

**ĐỒ ÁN MÔN HỌC****LẬP TRÌNH GAME**

**Ngành: Công Nghệ Thông Tin**

**Môn Học: Lập Trình Trên Thiết Bị Di Động**

**Giảng viên hướng dẫn: Thầy Huỳnh Nam**

Sinh viên thực hiện	MSSV	Lớp
1. Nguyễn Cát Tiên	1511060277	15DTH09
2. Trương Quang Học	1511060492	15DTH09
3. Nguyễn Minh Hoàng	1511060200	15DTH09
4. Từ Anh Hoàng	1511060309	15DTH09
5. Mai Tiến Dũng	1511060873	15DTH09

TP. Hồ Chí Minh, Ngày 17 tháng 12 năm 2018

# MỤC LỤC

LỜI MỞ ĐẦU .....	4
LỜI CẢM ƠN .....	5
NHẬN XÉT CỦA GIÁO VIÊN .....	5
CHƯƠNG 1. TỔNG QUAN.....	6
1.1 Nhiệm vụ đồ án .....	6
1.2 Quy tắc game hoạt động.....	6
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT .....	7
2.1 Giới thiệu về ngôn ngữ Java .....	7
2.1.1 Lịch sử phát triển .....	7
2.1.2 Khả năng của ngôn ngữ Java .....	7
2.1.3 Những đặc điểm của ngôn ngữ Java.....	7
2.2 Android Studio .....	8
2.2.1 Lý thuyết .....	8
2.2.2 Tải Android Studio.....	9
2.2.3 Cài đặt Android Studio .....	9
CHƯƠNG 3. QUÁ TRÌNH THỰC HIỆN.....	11
3.1 Xây dựng cấu trúc 2D .....	11
3.2 Tạo game loop.....	11
3.3 Test game loop .....	11
3.4 Truyền GamPanel cho MainThread.....	12
3.5 Hiển thị hình tại vị trí chạm .....	12
3.6 Đưa việc xử lý hình ảnh thành đối tượng .....	12
3.7 Parallax scrolling – nền chuyển động cho game .....	12
3.8 Chuyển động và tập hợp đối tượng.....	12
3.9 Tạo class bullet (viên đạn) .....	13
3.10 vẽ nhiều viên đạn bay tại vị trí của element (phithuyen) ...	13

3.11	Xây dựng thời gian nạp đạn giữa 2 lần bắn.....	13
3.12	Tạo kẻ thù .....	13
3.13	Va chạm .....	14
3.14	Nhạc nền .....	14
3.15	Âm thanh va chạm .....	14
CHƯƠNG 4. KẾT QUẢ THỰC HIỆN .....		15
CHƯƠNG 5. KẾT LUẬN.....		16
5.1	Những vấn đề khi thực hiện đồ án .....	16
5.2	Hướng phát triển .....	16
TÀI LIỆU THAM KHẢO .....		17

## LỜI MỞ ĐẦU

Nói đến game thì chắc phần lớn mọi người đều đồng ý đó là một loại hình giải trí có sức thu hút rất lớn, nhất là với giới trẻ hiện nay. Đặc biệt sau sự ra đời của Internet, game online xuất hiện và nhanh chóng gây được sức hấp dẫn to lớn. Chúng ta có thể chơi game online trên mạng, các game offline trên PC, các game trên các dòng máy chuyên như PS2, PS3 . .

