ họa bên cạnh phần nền tảng mặc định của họ, với các công nghệ và thuật toán để đảm bảo rằng màn hình hoạt động ở mức tỉ lệ khung hình ổn định 60fps trong mọi tác vụ để mang đến sự mượt mà cao hơn và thời gian phản hồi nhanh hơn.

Các trải nghiệm âm thanh cũng có sự cải tiến không kém với khả năng hỗ trợ đa kênh cùng lúc, trình phát thông qua ngoại vi với cổng USB. Các ứng dụng Launcher đến từ bên thứ 3 cũng bắt đầu có sự định nghĩa riêng của mình kể từ phiên bản này nhằm mang đến khả năng hỗ trợ tốt hơn trong các tính năng mà mỗi launcher mang lại, cũng như một cái nhìn khác hoàn toàn mới lạ hơn những gì mặc định mà họ vốn có.

Phiên bản nâng cấp Android 4.2 cũng không mất quá lâu kể từ ngày mà Android 4.1 ra mắt để được phát hành một cách rộng rãi tới tay các người dùng của mình. Ngày 13/11/2012 là thời điểm đánh dấu sự hỗ trợ của các widget tới màn hình khóa của thiết bị, cũng như cho phép người dùng có thể mở nhanh ứng dụng camera ngay cả từ màn hình khóa để chụp hình nhanh hơn thay cho việc bắt buộc phải khởi động từ màn hình chính như trước đây. Chế độ Quick Settings cũng được thêm vào trong việc cho phép người dùng thay đổi các cài đặt cơ bản của hệ thống một cách nhanh chóng, chế độ bảo vệ màn hình Daydream cũng được thêm vào, và đi kèm với đó là khả năng hỗ trợ đa tài khoản mở rộng trên tablet thay vì đơn thuần với smartphone.

Android 4.3 là phiên bản khác mang cùng tên mã Ice Cream Sandwich được ra mắt vào 24/7/2013 với sự bổ sung phần lớn các tính năng bao gồm khả năng hỗ trợ Open GL ES 3.0 trong việc mang đến một trải nghiệm đồ họa ấn tượng hơn cho ứng dụng và đặc biệt là trò chơi, hay tính năng tiết kiệm pin tối đa trong giao tiếp trên chuẩn Bluetooth, hỗ trợ đa dạng hơn các gói ngôn ngữ và tương thích với màn hình chuẩn độ phân giải 4K còn khá mới mẻ ở thời điểm hiện tại.[2]

1.5.9. Phiên bản Android 4.4



Hình 1.5.9.1 Phiên bản Android 4.4

Mặc dù Android K từng được Google kì vọng sẽ đặt tên Key Lime Pie cho phiên bản tiếp theo của nền tảng của mình, nhưng đến thời điểm 31/10/2013, thì Android 4.4 lại ra mắt với một biệt danh hoàn toàn mới với tên gọi KitKat, đồng thời đi kèm với đó là thiết bị Nexus 5 đầu tiên được cài đặt phiên bản này cũng được ra mắt trong cùng thời điểm.

Các sự tùy chỉnh mà Google mang đến cho Android 4.4 KitKat thực sự mang đến cho phiên bản này một cái nhìn hoàn toàn mới trong mặt giao diện người dùng, tuy nhiên đó không phải là các điểm nhấn duy nhất mà phiên bản này mang lại cho các khách hàng của họ. Chế độ Immersive Mode cũng được bổ sung kể từ phiên bản Android 4.4 KitKat cho phép các ứng dụng khởi động làm ẩn đi các phím bấm cảm ứng trên màn hình hay thanh trạng thái khi nó không thực sự cần đến, cho phép các ứng dụng giờ đây có thể mở rộng ra toàn bộ màn hình để mang đến một trải nghiệm liền mạch và đầy đủ hơn. Và nếu trong trường hợp bạn cần đến các phím cảm ứng đó, bạn có thể vuốt màn hình từ trên xuống dưới hoặc từ dưới lên trên để mang nó hiển thị trở lại trước khi ẩn đi một lần nữa sau một thời gian không còn tương tác đến nếu như bạn vẫn đang ở trong màn hình làm việc của một ứng dụng nào đó

Các tính năng liên quan đến thẻ NFC cũng được mở rộng trong Android 4.4 KitKat sau khi tích hợp công nghệ này vào trong các phiên bản trước đó, dù rằng nó

thực sự không gây quá nhiều sự ấn tượng với người dùng. Nhưng điều đó cũng không thể phủ nhận công dụng mà tính năng này mang lại khi người dùng có thể sử dụng các thẻ NFC và ghi nội dung lên đó để sử dụng chúng trong các giao dịch, hoặc trao đổi các nội dung liên quan đến danh bạ cho mọi người xung quanh.[2]

1.5.10. Phiên bản Android 5.0 – 5.1.1



Hình 1.5.10.1. Phiên bản Android 5.0 – 5.1.1

Android 5.0 Lollipop, mặc dù được giới thiệu từ khá sớm vào ngày 25/6/2014 trong khuôn khổ sự kiện Google I/O thường niên của Google, nhưng phải mãi đến thời điểm 12/11/2014, thì phiên bản này mới được chính thức phân phối tới tay người dùng để họ có thể cập nhật trên các thiết bị sẵn có của mình.

Một lần nữa, giao diện là vấn đề trọng tâm nhận được nhiều sự chú ý nhất từ phía đội ngũ phát triển của Google trong lần ra mắt lần này, với một thiết kế mang xu hướng phẳng hơn đặt nền móng cho chủ đề Material Design sau này. Các thông báo cũng được tối ưu hóa trong việc hỗ trợ hiển thị một cách trực tiếp trên màn hình khóa của thiết bị để người dùng có thể theo dõi một cách trực tiếp mà không cần phải mở khóa chúng lên, dù rằng khả năng tương tác với các widget trên màn hình này cũng bị loại bỏ chính thức kể từ phiên bản này.

Dự án Project Volta cũng được chính thức áp dụng trên phiên bản Android 5.0 Lollipop trong việc cải thiện thời lượng sử dụng pin một cách tốt hơn so với các

phiên bản trước đó, vốn là một vấn đề gây ra nhiều sự nhức nhối trong khoảng thời gian dài trước đó. Không gian lưu trữ trên thẻ nhớ microSD cũng được chia sẻ tốt hơn trong việc cho phép các ứng dụng xuất phát từ các bên thứ 3 có thể mang các dữ liệu của mình lên đó trong khi phần lõi chính ở thời điểm đó vẫn được cố định trong bộ nhớ máy. Tính năng Smart Lock cũng cải thiện tốt hơn hệ thống bảo mật vốn có với sự dễ dàng trong việc khóa và mở khóa thiết bị phụ thuộc vào các điều kiện sẵn có được thiết lập từ phía người dùng trong các lần sử dụng trước đó.

Phiên bản Android 5.1 Lollipop chính thức ra mắt vào 9/3/2015 mang đến hệ thống âm thanh chất lượng cao hơn trong các cuộc gọi thực hiện trên thiết bị, hỗ trợ nhu cầu sử dụng đa SIM trên các thiết bị di động, các cài đặt Wi-Fi và Bluetooth cũng được mang lên thành tùy chỉnh nhanh Quick Settings và hệ thống bảo mật tốt hơn và chính xác hơn.[2]

1.5.11. Phiên bản Android 6.0



Hình 1.5.11.1. Phiên bản Android 6.0

Cũng giống như người tiền nhiệm Android 5.0 Lollipop trước đó của mình, thì Android 6.0 Marshmallow cũng được giới thiệu từ khá sớm trong sự kiện Google I/O vào tháng 5/2015, nhưng mà mãi đến tháng 10 cùng năm, thì phiên bản

này mới chính thức đổ bộ một cách rộng rãi trên nhiều thiết bị khi được phân phối từ phía Google thông qua hệ thống OTA và tới các nhà sản xuất thiết bị.

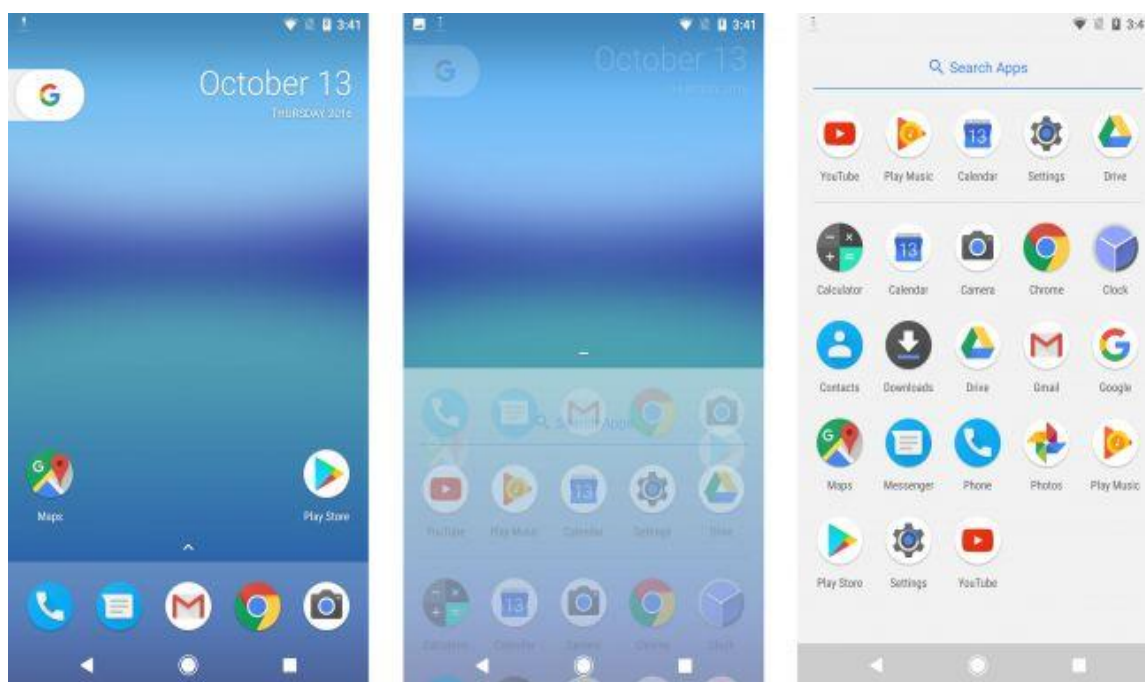
Bản nâng cấp Android 6.0 Marshmallow mang đến cho người dùng một tính năng hoàn toàn mới Now On Tap trong việc cung cấp một cách tốt hơn các kết quả tìm kiếm với Google dựa trên lịch sử tìm kiếm được thực hiện từ trước đó. Cùng với đó, điểm nhấn lớn nhất và được nhiều người dùng đánh giá cao nhất trên phiên bản này là hệ thống quản lý năng lượng tốt hơn được kế thừa và phát triển từ Android 5.0 Lollipop trong những nỗ lực mang đến thời lượng sử dụng dài hơn cho người dùng trên các phần cứng khác nhau mà nhà sản xuất thiết bị mang đến.

Bên cạnh đó, tính năng Adoptable Storage cũng đưa thẻ nhớ microSD trở nên tiện dụng hơn, khi bên cạnh một dung lượng hỗ trợ lớn hơn, người dùng giờ đây có thể sử dụng nó như một công cụ mở rộng cho bộ nhớ trong với các quyền truy xuất, cài đặt ứng dụng, hệ thống tương tự như cách mà thiết bị làm với bộ nhớ trong mặc định thiết bị.

Với sự phát triển của các tính năng bảo mật vân tay, và chuẩn kết nối USB Type-C phổ biến hơn, thì Android 6.0 Marshmallow cũng có sự hỗ trợ tốt hơn trong khả năng tương tác. Điều đó mở ra một sự dễ dàng hơn đối với các nhà sản xuất thiết bị trong việc tích hợp các tính năng này lên các thiết bị của họ mà không còn quá lo ngại về khả năng hoạt động chính xác của chúng như với các phiên bản Android khác được ra mắt trước đây.

Với những gì làm được của mình, Android 6.0 Marshmallow đã trở thành một trong những phiên bản được sử dụng nhiều nhất trong đại gia đình Android cho tới thời điểm hiện tại, với 24% thị phần trên toàn thế giới theo các thống kê tính toán đến thời điểm cuối tháng 12 năm 2016. [2]

1.5.12. Phiên bản Android 7.0



Hình 1.5.12.1. Phiên bản Android 7.0

Android 7.0 Nougat là phiên bản mới nhất của Android được ra mắt từ khá sớm dưới dạng phiên bản thử nghiệm vào thời điểm tháng 3/2016, nhưng phải đến tháng 8/2016, thì nó mới chính thức được phân phối dưới dạng phiên bản chính thức mà Google mang đến cho người dùng của mình.