Hiện nay game cũng đang được phát triển mạnh trên các thiết bị di động như điện thoại di động, các máy Xbox. Ở các nước phát triển như Nhật Bản, Mỹ có rất nhiều các công ty chuyên về sản xuất game như Nintendo, Sony, Sega và mỗi năm các công ty này đạt doanh thu hàng tỷ USD. Hay với các nước đi sau như Hàn Quốc, Trung Quốc cũng có được một ngành công nghệ game rất mạnh nhờ có sự đầu tư đúng đắn và chiến lược phát triển tốt. Mặc dù VINASA xếp game là một trong 3 định hướng công nghệ chiến lược của hiệp hội này trong giai đoạn 2006 - 2010, bên cạnh định hướng phần mềm nhúng và ERP ( bộ phần mềm hoạch định nguồn lực doanh nghiệp ) nhưng hiện tại thị trường game ở Việt Nam còn rất yếu. Chưa có nhiều công ty chuyên về lập trình game, những công ty phát triển game online thì chủ yếu là cung cấp các game nhập từ Hàn Quốc, Trung Quốc, các game nội hầu như vắng bóng trên thị trường. Đúng như theo nhận định của một chuyên gia phát triển game của Hàn Quốc “ ngành công nghiệp game Việt Nam hiện nay giống như một đứa trẻ sơ sinh, rất cần được sự giúp đỡ và nuôi nấng (của Chính phủ) để phát triển ”. Vì vậy nhóm em đã chọn chủ đề về game để nghiên cứu và phát triển cũng như làm đồ án môn học “Lập Trình Trên Thiết Bị Di Động”

## **LỜI CẢM ƠN**

Đôi lời cảm ơn thầy Huỳnh Nam đã đọc và chấm điểm đồ án của nhóm chúng em. Nhóm em đã dành rất nhiều thời gian để tham khảo trên mạng về những phần chưa biết. Dù đã có những cố gắng để hoàn thành đề tài theo đúng thời gian yêu cầu, nhưng do hạn chế về kinh nghiệm tự nghiên cứu và soạn thảo, nên chắc chắn còn nhiều khiếm khuyết về cả nội dung và hình thức trình bày trong bài báo cáo này. Em rất mong nhận được sự đánh giá và cho ý kiến của thầy Huỳnh Nam.

Cảm ơn thầy vì đã hướng dẫn nhóm chúng em từng bước hoàn thành đồ án !!

## **NHẬN XÉT CỦA GIÁO VIÊN**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# **CHƯƠNG 1. TỔNG QUAN**

## **1.1 NHIỆM VỤ ĐỒ ÁN**

Nhằm nắm vững kiến thức về môn học, nhóm chúng em thực hiện đề tài lập trình game android du hành không gian. Với các mục tiêu chính là tìm hiểu nguyên lý và cài đặt xây dựng trò chơi trên môi trường java dựa trên nền tảng ngôn ngữ C++, được thực hiện trên trình biên dịch Android Studio.

Đề tài của nhóm là tìm hiểu nguyên lý và xây dựng trò chơi du hành không gian.

## **1.2 QUY TẮC GAME HOẠT ĐỘNG**

Mục tiêu của game du hành không gian là di chuyển chiếc máy bay lên, xuống, qua phải, qua trái để máy bay bắn đạn vào các chướng ngại vật và phá huỷ chúng phá huỷ chúng, chúng ta sẽ nhận được tiền thưởng từ việc phá huỷ những chướng ngại vật, nếu máy bay chạm vào các chướng ngại vật thì game kết thúc.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1 GIỚI THIỆU VỀ NGÔN NGỮ JAVA

#### 2.1.1 Lịch sử phát triển

Năm 1990, Sun Microsystems thực hiện dự án Green nhằm phát triển phần mềm trong các thiết bị dân dụng. James Gosling, chuyên gia lập trình đã tạo ra một ngôn ngữ lập trình mới có tên là Oak. Ngôn ngữ này có cú pháp gần giống như C++ nhưng bỏ qua các tính năng nguy hiểm của C++ như truy cập trực tiếp tài nguyên hệ thống, con trỏ, định nghĩa chồng các tác tử... Khi ngôn ngữ Oak trưởng thành, WWW cũng đang vào thời kỳ phát triển mạnh mẽ, Sun cho rằng đây là một ngôn ngữ thích hợp cho Internet. Năm 1995, Oak đổi tên thành Java và sau đó đến 1996 Java đã được xem như một chuẩn công nghiệp cho Internet.

#### 2.1.2 Khả năng của ngôn ngữ Java

- Là một ngôn ngữ bậc cao như C, C++, Perl, SmallTalk,.. cho nên có thể được dùng để tạo ra các ứng dụng để giải quyết các vấn đề về số, xử lý văn bản, tạo ra trò chơi, và nhiều thứ khác.
- Có các môi trường lập trình đồ họa như Visual Java, Symantec Cafe, Jbuilder, Jcreator,...
- Có khả năng truy cập dữ liệu từ xa thông qua cầu nối JDBC (Java DataBase Connectivity)
- Hỗ trợ các lớp hữu ích, tiện lợi trong lập trình các ứng dụng mạng (Socket) cũng như truy xuất Web.
- Hỗ trợ lập trình phân tán (Remote Method Invocation ) cho phép một ứng dụng có thể được xử lý phân tán trên các máy tính khác nhau.
- Và luôn được bổ sung các tính năng cao cấp khác trong các phiên bản sau.

#### 2.1.3 Những đặc điểm của ngôn ngữ Java

- Ngôn ngữ hoàn toàn hướng đối tượng.
- Ngôn ngữ đa nền cho phép một chương trình có thể thực thi trên các hệ điều hành khác nhau (MS Windows, UNIX, Linux) mà không phải biên

dịch lại chương trình. Phương châm của java là "Viết một lần , Chạy trên nhiều nền" (Write Once, Run Anywhere).

- Ngôn ngữ đa luồng, cho phép trong một chương trình có thể có nhiều luồng điều khiển được thực thi song song nhau, rất hữu ích cho các xử lý song song
- Ngôn ngữ phân tán, cho phép các đối tượng của một ứng dụng được phân bố và thực thi trên các máy tính khác nhau.
- Ngôn ngữ động, cho phép mã lệnh của một chương trình được tải từ một máy tính về máy của người yêu cầu thực thi chương trình.
- Ngôn ngữ an toàn, tất cả các thao tác truy xuất vào các thiết bị vào ra đều thực hiện trên máy ảo nhờ đó hạn chế các thao tác nguy hiểm cho máy tính thật.
- Ngôn ngữ đơn giản, dễ học, kiến trúc chương trình đơn giản, trong sáng.

## **2.2 ANDROID STUDIO**

### *2.2.1 Lý thuyết*

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

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



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

### 2.2.2 *Tải Android Studio*

Trước khi tiến hành cài đặt Android Studio bạn cần tải Android Studio về máy bằng cách truy cập vào đường link ở phần tài liệu tham khảo và chọn phiên bản Android Studio phù hợp với hệ điều hành mà bạn đang sử dụng. Trang tải Android Studio ở trên có sử dụng một đoạn mã JavaScript được để phát hiện ra hệ điều hành máy tính của bạn và cung cấp phiên bản Android Studio phù hợp. Và do đó bạn chỉ việc nhấp vào link sẵn có trên trang để tải về.

### 2.2.3 *Cài đặt Android Studio*

#### 2.2.3.1 *Hệ Điều Hành Windows*

Việc cài đặt Android Studio trên Windows khá đơn giản. Sau khi tải xong bạn nhấp đúp vào file cài đặt để tiến hành cài đặt. Quá trình cài đặt không có gì đặc biệt và bạn có thể tự mình hoàn tất công đoạn này. Sau khi cài đặt xong máy tính sẽ khởi động chương trình Android Studio để bạn có thể bắt đầu phát triển ứng dụng đầu tiên.

#### 2.2.3.2 *Hệ Điều Hành Mac*

Tương tự như Windows việc cài đặt Android Studio trên Mac OS X cũng khá đơn giản. Sau khi tải về tập tin DMG dùng để cài đặt Android Studio trên Mac thì chúng ta sẽ nhấp đúp vào file này. Ở cửa sổ bật ra bạn cần nhấp chuột và kéo Android Studio vào thư mục Applications. Sau đó click đúp vào biểu tượng Android Studio để tiến hành cài đặt. Quá trình cài đặt không có gì đặc biệt và bạn có thể tự mình hoàn tất công đoạn này.