Dường như giữa Nougat và Marshmallow, sự khác biệt của các phiên bản trong hệ sinh thái Android này rất nhỏ, khi mà Nougat chỉ vượt trội hơn Marshmallow ở khả năng hỗ trợ chế độ Split Screen để người dùng có thể tương tác đồng thời nhiều ứng dụng cùng lúc trên các thiết bị có màn hình kích thước đủ lớn như với các dòng tablet hiện nay.

Cùng với đó, Nougat cũng cho phép người dùng tương tác với các thanh thông báo nhiều hơn, khi nó không đơn thuần là nhắc nhở về các sự kiện diễn ra trên thiết bị, mà với một vài chức năng liên quan đến tin nhắn hay Messenger, người dùng có thể trực tiếp trả lời lên trên đó một cách nhanh chóng mà không cần phải mở màn hình làm việc của ứng dụng nếu như họ cảm thấy không thực sự cần thiết.

Có thể Android 7.0 Nougat còn khá sớm để có thể so sánh về mặt thị phần với Android 6.0 Marshmallow, thế nhưng với các tính năng và sự cải tiến trong trải

nghiệm người dùng của mình, đây thực sự là một phiên bản hứa hẹn trong thời gian tới khi mà nó được ứng dụng một cách rộng rãi hơn.[2]

1.5.13. Phiên bản Android 8.0 – 8.1



Vào ngày 21 tháng 3 năm 2017, Google phát hành phiên bản xem trước đầu tiên cho nhà phát triển của Android "O", dành cho Nexus 5X, Nexus 6P, Nexus Player, Pixel C, và cả hai chiếc điện thoại thông minh Pixel. Phiên bản thứ hai, được coi là bản beta, được phát hành ngày 17 tháng 5 năm 2017. Phiên bản xem trước thứ ba được phát hành ngày 8 tháng 6 năm 2017. Phiên bản thứ tư và cũng là bản beta cuối cùng được phát hành ngày 24 tháng 7 năm 2017.

Phiên bản xem trước thứ ba đã hoàn thiện API của Oreo lên cấp độ 26, thay đổi giao diện người dùng máy ảnh, thay đổi lại thứ tự biểu tượng Wi-Fi và mạng di động, thêm các thông báo theo chủ đề, thêm hoạt họa pin trong phần Cài đặt: Pin, biểu tượng mới và nền tối hơn cho ứng dụng Đồng hồ, và thay đổi biểu tượng cho một số ứng dụng.

Vào ngày 18 tháng 8 năm 2017, Google ra mắt một trang web hé lộ theo chủ đề nhật thực, ghi ngày phát hành và giới thiệu tên của Android O vào ngày 21 tháng 8, trùng với sự kiện nhật thực tại Mỹ. Cái tên cuối cùng được giới thiệu là "Oreo", theo một thỏa thuận giữa Google và Nabisco, nhà sản xuất của thương hiệu bánh quy nổi tiếng thế giới. [2]

CHƯƠNG 2: MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG ANDROID STUDIO

2.1. Sơ lược về Android Studio

Google cung cấp một công cụ phát triển ứng dụng Android trên Website chính thức dựa trên nền tảng IntelliJ IDEA gọi là Android Studio. Android studio dựa vào IntelliJ IDEA, là một IDE tốt cho nhất Java hiện nay. Do đó Android Studio sẽ là môi trường phát triển ứng dụng tốt nhất cho Android.

2.1.1. Cài đặt Android Studio

Android Studio là môi trường phát triển tích hợp (IDE) chính thức dành cho phát triển nền tảng Android.

Nó được ra mắt vào ngày 16 tháng 5 năm 2013 tại hội nghị Google I/O. Android Studio được phát hành miễn phí theo giấy phép Apache Licence 2.0.

Android Studio ở giai đoạn truy cập xem trước sớm bắt đầu từ phiên bản 0.1 vào tháng 5.2013, sau đó bước vào giai đoạn beta từ phiên bản 0.8 được phát hành vào tháng 6 năm 2014. Phiên bản ổn định đầu tiên được ra mắt vào tháng 12 năm 2014, bắt đầu từ phiên bản 1.0.

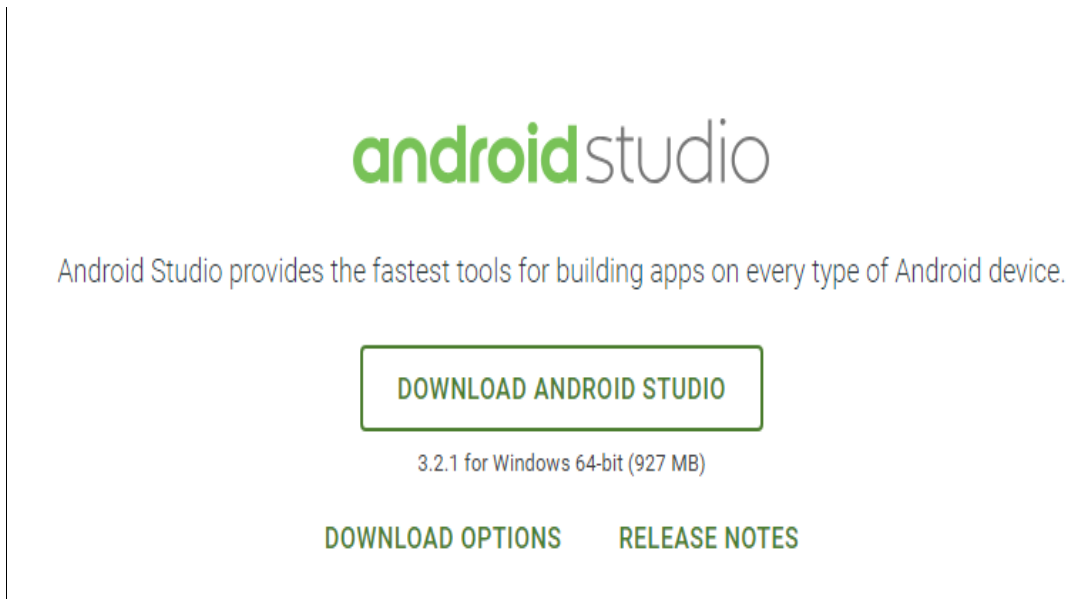
Dựa trên phần mềm IntelliJ IDEA của JetBrains, Android Studio được thiết kế đặc biệt để phát triển ứng dụng Android. Nó hỗ trợ các hệ điều hành Windows, Mac OS X và Linux, và là IDE chính thức của Google để phát triển ứng dụng Android gốc để thay thế cho Android Development Tools (ADT) dựa trên Eclipse.

a. Yêu cầu phần cứng máy tính

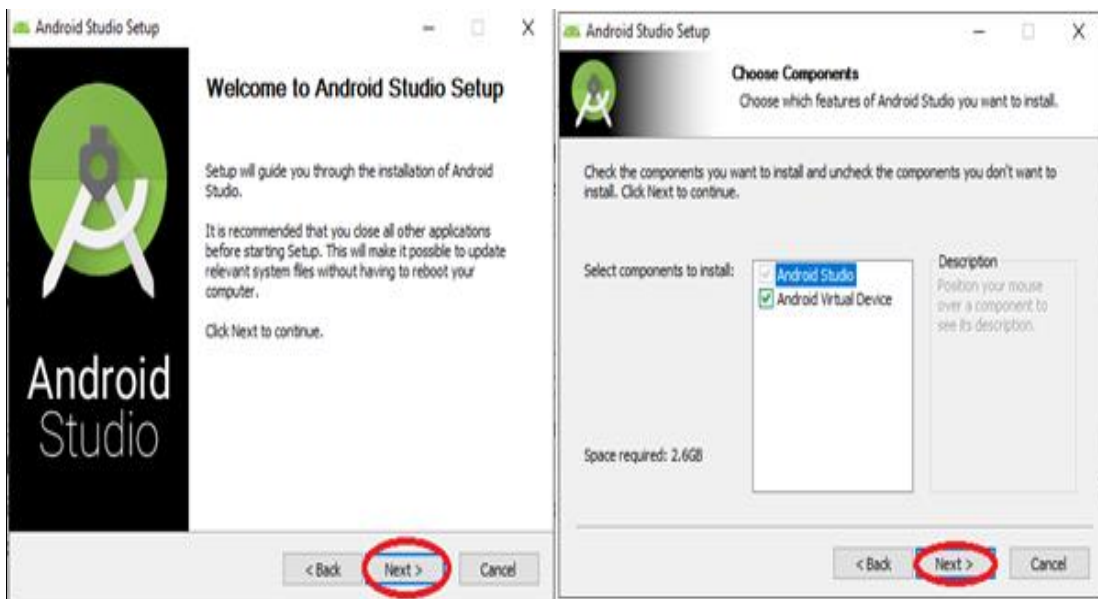
- Microsoft® Windows® 10/8/7 (32 or 64-bit)
 - GB RAM. (Khuyến cáo là 8GB)
 - Chip core I3 trở lên
 - 400 MB hard disk space + ít nhất 1GB cho Android SDK, emulator
- Độ phân giải tối thiểu 1366 x 768

b. Tải file cài đặt

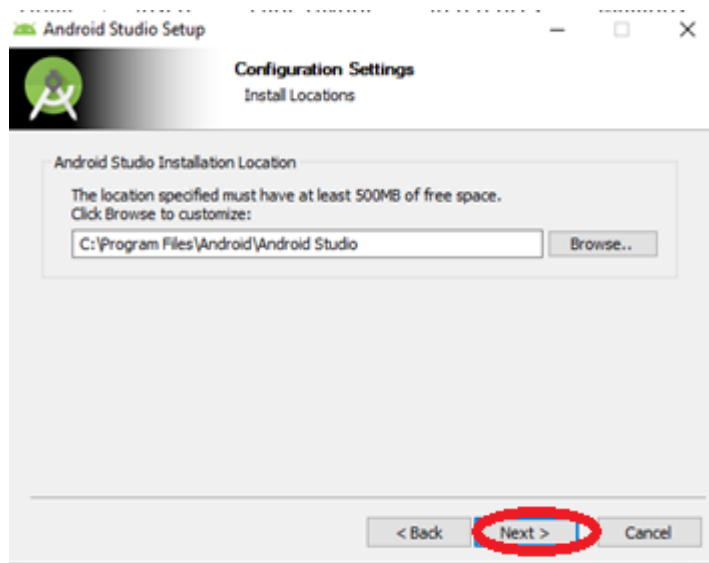
Download file cài đặt: <https://developer.android.com/studio/>

**c. Cách cài đặt**

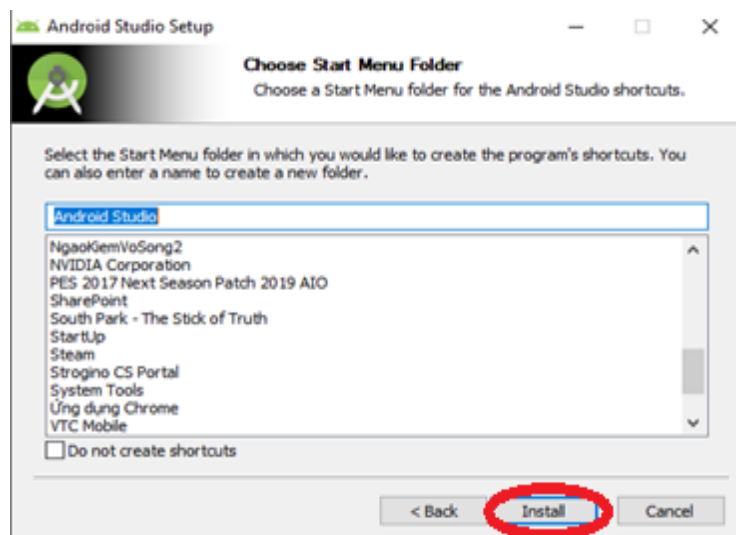
- Bước 1: Mở file cài đặt và ấn Next.



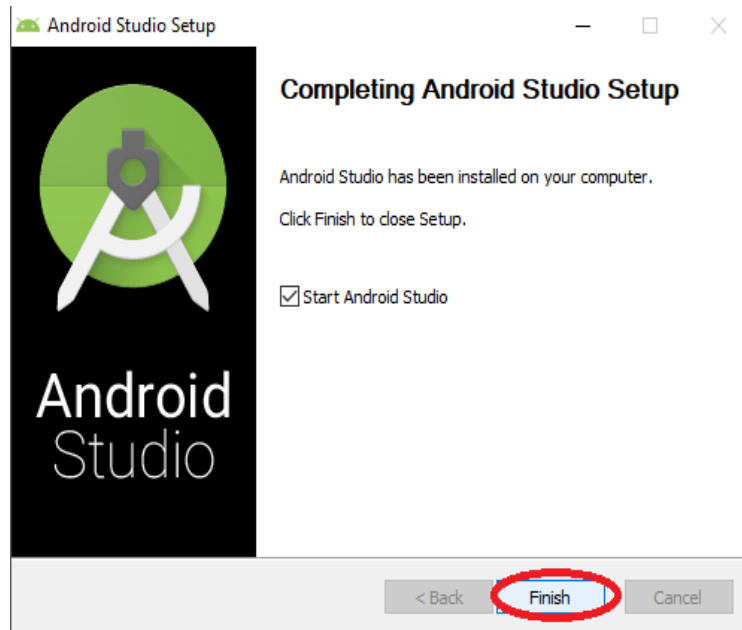
- Bước 2: Chọn nơi cài đặt Android Studio và ấn next



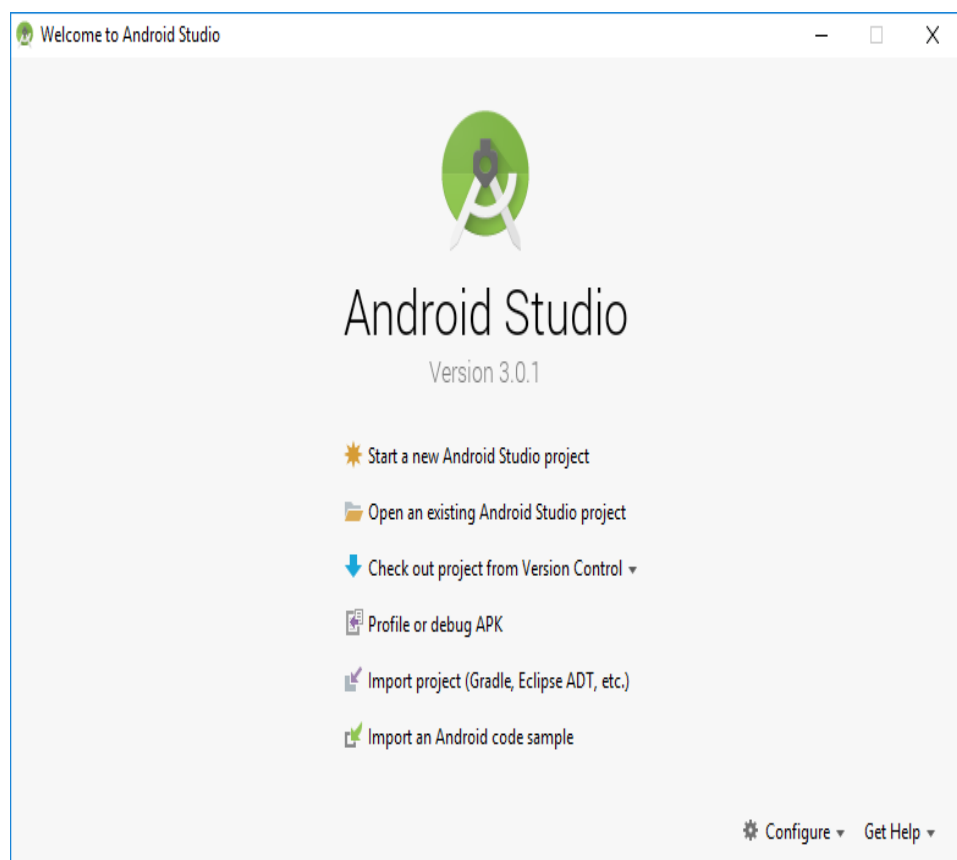
- Bước 3: Chọn Start Menu Folder. Ở bước này, có thể tích tạo short cut cho ứng dụng hoặc không. Bấm Install để bắt đầu cài đặt.



- Bước 4: Bấm Finish và click Start Android Studio để khởi động phần mềm.



- Bước 5: Lần đầu tiên khởi chạy **Android Studio**, nó sẽ tiến hành cập nhật một loạt các thành phần, chờ cho nó cập nhật xong xuất hiện màn hình Wizard, từ đó tùy bạn sử dụng có thể tạo Project mới hay mở Project đã có sẵn để làm việc.



2.1.2. Máy ảo Genymotion

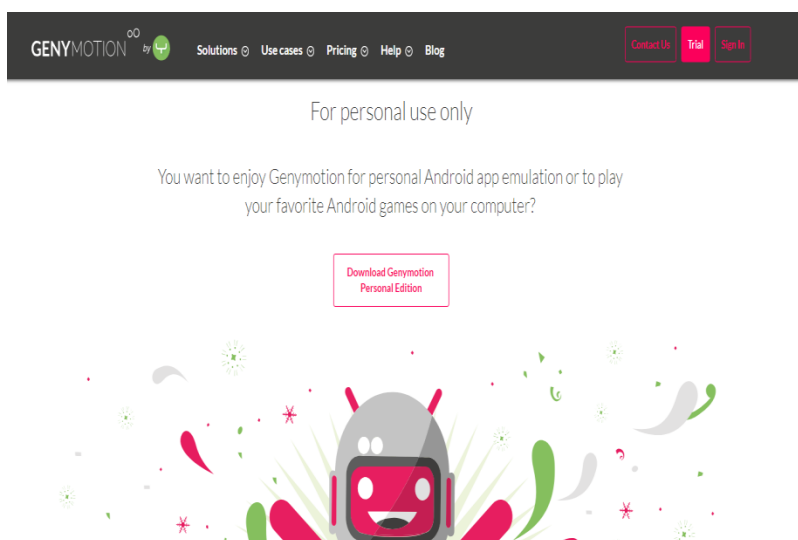
Genymotion là một trình giả lập dành cho Android và có sẵn các phiên bản Android được cấu hình sẵn. Đây là máy ảo rất tuyệt vời dùng để dành cho việc thử nghiệm ứng dụng.

a. Cấu hình yêu cầu:

- Card đồ họa OpenGL 2.0.
- CPU hỗ trợ VT-x hoặc AMD-V và kích hoạt thiết lập BIOS.
- RAM: Tối thiểu 1 GB.
- Dung lượng trống của ổ cứng: Ít nhất là 2GB để cài đặt Genymotion và các máy ảo chạy Genymotion (đây chỉ là mức yêu cầu tối thiểu, bởi nếu sử dụng nhiều máy ảo cùng lúc và có nhiều ứng dụng, phần mềm cài đặt thì dung lượng trống có thể được yêu cầu nhiều hơn gấp 4 lần).
- Đảm bảo có kết nối Internet
- Độ phân giải màn hình: ít nhất 1024 x 768 pixel
- Oracle VirtualBox 4.1 trở lên
- Ngoài ra, người dùng cần có một tài khoản Genymotion để có thể sử dụng.

b. Tải file cài đặt:

Download tại: <https://www.genymotion.com/fun-zone/>

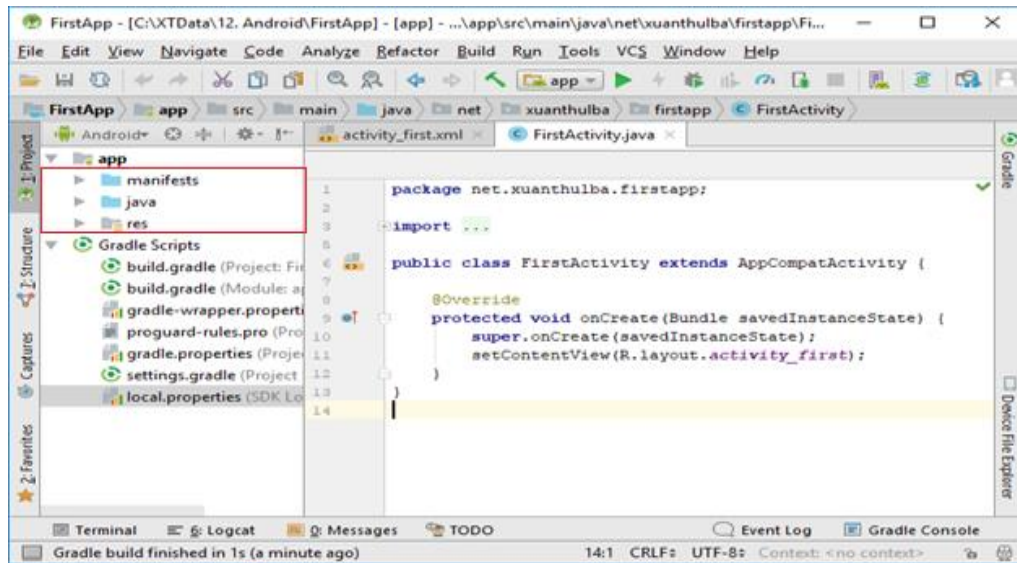


c. Cách cài đặt

- Bước 1: Tìm tới vị trí lưu file vừa tải về đầu tiên, sau đó click đúp chuột trái vào file đó để bắt đầu tiến hành. Giao diện đầu tiên sẽ cho lựa chọn ngôn ngữ để sử dụng.
- Bước 2: Tùy theo ý muốn người dùng mà có thể Next ngay và cài đặt Genymotion ngay tại vị trí được gợi ý hoặc chọn Browse để tìm vị trí khác.
- Bước 3: Tiếp tục Next.
- Bước 4: Chọn mục Create a desktop shortcut để tạo biểu tượng cho phần mềm này trên máy tính, sau đó Next tiếp. Chọn vào Install để thực hiện cài đặt.
- Bước 5: Rất có thể Genymotion sẽ tự động cài thêm một số phần mềm hỗ trợ, đồng thời hỏi có muốn cài đặt VirtualBox hay không? Nếu máy tính của đã có sẵn phần mềm VirtualBox mới nhất rồi thì thôi, còn nếu không, chọn Yes để cài đặt tiếp.
- Bước 6: Tới đây, ngoài cửa sổ cài đặt Genymotion (sẽ tạm thời bị dừng lại), sẽ thấy có thêm một giao diện khác để tiến hành cài đặt VirtualBox.
- Bước 7: Sẽ không có thêm bất cứ phần mềm hay thao tác thiết lập nào khác, nên các có thể yên tâm Next tới khi kết thúc > Next tiếp nào > Click chọn Yes.
- Bước 8: Install.
- Bước 9: Bỏ dấu tick ở Start Oracle VM VirtualBox và chọn vào Finish.
- Bước 10: Lúc này cửa sổ cài đặt cuối cùng của Genymotion xuất hiện, Finish để kết thúc quá trình.

2.2. Thành phần trong một dự án Android

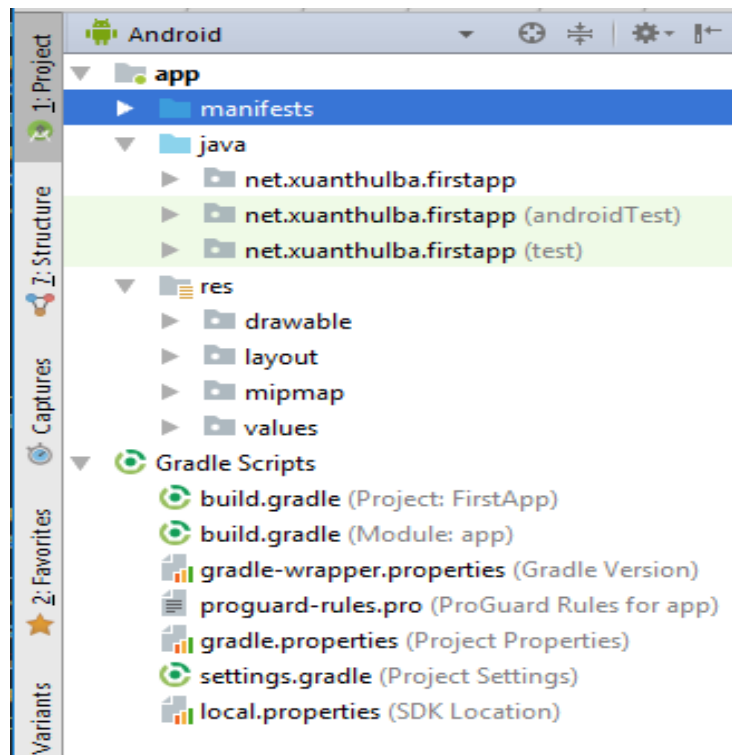
Một project android gồm có ba thành phần chính :



Hình 2.2.1. Thành phần chính trong một dự án Android

2.2.1. Tìm hiểu về cấu trúc Project

Nhìn vào phần của sổ Project như hình 2.2.1, bạn thấy nó có cấu trúc dạng cây, mô tả chính cấu trúc một dự án Android



Hình 2.2.1.1. Cấu trúc Project

Có thể click đúp vào từng thành phần để xem chi tiết, hay mở nó ra trong cửa sổ soạn thảo để biên tập các thành phần tương ứng:

- **manifests** : tham chiếu đến file `AndroidManifest.xml`, file này để bạn mô tả cấu trúc chính của App, nhằm đúng Android OS biết được các thiết lập cơ bản của App để khởi chạy được nó, như Activity nào sẽ được chạy, ứng dụng xin những quyền gì trong thiết bị ...
- **java**: thư mục lưu trữ các file code java của ứng dụng, lớp `FirstActivity` được định nghĩa với file `FirstActivity.java` lưu trong cấu trúc thư mục này.
- **res** : lưu trữ các file tài nguyên mà ứng dụng sẽ sử dụng đến, nó tổ chức thành các thư mục con như:
 - **drawable** : ở đây cơ bản lưu các đối tượng đồ họa như các ảnh dạng png, các ảnh dạng xml ...
 - **layout** : lưu trữ các file xml biểu diễn về thành phần, bố cục của các thành phần hiển thị được trên màn hình
 - **mipmap** : cũng để lưu các đối tượng hình ảnh, ví dụ icon ứng dụng `ic_launcher` đặt ở đây
- **values**: chứa các file `colors.xml`, `dimens.xml`, `strings.xml`, `styles.xml`, đây là các file xml định nghĩa các giá trị có thể sử dụng trong ứng dụng như màu sắc, kích thước, các chuỗi, các theme ...
- **GradleScripts**: chứa nhiều nhánh con `build.gradle`, `local.properties` ... là nơi bạn thiết lập các thông số để Gradle build ứng dụng. Bạn lưu ý Gradle là một công cụ tích hợp vào Android Studio, chức năng của nó build mã nguồn, kết hợp tài nguyên, phân tích xml ... rồi kết hợp chúng lại với nhau tạo ra ứng dụng chạy trên JVM.[3]

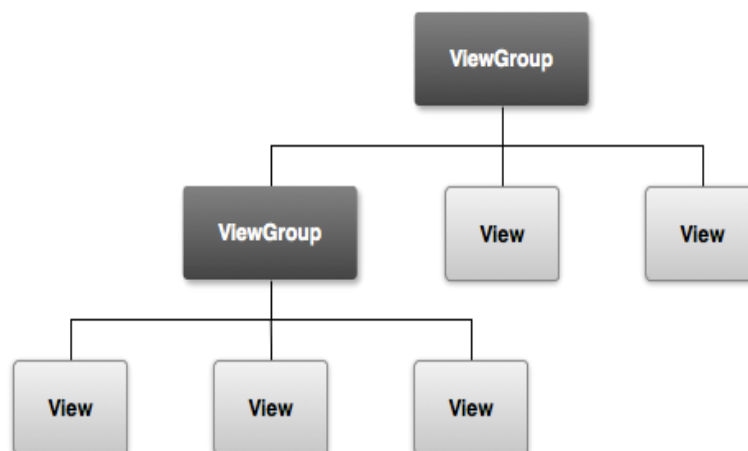
2.3. Thành phần giao diện

2.3.1. View Group

a. Khái niệm

Một `ViewGroup` là một đối tượng được sử dụng để chứa các đối tượng `View` và `ViewGroup` khác để tổ chức và kiểm soát layout của một màn hình. Các đối tượng `ViewGroup` được sử dụng cho việc tạo ra một hệ thống phân cấp của các đối

tượng View (xem bên dưới) do đó có thể tạo các layout phức tạp hơn. Chúng ta có thể tham khảo sơ đồ phân cấp dưới đây:



Hình 2.3.1.1. Sơ đồ phân cấp các thành phần giao diện

b. Một số View Group cơ bản

- **LinearLayout**: Tồn tại để hiển thị các phần tử theo một thứ tự xếp chồng lên nhau theo chiều ngang hoặc chiều dọc. LinearLayout cũng có thể được sử dụng để gán weight cho các phần tử View con để các phần tử được cách khoảng trên màn hình theo tỉ lệ tương ứng với nhau.
- **RelativeLayout**: Lớp con này của ViewGroup cho phép hiển thị các phần tử trên màn hình tương đối với nhau, cung cấp nhiều tính linh hoạt hơn và tự do trong cách layout của xuất hiện so với LinearLayout.
- **FrameLayout**: Được thiết kế để hiển thị một View con tại một thời điểm, FrameLayout vẽ các phần tử trong một ngăn xếp và cung cấp một cách đơn giản để hiển thị một phần tử trên các kích cỡ màn hình khác nhau.
- **ScrollView**: Một lớp mở rộng của FrameLayout, lớp ScrollView xử lý việc cuộn các đối tượng con của nó trên màn hình.
- **RecyclerView**: Lớp RecyclerView là một lớp con của ViewGroup, nó liên quan đến các lớp ListView và GridView và nó được cung cấp bởi Google thông qua thư viện hỗ trợ RecyclerView cho các phiên bản Android cũ hơn. Lớp RecyclerView đòi hỏi việc sử dụng các mẫu thiết kế view holder để tái sử dụng phần tử một cách có hiệu quả và nó hỗ trợ việc sử dụng một layoutManager, một thành phần trang trí, và một phần tử động để làm cho thành phần này vô cùng linh hoạt và đơn giản.

- CoordinatorLayout: Được thêm gần đây vào thư viện hỗ trợ thiết kế, lớp CoordinatorLayout sử dụng một đối tượng Behavior để xác định cách các phân tử View con sẽ được sắp xếp và di chuyển khi người dùng tương tác với ứng dụng của .[4]

2.3.2. View

a. Khái niệm

Trong một ứng dụng Android, giao diện người dùng được xây dựng từ các đối tượng View và ViewGroup. Có nhiều kiểu View và ViewGroup. Tất cả các kiểu đó được gọi là các Widget. Tất cả mọi widget đều có chung các thuộc tính cơ bản như là cách trình bày vị trí, background, kích thước, lề,... Tất cả những thuộc tính chung này được thể hiện hết ở trong đối tượng View. Trong Android Platform, các screen luôn được bố trí theo một kiểu cấu trúc. Một screen là một tập hợp các Layout và các widget được bố trí có thứ tự.

b. Một số view cơ bản trong Anroid

Tên widget	Chức năng
TextView	Cho phép người dùng hiển thị một đoạn văn bản lên màn hình mà không cho phép người dùng sửa nó.
EditText	Cho phép người dùng nhập, xóa, sửa một đoạn văn bản vào trong đó. Có nhiều dạng EditText khác nhau như: Plaintext, Person Name, Password, Email, Phone,...
Button	Dùng để thiết lập các sự kiện khi người dùng thao tác với nó.
ImageButton	Là dạng nút bấm nhưng có thể chèn thêm hình ảnh vào để giao diện thêm sinh động, trực quan hơn.
CheckBox	Một dạng nút bấm đặc biệt chỉ có hai trạng thái là check và uncheck.
ImageView	Hiển thị hình ảnh
SeekBar	Là một tiện ích cho phép người dùng chỉ định một giá trị bằng cách kéo bên phải hoặc trái khi lập trình android
ListView	Hiển thị các thành phần theo một danh sách, có thể cuộn

	được theo chiều thẳng đứng.
ProgressBar	Thể hiện tiến trình, mức độ

c. Thuộc tính cơ bản widget

Tên thuộc tính	Công dụng
layout_width	Chiều rộng của control (Tính theo trục Ox từ góc trên bên trái qua phải). Có ba thuộc tính: wrap_content (Co dẫn theo nội dung control), match_parent (kích thước chiều ngang bằng kích thước đối tượng chứa nó)
layout_height	Chiều cao của control (Tính theo trục Oy từ góc trên bên trái xuống dưới). Có ba thuộc tính như layout:width
text	Nội dung hiển thị
textcolor	Màu chữ
background	Màu nền của view
id	Định danh của view
gravity	Căn lề cho nội dung của view
textStyle	Thiết lập kiểu dáng chữ: đậm, in nghiêng, gạch chân
textSize	Thiết lập cỡ chữ
fontFamily	Thiết lập họ phông chữ
hint	Đoạn văn bản gợi ý cho người dùng biết về chức năng hay ràng buộc gì đó...
inputType	Thiết lập kiểu nhập dữ liệu

2.4. Bắt và xử lý sự kiện trên giao diện

Sự kiện là một cách hữu ích để thu thập dữ liệu về sự tương tác của người dùng với các thành phần tương tác của ứng dụng. Giống như bấm vào một nút hoặc chạm vào màn hình cảm ứng, vv. Ta có thể nắm bắt những sự kiện trong chương trình và có những xử lý thích hợp theo yêu cầu. Có hai khái niệm liên quan đến quản lý sự kiện Android:

- Event Listeners là một interface. Event Listeners được sử dụng để đăng ký sự kiện cho các thành phần trong UI. (Đăng ký sự kiện) [5]. Trong các giao tiếp event listener có những phương thức sau đây:
 - **onClick()**: Thuộc View.OnClickListener. Nó được gọi khi người dùng hoặc chạm vào item (khi ở chế độ cảm ứng), hoặc lựa chọn vào item với các phím điều hướng và nhấn nút "enter" phù hợp.
 - **onLongClick()**: Thuộc View.OnLongClickListener. Nó được gọi khi người dùng chạm và giữ item (khi ở chế độ cảm ứng), hoặc lựa chọn vào item với các phím điều hướng sau đó nhấn và giữ phím "enter".
 - **onFocusChange()**: Thuộc View.OnFocusChangeListener. Nó được gọi khi người dùng điều hướng ra khỏi item, bằng cách sử dụng phím điều hướng.
 - **onKey()**: Thuộc View.OnKeyListener. Nó được gọi khi người dùng lựa chọn và nhấn lên item.
 - **onTouchEvent()**: Thuộc View.OnTouchListener. Nó được gọi khi người dùng thực hiện một hành động xác định đủ điều kiện như là một sự kiện cảm ứng, bao gồm việc nhấn, thoát ra, hoặc bất kỳ cử chỉ chuyển động vẽ trên màn hình (bên trong phạm vi của item).
 - **onCreateContextMenu()**: Thuộc View.OnCreateContextMenuListener. Nó được gọi khi một menu ngữ cảnh (Context Menu) đang được xây dựng (là kết quả của một "long click"). Xem thêm thông tin về context menus trong hướng dẫn phát triển Menus.

2.5. Thành phần trong ứng dụng Android

Có 4 kiểu thành phần trong ứng dụng của Android bao gồm:

- Activities.
- Services.
- Content providers.
- Broadcast receivers.



Hình 2.5.1. Các thành phần trong ứng dụng Android

2.5.1. Vòng đời hoạt động

Một Activity được xem như một điểm tiếp xúc với người dùng. Nó là một màn hình đơn với giao diện trên đó. Ví dụ các bạn cài và chạy lên ứng dụng này thì màn hình chạy lên các bạn sẽ thấy giao diện hiện thị cho chúng ta xem, đó là một Activity. Activity giúp người dùng tương tác với hệ thống, thực hiện các chức năng cần thiết trên đó, chuyển đổi qua lại giữa các màn hình giao diện chức năng.

Thường thường khi sử dụng Activity chúng ta sẽ kế thừa từ lớp cha của nó là Activity (tất nhiên hiện tại Android SDK ở các phiên bản mới đã có nhiều subActivity hỗ trợ chúng ta trong từng trường hợp thuận tiện).[6]

2.5.2. Services

Service có chức năng giúp ứng dụng vẫn chạy được, nhưng không cần hiển thị trên giao diện (gọi là chạy ngầm bên dưới). Ví dụ các bạn dùng các ứng dụng nghe nhạc, mặc dù các bạn tắt ứng dụng rồi nhưng vẫn nghe được nhạc (đó là vì nó đang chạy dưới nền /background). Chúng ta có thể liên kết/ kết nối giữa một Activity với một service, ví dụ: khi download một file từ trên mạng, việc download thực hiện ở service, sau đó sẽ trả kết quả phần trăm download lên activity để hiển thị cho người dùng biết. Chú ý: mặc dù service chạy ở chế độ background nhưng cần phân biệt giữa service và thread. Service không phải thread, do đó tùy trường hợp mà chúng ta sử dụng và xử lý cho phù hợp để tránh trường hợp sử dụng service làm ứng dụng bị đơ/ chậm khi xử lý các luồng dữ liệu/ giao diện khác. Khi sử dụng service chúng ta sẽ kế thừa từ lớp cha là: Service.[6]

2.5.3. BroadcastReceiver

Broadcast receiver được sử dụng trong nhiều trường hợp, ví dụ: chúng ta có thể chuyển dữ liệu từ service lên activity (ngoài sử dụng binding) chúng ta có thể sử dụng broadcast để gửi dữ liệu. Hoặc trong các ứng dụng như hẹn giờ, khi đến giờ hẹn, ứng dụng sẽ sử dụng broadcast báo thức, tạo ra notification trên màn hình để báo cho người dùng biết[6].