#### 2.2.3.3 *Hệ Điều Hành Linux*

Để cài đặt Android Studio trên Linux thì sau khi tải tập tin .zip về máy bạn thực hiện các bước sau:

Bước 1: Giải nén tập tin tải về sử dụng chương trình bằng cách nhấp đúp lên tập tin này. Chương trình Archive Manager có sẵn trên Ubuntu sẽ được khởi động để hỗ trợ bạn thực hiện việc giải nén.

Bước 2: Chép thư mục vừa được giải nén vào bên trong thư mục `/usr/local` hoặc `/opt`

Bước 3: Mở cửa sổ dòng lệnh terminal(sử dụng tổ hợp phím tắt `Ctrl + Alt + T`).

Bước 4: Di vào bên trong thư mục android-studio/bin của tập tin vừa được giải nén (sau bước 3 ở trên thì thư mục này bây giờ sẽ nằm bên trong /usr/local hoặc /opt).

Bước 5: Chạy tập tin studio.sh trong thư mục bin trên bằng cách nhập tên tập tin này và gõ Enter.

Sau bước 5, máy tính sẽ khởi động chương trình cài đặt Android Studio Setup Wizard và bạn có thể dễ dàng thực hiện các công đoạn còn lại để cài Android Studio.

Sau khi cài đặt xong Android Studio thì đối với hệ điều hành Linux 64-bit thì bạn cần tiến hành thêm một bước sau để cài đặt bổ sung một số bộ thư viện cần thiết cho Android Studio:

Với các máy tính chạy phiên bản 64bit của Ubuntu thì chúng ta sẽ cần chạy câu lệnh:

```
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 libbz2-1.0:i386
```

Với các máy tính chạy phiên bản 64-bit của Fedora thì chúng ta sẽ cần chạy câu lệnh:

```
sudo yum install zlib.i686 ncurses-libs.i686 bzip2-libs.i686
```

## CHƯƠNG 3. QUÁ TRÌNH THỰC HIỆN

Cấu trúc game cơ bản như sau:

- Xử lý đầu vào (input).
- Cập nhật trạng thái bên trong.
- Render ra màn hình (có thêm phần phát âm thanh, rung....)

Ba việc này lặp đi lặp lại tạo thành loop game. Update và Render được nhóm lại 1 cách logic.

Tất cả mọi thứ trong Android đều xảy ra trong Activity. Activity sẽ tạo ra một View. View là nơi mọi thứ sẽ xảy ra. Nó là nơi nhận việc chạm tay và cũng là nơi hiển thị hình ảnh và dữ liệu. Tưởng tượng Activity là một vật chứa, chứa 1 tờ giấy (View), trên tờ giấy ta có thể vẽ nhiều thứ. Ta sẽ dùng bút chì để vẽ lên tờ giấy. Chạm tay là một hành động vật lý xảy ra trên tờ giấy và kết quả của sự tương tác trên view sẽ là hình ảnh

### 3.1 XÂY DỰNG CẤU TRÚC 2D

SurfaceView là một bề mặt để vẽ và có thể nhúng vào các View. Ta có thể điều khiển kích thước bề mặt vẽ này. Surface như 1 bề mặt được đặt chồng lên trên một cửa sổ (cửa sổ này là view chứa nó).

Việc truy cập đến bề mặt vẽ này được cung cấp thông qua interface tên SurfaceHolder, truy xuất bằng cách gọi getHolder(). Thực thi từ interface tên SurfaceHolder.Callback để truy xuất thông tin về sự thay đổi của bề mặt vẽ.

### 3.2 TẠO GAME LOOP

Tạo ra một thread mà nó sẽ là game loop . Tạo một class mới tên MainThread kế thừa từ thread. Trong class vừa tạo tạo ra một biến tên “running” để lưu trạng thái của thread. True là đang chạy. Viết một hàm setRunning để gán lại biến “running”. Tiếp theo override lên hàm run.

### 3.3 TEST GAME LOOP

- Trong hàm run ta tạo ra biến đếm và trong vòng lặp while ta cho biến đếm tăng lên rồi dùng lệnh Log.d để xuất ra Logcat.
- Tương tự bên file “cautrucobanActivity.java” trong hàm onDestroy ta cũng xuất ra logcat để test khi kết thúc chương trình.