2.5.4. Content providers

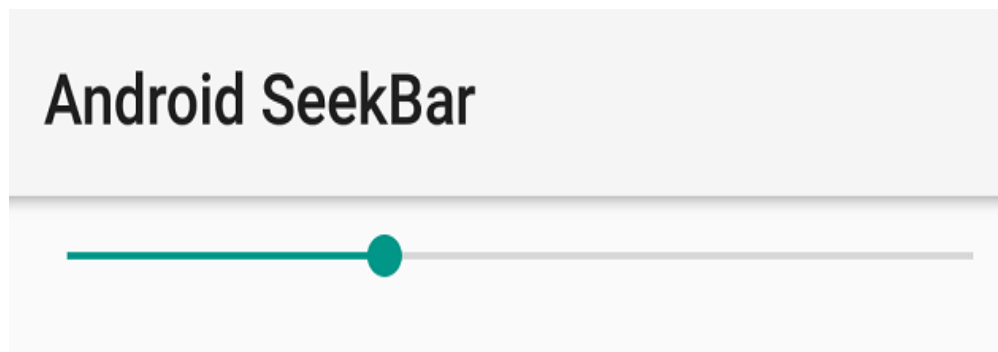
Content provider quản lý các cách để ứng dụng có thể lưu trữ dữ liệu trên hệ thống. Chúng ta sẽ biết cụ thể về thành phần này khi xây dựng các ứng dụng cần lưu trữ vào SQLite. Ví dụ các ứng dụng từ điển, các bạn sẽ thấy dữ liệu, từ vựng chúng ta tra hiện thị, thì dữ liệu hiện thị đó được lưu trữ trong Slite và Content provider gọi để lấy ra cho người dùng xem. Ngoài ra thành phần này còn thực hiện các chức năng, thêm, sửa, xóa dữ liệu...[6]

2.6. Một số công cụ để phát triển ứng dụng

2.6.1. Giới thiệu về Seekbar

Một SeekBar là một lớp mở rộng từ ProgressBar nó có thêm một cái cần gạt. Người dùng có thể chạm vào cần gạt và kéo sang trái hoặc phải để thiết lập giá trị

của tiến trình (progress) hoặc sử dụng các phím mũi tên để di chuyển cần gạt. Bạn cũng có thể chạm vào thanh bên trái hoặc bên phải cần gạt để di chuyển nó.



SeekBar code Trong XML:

```
<SeekBar
    android:id="@+id/seekbarba"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

Thuộc tính dùng trong seekbar:

1. **android:id**: Là thuộc tính duy nhất của SeekBar.

```
<SeekBar
    android:id="@+id/seekbarba" id là thuộc tính duy nhất trong project Android
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

2. **android:progressDrawable**: Thuộc tính này thường được sử dụng để tùy chỉnh lại ProgressBar. Trong ví dụ sau chúng ta sẽ tùy chỉnh lại một SeekBar:

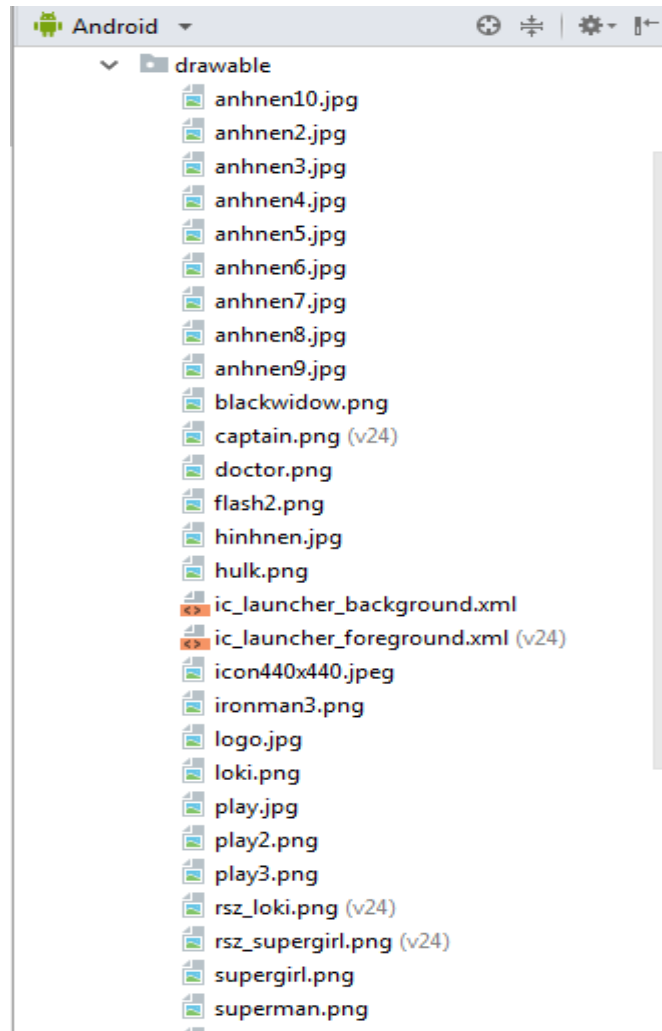
Thêm code vào main_activity.xml

```
<SeekBar
    android:id="@+id/seekbarhai"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:progressDrawable="@android:color/transparent"/>Thiết lập lại tùy chỉnh
seekbar
```

3. **android:thumb**: Thumb mặc định trong Android là nút để di chuyển seekbar sang trái hay sang phải, bạn có thể đổi cái thumb khác bằng hình ảnh trong thuộc tính này. Hình sau đây làm thumb mặc định của SeekBar.



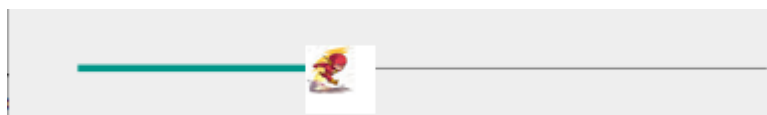
Ví dụ chúng ta đổi thành thumb có hình sau, trước hết phải download hình lưu vào thư mục drawable như hình sau:



Thêm code vào seekbar :

```
<SeekBar
    android:id="@+id/seekbarhai"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:progressDrawable="@android:color/transparent"
    android:thumb="@drawable/flash2" />
```

Kết quả thu được:



2.6.2. Giới thiệu về CheckBox

CheckBox là thành phần thể hiện trạng thái chọn (checked) hoặc không chọn (unchecked) **CheckBox** thường dùng khi người dùng có nhiều lựa chọn và được phép chọn một hoặc nhiều lựa chọn cùng lúc.



CheckBox trong code xml

```
<CheckBox
    android:id="@+id/checkboxba"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

Thuộc tính thường dùng trong checkbox

1. **android:id**: Là thuộc tính duy nhất của **CheckBox**. Xem ví dụ sau:

```
<CheckBox
    android:id="@+id/checkboxba"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

2. **android:text**: thuộc tính text dùng hiển thị nội dung trong một **CheckBox**. Chúng ta có thể set thuộc tính này trong tập tin xml.

```
<CheckBox
    android:id="@+id/checkboxba"
    android:text="Vidu"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

3. **android:textColor**: Thuộc tính này dùng xác định màu chữ, dạng màu chữ: "#argb", "#rgb", "#rrggbb", hoặc "#aarrggbb". Ví dụ sau set nội dung văn bản trong **CheckBox** màu đỏ.

```
<CheckBox
    android:id="@+id/checkboxba"
    android:text="Vidu"
    android:textColor="#f00" đổi màu chữ sang màu đỏ
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

2.6.3. Giới thiệu về ImageButton

ImageButton là cách sử dụng một hình ảnh như một nút bấm. Theo mặc định nó trông giống như một Button bình thường có background chuẩn. Ví dụ nút play trong ứng dụng:



ImageButton trong code xml

```
<ImageButton  
    android:id="@+id/buttonPlay"  
    android:layout_centerInParent="true"  
    android:src="@drawable/play"  
    android:layout_width="50dp"  
    android:layout_height="50dp" />
```

Thuộc tính thường dùng trong ImageButton

1. **android:id**: Là thuộc tính duy nhất của **ImageButton** trong project **Android**. Xem ví dụ sau:

```
<ImageButton  
    android:id="@+id/buttonPlay"  
    android:layout_centerInParent="true"  
    android:src="@drawable/play"  
    android:layout_width="50dp"  
    android:layout_height="50dp" />
```

2. **android:src**: là thuộc tính chứa tập tin hình ảnh cần hiển thị. Chúng ta cũng có thể tạo hình tại chế độ thiết kế:

```
<ImageButton  
    android:id="@+id/buttonPlay"  
    android:layout_centerInParent="true"  
    android:src="@drawable/play"  
    android:layout_width="50dp"  
    android:layout_height="50dp" />
```

2.6.4. Giới thiệu về CountdownTimer

CountdownTimer là class cung cấp chức năng giúp người lập trình tạo bộ đếm ngược trong Android.

Ví dụ:

```
final CountdownTimer countDownTimer = new CountdownTimer(60000, 200) {
    @Override
    public void onTick(long millisUntilFinished) {
        int number = 5;
        Random random = new Random();
        int one = random.nextInt(number);
        int two = random.nextInt(number);
        int three = random.nextInt(number);
        // kiểm tra đường 1 có về trước
        if(seekbarmot.getProgress() >= seekbarmot.getMax()){
            this.cancel();
            buttonPlay.setVisibility(View.VISIBLE);

            //Kiểm tra đặt cược
            if(checkboxmot.isChecked()){
                sodiem += 10;
                Toast.makeText(MainActivity.this, "BẠN ĐÃ ĐOÁN ĐÚNG", Toast.LENGTH_SHORT).show();
            } else {
                sodiem -=10;
                Toast.makeText(MainActivity.this, "BẠN ĐOÁN SAI", Toast.LENGTH_SHORT).show();
            }
            textviewdiemso.setText(sodiem + "");
            EnableCheckBox();
        }
        ////Kiểm tra xem đường 2 có về trước
        if(seekbarhai.getProgress() >= seekbarhai.getMax()){
            this.cancel();
            buttonPlay.setVisibility(View.VISIBLE);

            //kiểm tra đặt cược
            if(checkboxhai.isChecked()){
                sodiem += 10;
                Toast.makeText(MainActivity.this, "BẠN ĐÃ ĐOÁN ĐÚNG", Toast.LENGTH_SHORT).show();
            } else {
                sodiem -=10;
                Toast.makeText(MainActivity.this, "BẠN ĐOÁN SAI", Toast.LENGTH_SHORT).show();
            }
            textviewdiemso.setText(sodiem + "");
            EnableCheckBox();
        }
        ////Kiểm tra xem đường 3 có về trước
        if(seekbarba.getProgress() >= seekbarba.getMax()){
            this.cancel();
            buttonPlay.setVisibility(View.VISIBLE);

            //Kiểm tra đặt cược
            if(checkboxba.isChecked()){
                sodiem += 10;
                Toast.makeText(MainActivity.this, "BẠN ĐÃ ĐOÁN ĐÚNG", Toast.LENGTH_SHORT).show();
            } else {
                sodiem -=10;
                Toast.makeText(MainActivity.this, "BẠN ĐOÁN SAI", Toast.LENGTH_SHORT).show();
            }
            textviewdiemso.setText(sodiem + "");
            EnableCheckBox();
        }
    }

    seekbarmot.setProgress(seekbarmot.getProgress() + one);
}
```

```
seekbarhai.setProgress(seekbarhai.getProgress() + two);  
seekbarba.setProgress(seekbarba.getProgress() + three);  
}  
public void onFinish
```

Đoạn code trên ta hiểu rằng, một đối tượng CountdownTimer có là tên Timer được tạo ra kèm theo hai đối số 60000 và 200. 60000 và 200 có nghĩa là cứ 200 mi-li giây thì phương thức onTick(long millisUntilFinished) sẽ được chạy một lần, cứ chạy như thế trong suốt 60000 mi-li giây. 60000 mili giây trôi qua, thì phương thức onFinish() sẽ chạy.

Phương thức onTick, kèm theo một đối số millisUntilFinished, đây là đối số cho ta biết số milli còn lại của bộ đếm .

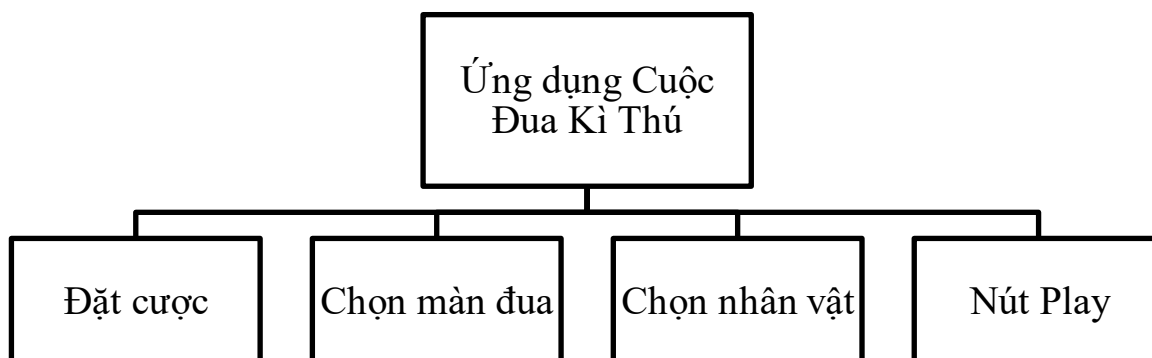
Phương thức onFinish, phương thức này được tự động gọi khi 30000 mi-li giây của bộ đếm trôi qua, đồng nghĩa với “Hết giờ”. Đoạn code trên ta có thể hiểu : Một bộ đếm ngược 60 giây được tạo ra.

CHƯƠNG 3: CHƯƠNG TRÌNH THỰC NGHIỆM

3.1. Phát biểu bài toán

Những chiếc smart phone chạy hệ điều hành Android ngày càng trở nên không thể thiếu được với chúng ta. Chúng vừa là phương tiện liên lạc cho công việc, vừa là phương tiện học tập và giải trí. Chương trình game Cuộc Đua Kì Thú mà người chơi đã chọn sẽ giúp bạn quên đi thời gian chờ đợi với luật chơi rất đơn giản: đầu tiên người chơi đặt cược cho nhân vật mà mình yêu thích, người chơi có thể chọn nhân vật và màn đua mà mình yêu thích, tiếp đó người chơi ấn nút play, nếu nhân vật bạn chọn về đích trước bạn sẽ được cộng 10 điểm, nếu bạn đoán sai thì sẽ bị trừ 10 điểm.

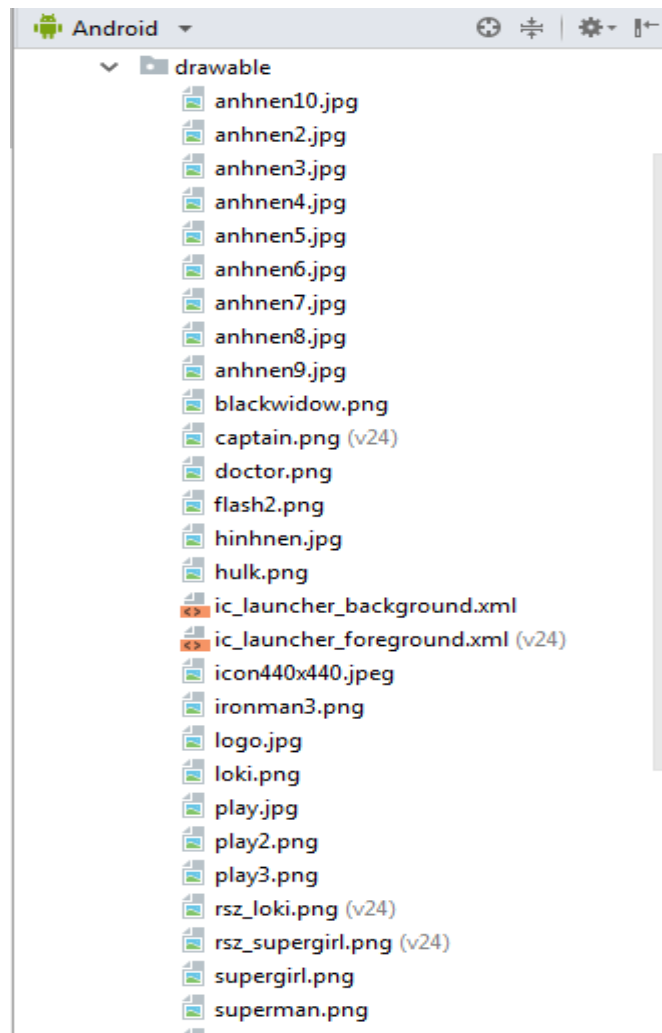
3.2. Sơ đồ chức năng



- Chức năng đặt cược: Khi người dùng ấn checkbox để đặt cược, người dùng chọn nhân vật yêu thích cho từng đường đua.
- Chức năng chọn màn đua: Khi người dùng ấn chọn màn đua, người dùng chọn màn đua mình yêu thích.
- Chức năng chọn nhân vật: Khi người dùng ấn chọn màn đua, người dùng chọn nhân vật yêu thích cho từng đường đua.
- Chức năng play: Khi người dùng ấn play, đường đua sẽ chạy về đích.

3.3. Kỹ thuật lập trình Cuộc Đua Kì Thú

Trước khi tạo giao diện cho chương trình, chúng ta sẽ nhập ảnh cần thiết vào trong thư mục drawable của project.



Hình 3.3.1. Nhập ảnh vào thư mục drawable

3.3.1. Tạo màn hình giao diện trò chơi

Giao diện chính của chương trình sử dụng `RelativeLayout` mặc định, khai báo trong file `activity_main.xml`.

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:tools="http://schemas.android.com/tools"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:background="@drawable/logo"
    android:id="@+id/manhinh"
    xmlns:android="http://schemas.android.com/apk/res/android">
    //Khai báo cho các View con được viết tại đây
</RelativeLayout>
```

Chúng ta chia giao diện thành 4 khu vực chính: chỗ đặt cược, đường đua, điểm số và nút Play.

a) Chỗ để người chơi đặt cược nhân vật mình yêu thích

Giao diện đặt cược để thể hiện qua CheckBox.

```
<CheckBox
    android:buttonTint="@color/colorAccent"
    android:layout_marginBottom="20dp"
    android:layout_marginLeft="5dp"
    android:layout_alignParentBottom="true"
    android:id="@+id/checkboxba"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="5dp" />

<CheckBox
    android:id="@+id/checkboxhai"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@id/checkboxba"
    android:layout_marginStart="5dp"
    android:layout_marginLeft="5dp"
    android:layout_marginBottom="25dp"
    android:buttonTint="@color/colorPrimary" />

<CheckBox
    android:id="@+id/checkboxmot"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/checkboxhai"
    android:layout_marginStart="5dp"
    android:layout_marginLeft="5dp"
    android:layout_marginBottom="23dp"
    android:buttonTint="@color/colorPrimaryDark" />
```

b) Đường đua

Đường đua và hình ảnh của các nhân vật được thể hiện qua SeekBar.

```
<SeekBar
    android:progressDrawable="@android:color/transparent"
    android:layout_marginBottom="10dp"
    android:thumb="@drawable/ironman3"
    android:layout_alignParentBottom="true"
    android:layout_toRightOf="@id/checkboxba"
    android:id="@+id/seekbarba"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_toEndOf="@id/checkboxba" />

<SeekBar
    android:id="@+id/seekbarhai"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/seekbarba"
    android:layout_marginStart="-1dp"
    android:layout_marginLeft="-1dp"
    android:layout_marginBottom="6dp"
    android:layout_toEndOf="@id/checkboxhai"
    android:layout_toRightOf="@id/checkboxhai"
    android:progressDrawable="@android:color/transparent"
    android:thumb="@drawable/flash2" />

<SeekBar
```

```
android:id="@+id/seekbarmot"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_above="@id/seekbarhai"
android:layout_marginStart="0dp"
android:layout_marginLeft="0dp"
android:layout_marginBottom="6dp"
android:layout_toEndOf="@id/checkboxxmot"
android:layout_toRightOf="@id/checkboxxmot"
android:progressDrawable="@android:color/transparent"
android:thumb="@drawable/thor2" />
```

c) Điểm số

Điểm số được thể hiện qua TextView.

```
<TextView
    android:id="@+id/textviewdiemso"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:text="@string/_100"
    android:textColor="#ff0"
    android:textSize="70sp"
    android:textStyle="bold" />
```

d) Nút Play

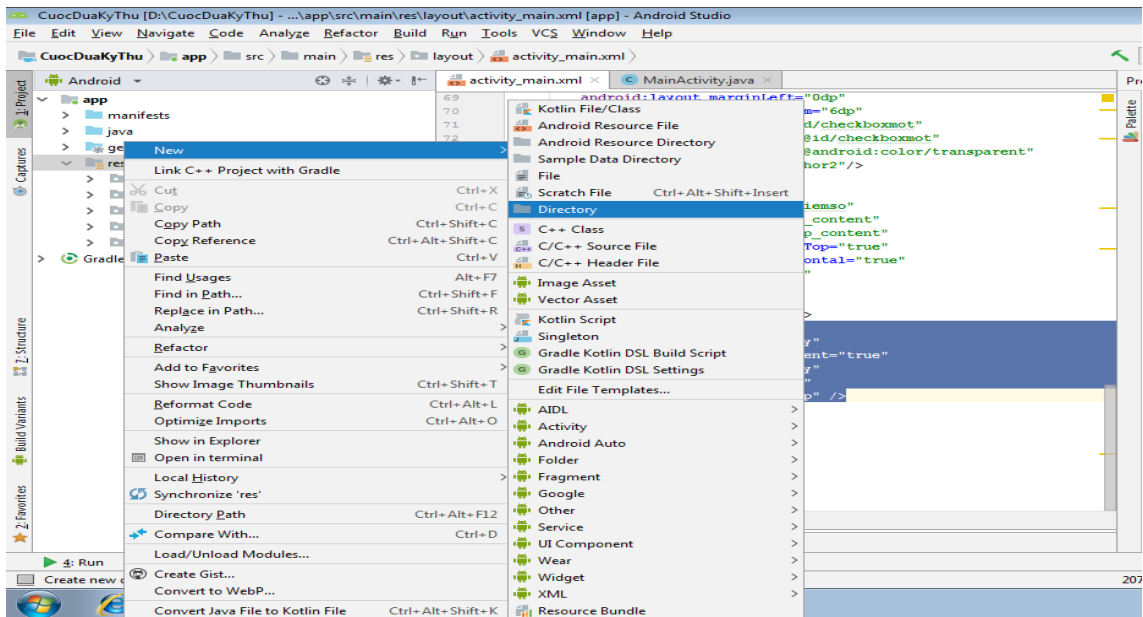
Nút Play được thể hiện qua ImageButton.

```
<ImageButton
    android:id="@+id/buttonPlay"
    android:layout_centerInParent="true"
    android:src="@drawable/play"
    android:layout_width="50dp"
    android:layout_height="50dp" />
```

3.3.2. Tạo menu giao diện chọn màn đưa và nhân vật

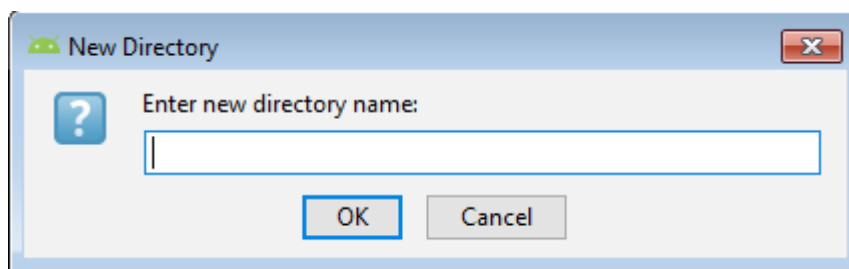
Chúng ta vào phần res ở cấu trúc project -> chuột phải -> New ->

Directory như hình 3.3.2.1:



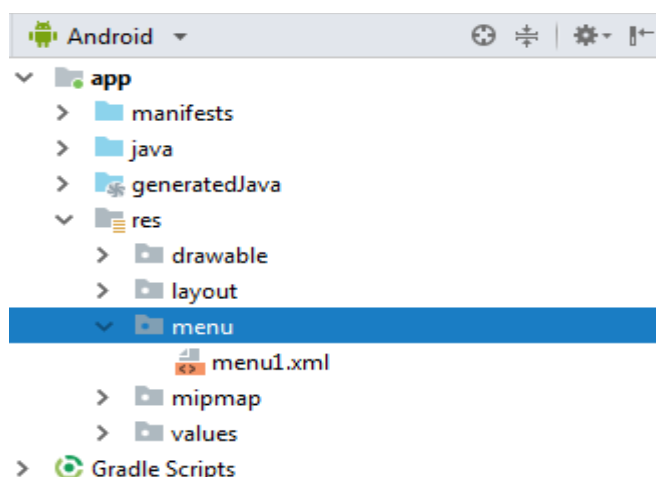
Hình 3.3.2.1. Tạo menu

Đặt tên thư mục cho menu như hình dưới:



Hình 3.3.2.2. Đặt tên thư mục menu

Click chuột phải vào thư mục menu vừa tạo chọn new và chọn Menu resource file ở đây mình đặt tên cho menu là menu1.



Hình 3.3.2.3. Tạo thư mục con menu1

Giao diện chính của menu khai báo trong file **menu1.xml**

```
<?xml version="1.0" encoding="utf-8" ?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

</menu>
```

Chúng ta chia menu làm 2 chức năng chính đó là chọn màn đưa và chọn nhân vật cho mỗi đường đưa.

a) Giao diện màn đưa.