- Chạy và xem kết quả trên Logcat bằng cách Trên Eclipse chọn Window -> Show View -> LogCat để mở hộp thoại LogCat. Chạy chương trình và xem kết quả loop liên tục trên logcat. Tắt chương trình để thấy onDestroy được chạy.

### **3.4 TRUYỀN GAMEPANEL CHO MAINTHREAD**

Trong Mainthread ta cần phải có GamePanel để điều khiển và truy xuất cũng như khóa vùng vẽ do đó ta cần phải truyền gamepanel vào cho Mainthread.

File MainThread cần phải có gamePanel để có thể khóa surface khi vẽ lên, điều đó cần phải thông qua surfaceholder. Thật ra ta có thể không cần surfaceholder truyền vào vì khi có gamepanel ta vẫn có thể dùng gamepanel.getHolder() để truy đến surfaceholder của nó.

### **3.5 HIỂN THỊ HÌNH TẠI VỊ TRÍ CHẠM**

Khi ta chạm tay vào thì nơi đó xuất hiện tọa độ mX, mY.

Nhưng ta thấy rằng ta chạm vào nhiều vị trí thì hình của ta vẫn giữ lại do vẫn vẽ tiếp lên vùng vẽ trước đó. Do đó để vẽ lại ta phải vẽ lại toàn bộ. Ở đây đơn giản ta chỉ cần vẽ canvas lại với vùng màu đen.

### **3.6 ĐƯA VIỆC XỬ LÝ HÌNH ẢNH THÀNH ĐỐI TƯỢNG**

Ta thấy rằng mã lệnh của class thật ra rất giống phần ta đã bỏ nhưng được tách ra để xây thành đối tượng. Ở đây ta viết hàm doDraw đảm trách việc vẽ đối tượng do đó hàm onDraw ở GamePanel giờ chỉ quản lý việc vẽ chứ không trực tiếp vẽ nữa.

### **3.7 PARALLAX SCROLLING – NỀN CHUYỂN ĐỘNG CHO GAME**

Parallax scrolling 1982 Irem giới thiệu kỹ thuật Parallax scrolling trong game Moon Patrol. Nó là kỹ thuật tạo ra ảo giác của chuyển động. Nó không ngừng phát triển và đến nay nó còn được xây dựng trên web với sự hỗ trợ của HTML5 và CSS3.

Trong game và các ứng dụng 2D, kỹ thuật parallax scrolling dùng nhiều lớp đặt chồng lên nhau và từ từ duy chuyển qua vùng nhìn thấy. Mỗi lớp sẽ di chuyển với vận tốc khác nhau tạo thành hiệu ứng chiều sâu.

### **3.8 CHUYỂN ĐỘNG VÀ TẬP HỢP ĐỐI TƯỢNG**

Đối với việc chuyển động 1 đối tượng, thread chạy liên tục, các hàm vẽ cũng được chạy liên tục do đó khi mỗi lần vẽ ta chỉ cần cập nhật lại tọa độ x,y của đối tượng để tạo thành chuyển động.

Khi có nhiều đối tượng của cùng một class (vd: Các viên đạn được bắn ra) ta nên đưa nó vào 1 tập hợp (vd: dùng ArrayList) để dễ quản lý, và khi một đối tượng không còn dùng nữa (vd:viên đạn đã bay ra khỏi màn hình) ta sẽ xóa nó khỏi tập hợp. Cơ chế dọn rác của java sẽ dọn nó.

### **3.9 TẠO CLASS BULLET (VIÊN ĐẠN)**

Class này có 2 biến x,y dùng để điều khiển tọa độ x,y cho đối tượng. Biến bitmap dùng để lưu hình. Đặc biệt là biến tốc độ, do việc vẽ được thực hiện liên tục khi ta vẽ viên đạn ra ta cần cập nhật lại tọa độ cho viên đạn, nếu tăng 1 sẽ rất chậm do đó ta tạo ra biến tốc độ để mỗi lần vẽ đối tượng được cập nhật theo tốc độ để viên đạn bay nhanh hơn.