```
<item android:title="@string/m_n_ua"
      android:id="@+id/menuhinhnen">
  <menu>
    <item android:title="@string/m_y"
          android:id="@+id/anhnen2"/>
    <item android:title="@string/th_m_ng"
          android:id="@+id/anhnen3"/>
    <item android:title="@string/n_c"
          android:id="@+id/anhnen4"/>
    <item android:title="@string/l_u_i"
          android:id="@+id/anhnen5"/>
    <item android:title="@string/m_a_h"
          android:id="@+id/anhnen6"/>
    <item android:title="@string/m_a_ng"
          android:id="@+id/anhnen7"/>
    <item android:title="@string/m_a_xu_n"
          android:id="@+id/anhnen8"/>
    <item android:title="@string/m_a_thu"
          android:id="@+id/anhnen9"/>
  </menu>
</item>
```

b) Giao diện chọn nhân vật.

```
<item android:title="@string/ch_n_nh_n_v_t"
      android:id="@+id/menunhanvat">
  <menu>
    <item android:title="@string/nh_n_v_t_1">
      <menu>
        <item android:title="@string/hulk"
              android:icon="@drawable/hulk"
              android:id="@+id/nv1"/>
        <item android:title="@string/blackwidow"
              android:icon="@drawable/blackwidow"
              android:id="@+id/nv2"/>
        <item android:title="@string/thor"
              android:icon="@drawable/thor2"
              android:id="@+id/nv3"/>
      </menu>
    </item>
    <item android:title="@string/nh_n_v_t_2">
      <menu>
        <item android:title="@string/loki"
              android:icon="@drawable/loki"
              android:id="@+id/nv4"/>
        <item android:title="@string/doctor"
              android:icon="@drawable/doctor"
              android:id="@+id/nv5"/>
        <item android:title="@string/flash"
              android:icon="@drawable/flash2"
              android:id="@+id/nv6"/>
      </menu>
    </item>
  </menu>
</item>
```

```

        </menu>
    </item>

    <item android:title="@string/nh_n_v_t_3">
        <menu>
            <item android:title="@string/captian"
                android:icon="@drawable/captain"
                android:id="@+id/nv7"/>
            <item android:title="@string/superman"
                android:icon="@drawable/supergirl"
                android:id="@+id/nv8"/>
            <item android:title="@string/ironman"
                android:icon="@drawable/ironman3"
                android:id="@+id/nv9"/>
        </menu>
    </item>
</menu>
</item>

```

3.3.3. Kỹ thuật cho 3 đường đua chạy ngẫu nhiên.

Ý tưởng là: 3 đường đua sẽ chạy ngẫu nhiên cho đến khi một đường đua về đến đích thì dừng lại, và kiểm tra xem đường đua nào về trước.

```

final CountdownTimer countDownTimer = new CountdownTimer(60000, 200) {
    @Override
    public void onTick(long millisUntilFinished) {
        int number = 5;
        Random random = new Random();
        int one = random.nextInt(number);
        int two = random.nextInt(number);
        int three = random.nextInt(number);
        // kiểm tra đường 1 có về trước
        if (seekbarmot.getProgress() >= seekbarmot.getMax()) {
            this.cancel();
            buttonPlay.setVisibility(View.VISIBLE);

            //Kiểm tra đặt cược
            if (checkboxxmot.isChecked()) {
                sodiem += 10;
                Toast.makeText(MainActivity.this, "BẠN ĐÃ ĐOÁN ĐÚNG",
                    Toast.LENGTH_SHORT).show();
            } else {
                sodiem -= 10;
                Toast.makeText(MainActivity.this, "BẠN ĐOÁN SAI",
                    Toast.LENGTH_SHORT).show();
            }
            textviewdiemso.setText(sodiem + "");
            EnableCheckBox();
        }
        //Kiểm tra xem đường 2 có về trước
        if (seekbarhai.getProgress() >= seekbarhai.getMax()) {
            this.cancel();
            buttonPlay.setVisibility(View.VISIBLE);

            //kiểm tra đặt cược
            if (checkboxxhai.isChecked()) {
                sodiem += 10;
                Toast.makeText(MainActivity.this, "BẠN ĐÃ ĐOÁN ĐÚNG",
                    Toast.LENGTH_SHORT).show();
            } else {
                sodiem -= 10;
                Toast.makeText(MainActivity.this, "BẠN ĐOÁN SAI",
                    Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```

    }
    textviewdiemso.setText(sodiem + "");
    EnableCheckBox();
}
//////Kiểm tra xem đường 3 có về trước
if(seekbarba.getProgress() >= seekbarba.getMax()){
    this.cancel();
    buttonPlay.setVisibility(View.VISIBLE);

    ///Kiểm tra đặt cược
    if(checkboxba.isChecked()){
        sodiem += 10;
        Toast.makeText(MainActivity.this, "BẠN ĐÃ ĐOÁN ĐÚNG",
Toast.LENGTH_SHORT).show();
    }else {
        sodiem -=10;
        Toast.makeText(MainActivity.this, "BẠN ĐOÁN SAI",
Toast.LENGTH_SHORT).show();
    }
    textviewdiemso.setText(sodiem + "");
    EnableCheckBox();
}

seekbarmot.setProgress(seekbarmot.getProgress() + one);
seekbarhai.setProgress(seekbarhai.getProgress() + two);
seekbarba.setProgress(seekbarba.getProgress() + three);
}
}

```

3.3.4. Kỹ thuật sử dụng ImageButton.

Ý tưởng là: Kiểm tra xem các checkbox đã được nhấn, sau khi nhấn nút play không cho người chơi tích vào checkbox nữa.

```

buttonPlay.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(checkboxmot.isChecked() || checkboxhai.isChecked() ||
checkboxboxba.isChecked()){
            seekbarmot.setProgress(0);
            seekbarhai.setProgress(0);
            seekbarba.setProgress(0);
            buttonPlay.setVisibility(View.INVISIBLE);

            countdownTimer.start();