### **3.10 VẼ NHIỀU VIÊN ĐẠN BAY TẠI VỊ TRÍ CỦA ELEMENT (PHITHUYEN)**

Do có nhiều viên đạn được bắn ra nên ta phải tạo ra một tập hợp các viên đạn. Ở đây ta sẽ dùng ArrayList.

### **3.11 XÂY DỰNG THỜI GIAN NẠP ĐẠN GIỮA 2 LẦN BẮN**

Để tạo độ trễ giữa 2 lần bắn thì ta sẽ tạo ra 1 biến. sau đó mỗi lần vẽ lại ta tăng biến này lên. Trong hàm vẽ các bullet ta sẽ xét biến này khi nó bằng 10 (tức là đã vẽ lại 10 lần) thì ta mới tạo đối tượng bullet mới và cập nhật biến về 0.

### **3.12 TẠO KẺ THÙ**

Ta sẽ tạo ra các kẻ thù được chạy ra từ bên phải-> trái, các kẻ thù này có hình khác nhau và tốc độ chạy cũng khác nhau. (thật ra nó hơi giống với lớp viên đạn).

Tạo ra các biến toàn cục. Biến gồm x,y là tọa độ của kẻ thù, hình của kẻ thù và đặc biệt là 2 mảng tốc độ và mảng hình vì mỗi lần tạo một đối tượng kẻ thù ta sẽ lấy ngẫu nhiên để ra một hình và tốc độ của nó trong mảng.

Trong hàm ta tạo ngẫu nhiên sau đó lấy hình theo ngẫu nhiên. Tọa độ x của kẻ thù bằng chiều rộng canvas (bắt đầu từ phải), tọa độ y là lấy ngẫu nhiên trong chiều cao canvas. Trong hàm doDraw ta sẽ vẽ đối tượng và mỗi lần vẽ ta sẽ cập nhật lại trục x, bằng cách giảm x đi (vì chạy từ phải -> trái). Các hàm còn lại linh tinh không dùng đến.

### **3.13 VA CHẠM**

Ta xét kích thước từ tâm của vật thể A đến tâm của vật thể B theo trục  $x$  và kích thước từ tâm của vật thể A đến tâm của vật thể B theo trục  $y$

### **3.14 NHẠC NỀN**

Trong quá trình chơi game sẽ có nhạc nền phát cùng

### **3.15 ÂM THANH VA CHẠM**

Khi đạn đạn bay trúng vào vật thể tạo ra âm thanh nổ và khi element va chạm vào vật thể có âm thanh báo va chạm

## CHƯƠNG 4. KẾT QUẢ THỰC HIỆN



Hình 4.1 Giao diện game

## **CHƯƠNG 5. KẾT LUẬN**

### **5.1 NHỮNG VẤN ĐỀ KHI THỰC HIỆN ĐỒ ÁN**

Thuận lợi:

- Mã nguồn game đã được hướng dẫn rõ ràng
- Các hướng dẫn đều có trên Youtube trực quan, dễ hiểu.
- Đề tài hấp dẫn, tạo nhiều ý tưởng khi phát triển.

Khó khăn:

- Mới tiếp cận với lĩnh vực mới nên có nhiều trở ngại về kiến thức

### **5.2 HƯỚNG PHÁT TRIỂN**

- Chưa thông báo được điểm cao nhất
- Chưa có hướng dẫn



## TÀI LIỆU THAM KHẢO

1. Youtube

URL:< <https://www.youtube.com/watch?v=wFnDXhnRFAM>>

2. Bài lab hướng dẫn

URL:<<https://www.mediafire.com/file/z8t43k7a0wylt9b/laptrinhgame.zip/file?fbclid=IwAR3F7iLRN5wi5qvwzP5WGW94wPN3zJT4p436LLb1kxqub2xE6vv2bPukB1s>>

3. Download Android Studio

URL:< <https://developer.android.com/studio/>>