            DisableCheckBox();
        }else {
            Toast.makeText(MainActivity.this, "ĐẶT CƯỢC TRƯỚC KHI CHƠI",
Toast.LENGTH_SHORT).show();
        }
    }
});

```


3.4. Các giao diện chương trình

3.4.1. Giao diện chính

- Người chơi có thể đặt cược vào checkbox và xem điểm như hình

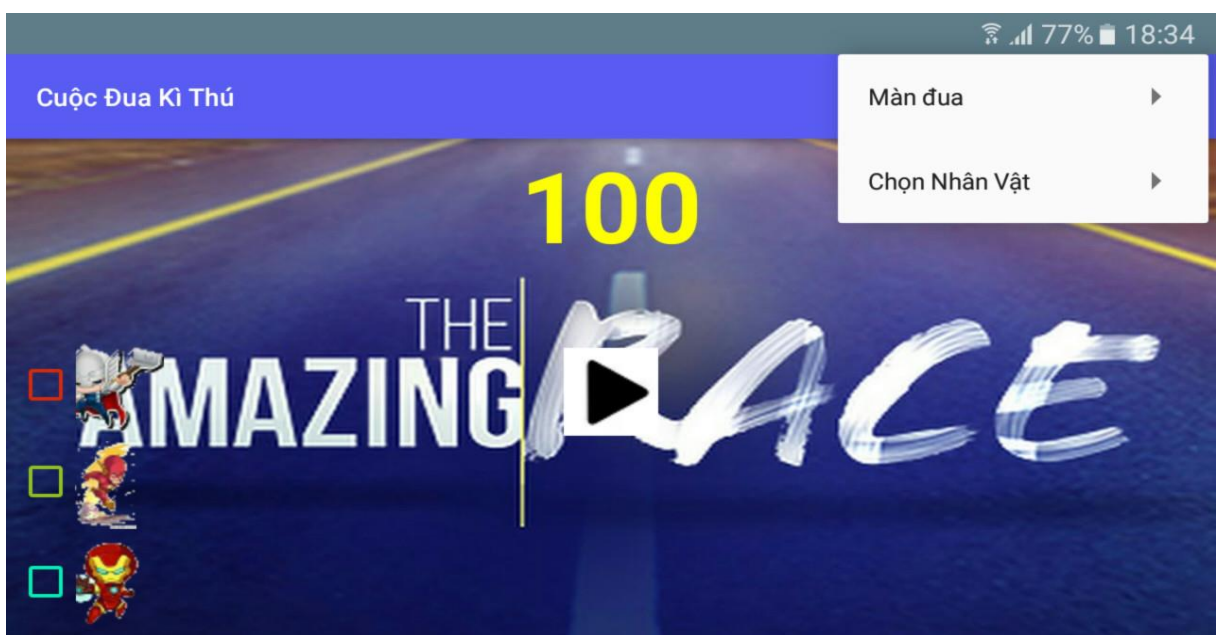
3.4.1.1:



Hình 3.4.1.1. Giao diện chính

3.4.2. Giao diện menu chính

- Người chơi có thể chọn màn đua và chọn nhân vật mình thích cho từng đường đua như hình 3.4.2.1 dưới đây:



Hình 4.3.2.1. Giao diện menu

- Giao diện chọn màn đua.



Hình 4.3.2.2. Giao diện màn đua

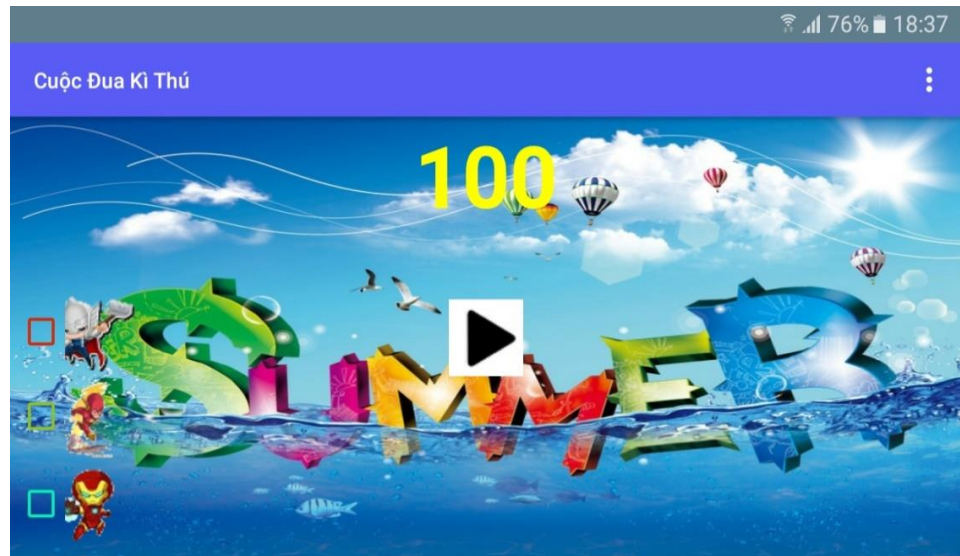
- Một số giao diện màn đua



Hình 4.3.2.3. Giao diện màn đua mùa xuân

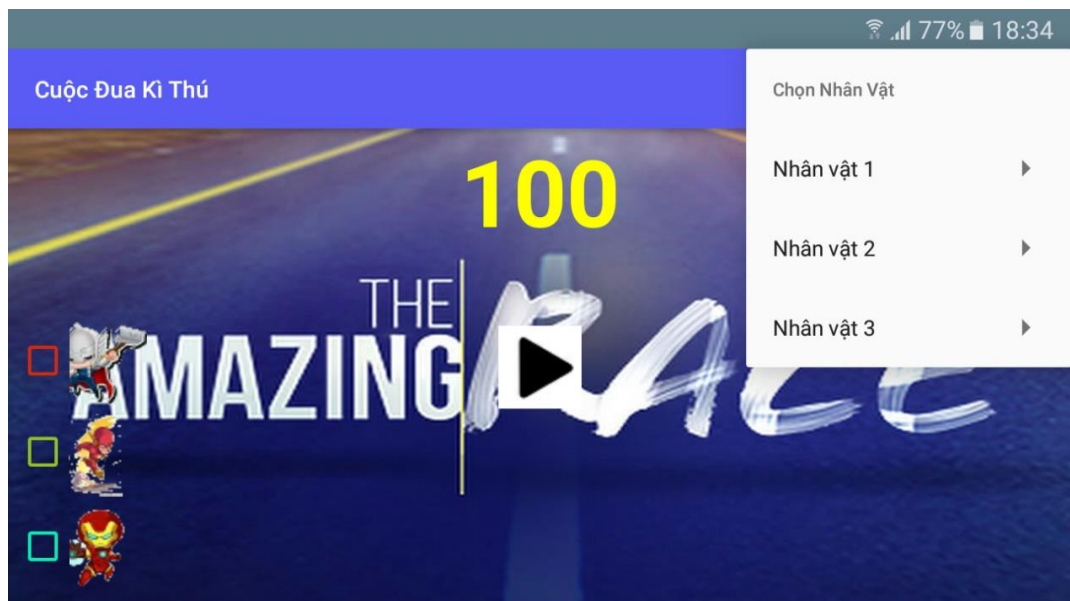


Hình 4.3.2.4. Giao diện màn đua mùa đông



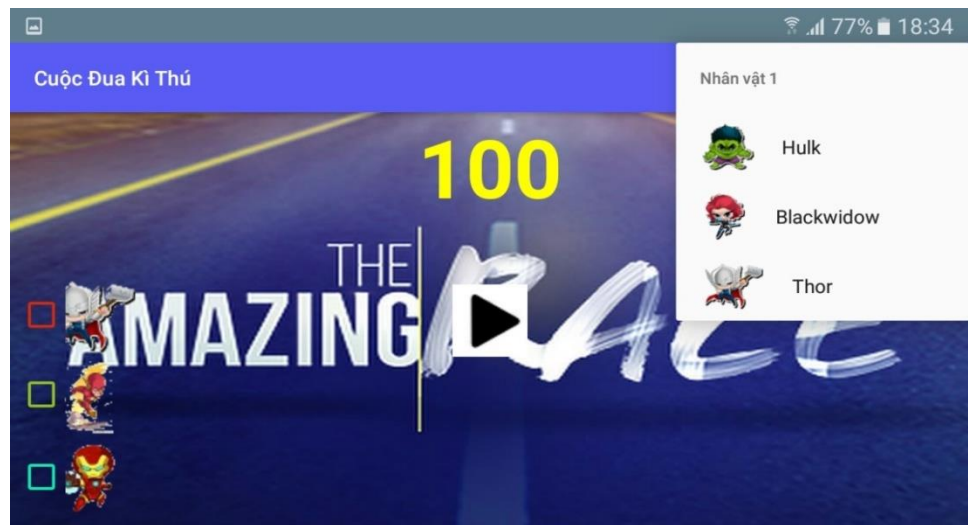
Hình 4.3.2.5. Giao diện màn đua mùa hè

- Giao diện chọn nhân vật cho từng đường đua và các nhân vật bên trong.



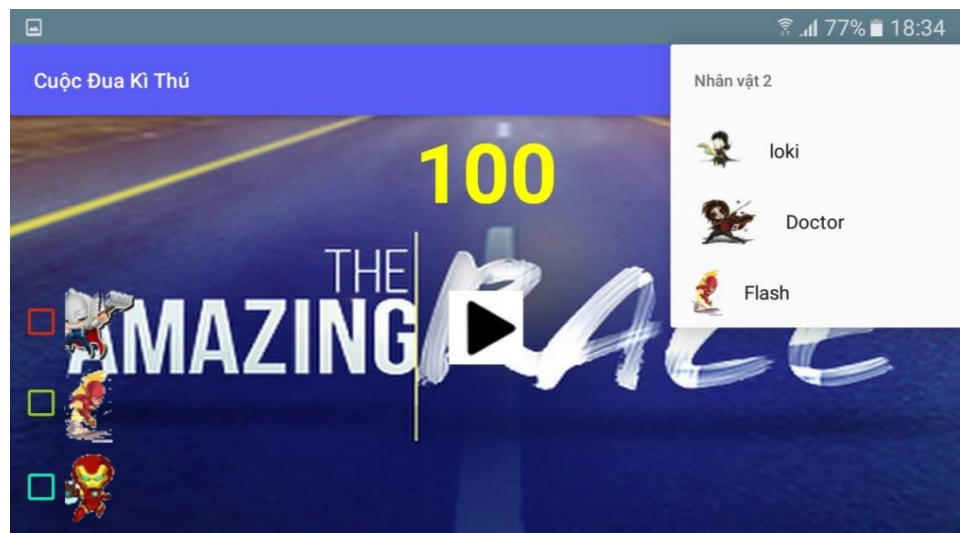
Hình 4.3.2.6. Giao diện chọn nhân vật

○ Nhân vật 1



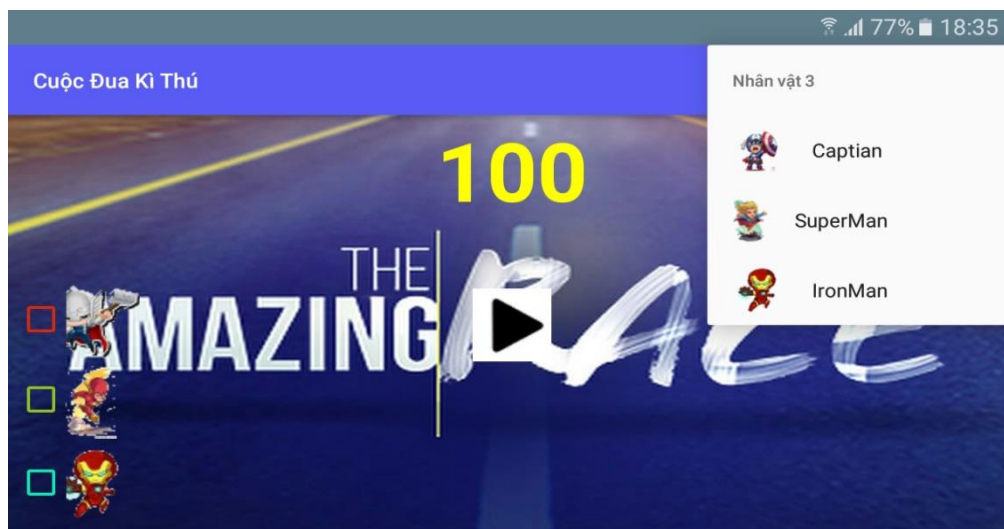
Hình 4.3.2.7. Giao diện nhân vật 1

○ Nhân vật 2



Hình 4.3.2.8. Giao diện nhân vật 2

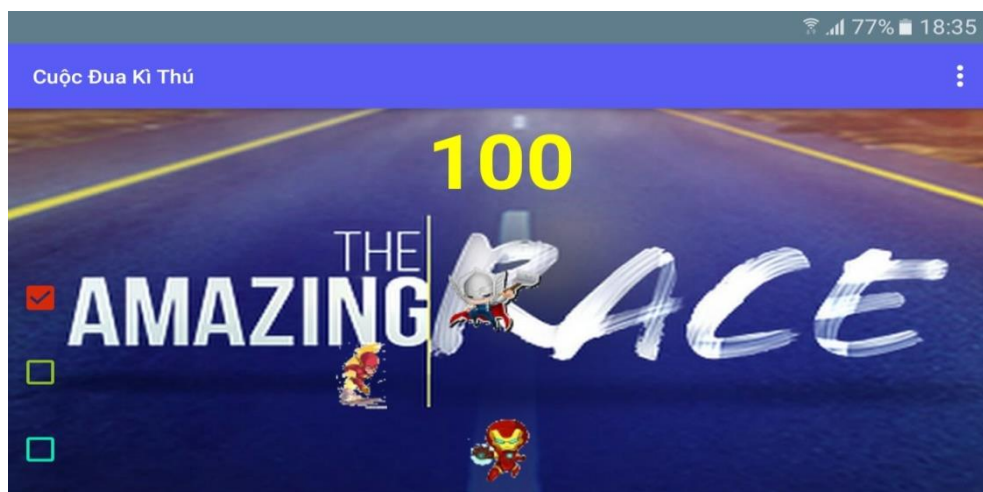
○ Nhân vật 3



Hình 4.3.2.9. Giao diện nhân vật 3

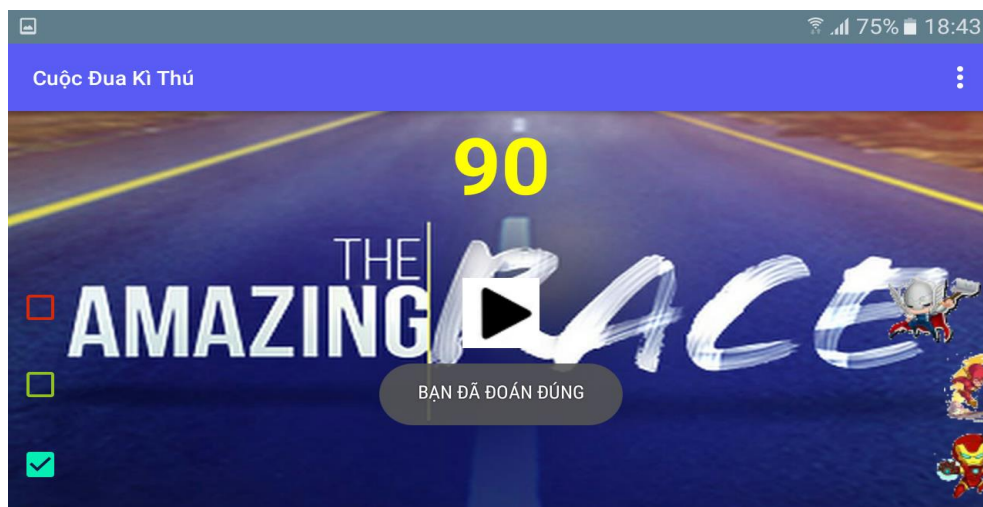
3.4.3. Giao diện khi chạy đua

- Giao diện đang chạy đua



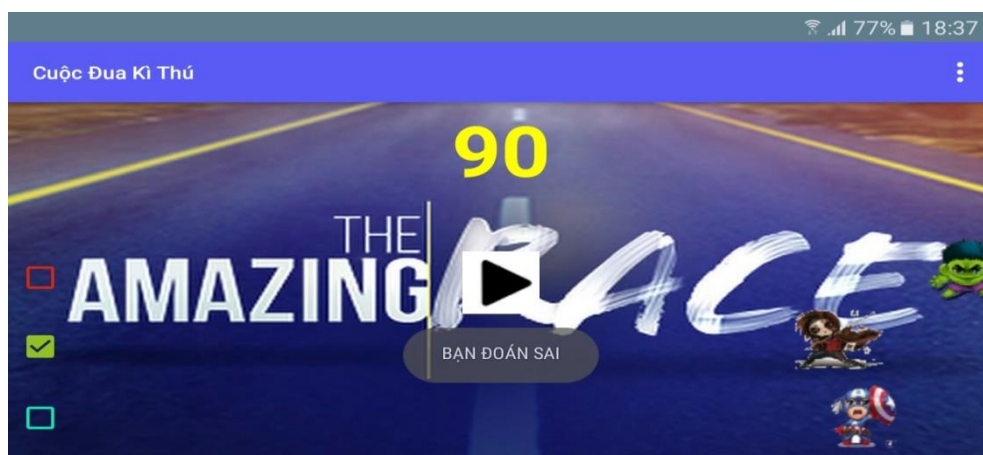
Hình 4.3.3.1. Giao diện khi đang chạy đua

- Giao diện khi người chơi chọn đúng



Hình 4.3.3.2. Giao diện khi bạn chọn đúng

- Giao diện khi người chơi chọn sai



Hình 4.3.3.2. Giao diện khi bạn chọn sai

KẾT LUẬN

Trên đây là giới thiệu về mặt lý thuyết và thực nghiệm về với việc lập trình ứng dụng Cuộc Đua Kì Thú. Đồ án hướng tới mục tiêu xây dựng một ứng dụng nhỏ, có khả năng chạy trên các thiết bị thật; qua đó tìm hiểu lập trình trên Android và có thể phát triển xây dựng ứng dụng khác – có tính thực tế cao. Vì thực hiện đề tài trong khoảng thời gian nhất định, nên còn nhiều vấn đề chưa hoàn chỉnh. Tuy nhiên, đồ án cũng đạt được một số kết quả:

- Về mặt lý thuyết: Tìm hiểu, nghiên cứu cách thức lập trình trên Android Studio. Tìm hiểu cách thức xây dựng một ứng dụng.
- Về thực nghiệm: Sử dụng kiến thức tìm hiểu được, xây dựng thành công một ứng dụng nhỏ: game Cuộc Đua Kì Thú

Do thời gian có hạn nên em mới chỉ nghiên cứu được một vài thành phần chính để tham gia xây dựng ứng dụng trên Android.

TÀI LIỆU THAM KHẢO

- [1] [https://vi.wikipedia.org/wiki/Android_\(h%E1%BB%87_%C4%91i%E1%B%81u_h%C3%A0nh\)](https://vi.wikipedia.org/wiki/Android_(h%E1%BB%87_%C4%91i%E1%B%81u_h%C3%A0nh))
- [2] <http://vforum.vn/diendan/showthread.php?106077-Android-va-su-phat-trien-cua-minh-thong-qua-tung-phien-ban-duoc-phat-hanh>
- [3] <https://xuanthulab.net/cai-dat-android-studio-may-ao-android-tao-du-ang-dung-dau-tien.html>
- [4] <https://code.tutsplus.com/vi/tutorials/android-from-scratch-understanding-views-and-view-groups--cms-26043>
- [5] <https://androidcoban.com/bai-4-xu-ly-su-kien-event-handling-trong-android.html>
- [6] <https://zmikisoft.com/cac-thanh-phan-trong-ung-dung-android/>