**ĐẠI HỌC ĐÀ NẴNG**



**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO ĐỒ ÁN**

**CƠ SỞ NGÀNH MẠNG**

*Đề tài:*

*Xây dựng ứng dụng chơi cờ tướng*

*Giảng viên hướng dẫn:* ***Lê Trần Đức***

*Sinh viên thực hiện: Phạm Nhật Khánh*

*Lớp: 17T3*

***Đà Nẵng, 11/2020***

**MỤC LỤC**

[**Chương 1. CƠ SỞ LÝ THUYẾT** 2](#_Toc55463187)

[**1.1.** **Giới thiệu về hệ điều hành android** 2](#_Toc55463188)

[**1.1.1.** **Tổng quan** 2](#_Toc55463189)

[**1.1.2.** **Kiến trúc ứng dụng** 3](#_Toc55463190)

[**1.1.3.** **Chu kỳ sống của ứng dụng** 4](#_Toc55463191)

[**1.1.4.** **Các tiến trình** 5](#_Toc55463192)

[**1.1.5.** **Công cụ lập trình** 5](#_Toc55463193)

[**1.2.**  **Tổng quan về mạng máy tính** 6](#_Toc55463194)

[**1.2.1.** **Khái niệm** 6](#_Toc55463195)

[**1.2.2.** **Các thành phần** 6](#_Toc55463196)

[**1.2.3.** **Mô hình OSI** 6](#_Toc55463197)

[**1.2.4.** **Giao thức TCP/IP** 7](#_Toc55463198)

[**1.3.** **Giới thiệu về lập trình socket** 8](#_Toc55463199)

[**1.3.1.** **Khái niệm về socket** 8](#_Toc55463200)

[**1.3.2.** **Cơ chế giao tiếp** 8](#_Toc55463201)

[**1.3.3.** **Các chế độ giao tiếp** 9](#_Toc55463202)

[**1.4.** **Lập trình luồng (Thread)** 9](#_Toc55463203)

[**1.5.** **Giới thiệu về cờ tướng** 9](#_Toc55463204)

[**Chương 2. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH** 11](#_Toc55463205)

[**2.1.** **Server** 11](#_Toc55463206)

[**2.2.** **Client** 11](#_Toc55463207)

[**2.3.** **Xây dựng bàn cờ** 11](#_Toc55463208)

[**2.4.** **Giải thuật** 12](#_Toc55463209)

# **Chương 1. CƠ SỞ LÝ THUYẾT**

## **1.1. Giới thiệu về hệ điều hành android**

### **1.1.1. Tổng quan**

Android là một hệ điều hành trên điện thoại, nhưng Android còn được cho là một nền tảng mở cho thiết bị di động (gồm hệ điều hành, middleware và một số ứng dụng cơ bản) được phát triển bởi Google và dựa trên nền tảng Linux. Android là một sản phẩm kết tinh từ ý tưởng của Khối Liên minh thiết bị cầm tay mở do Google dẫn đầu, với thành viên là các công ty hàng đầu về công nghệ và di động toàn cầu như Qualcomm, Intel, Motorola, Texas Instruments và LG Electronics, các nhà mạng như T-Mobile, Sprint Nextel, NTT DoCoMo và China Mobile. Google công bố hầu hết các mã nguồn của Android theo bản cấp phép Apache. Hệ điều hành Android bao gồm 12 triệu dòng mã; 3 triệu dòng XML, 2.8 triệu dòng mã C, 2.1 triệu mã Java và 1.75 triệu dòng mã C++.

Dựa trên ngôn ngữ Java và có thể sử dụng miễn phí bộ Android Software Development Kit (SDK), Android có cả một cộng đồng những nhà phát triển rất lớn viết các ứng dụng cho hệ điều hành của mình.

Về cơ bản Android sẽ hỗ trợ các chức năng sau:

- Màn hình cảm ứng (đa điểm).

- 3G, wifi.

- Trình duyệt dựa trên webkit.

- Tin nhắn (SMS) theo luồng.

- Định dạng MPEG-4, H.264, MP3, AAC.

- Bộ tăng tốc đồ họa 3D.

Android là hệ điều hành mở, do liên minh của Google phát triển (Google trực tiếp). Android đang lớn mạnh từng ngày và có khả năng sẽ vuợt mặt Iphone OS trong dòng Smartphone. Bởi tốc độ xử lý nhanh, bộ nhớ SDCard lớn cho phép thực hiện công việc hay giải trí, nên điện thoại đã dần dần thay thế cho máy tính.

Hệ điều hành Android với lợi thế là một hệ điều hành mã nguồn mở, được hỗ trợ bởi Google và đông đảo cộng đồng phát triển nên Android đang trở thành xu hướng phát triển của các hãng di động như HTC, LG, Samsung, Motorola, Kyocera, Sony Ericsson. Trong bối cảnh thị trường Smartphone sôi động như hiện nay, các hệ điều hành luôn cạnh tranh với nhau để giành lấy thị trường. Hệ điều hành có thị phần cao nhất hiện nay là Android.

Thêm vào đó, xu hướng công nghệ hiện nay đang coi Android là nền tảng để phát triển ứng dụng cho các thiết bị di động. Một bức tranh phác thảo toàn cảnh về Android cho thấy Android ra đời đã mang lại cho người dùng điện thoại sự cảm nhận khác biệt về một chiếc Smartphone.

### **1.1.2. Kiến trúc ứng dụng**

Kiến trúc Android gồm 4 phần chính. Nhân của Android được phát triển dựa vào Kernel Linux 2.6. Mỗi tầng trong kiến trúc Android hoạt động dựa vào tầng bên dưới nó.

*a. Linux Kernel*

Linux Kernel là lớp dưới cùng trong kiến trúc của hệ điều hành Android. Linux Kernel được phát triển từ nhân Linux phiên bản 2.6 với khoảng 115 bản vá có các chức năng cơ bản là quản lý bộ nhớ, giao tiếp phần cứng, thực hiện bảo mật, quản lý tiến trình. Linux Kernel còn có chức năng tạo lớp giao tiếp ảo với các tầng trên của nó.

*b. Library và Android Runtime*

Library: chứa một tập hợp các thư viện được viết từ ngôn ngữ C/C++ để cung cấp cho các thành phần khác của Android.

Android Runtime: là môi trường thực thi ứng dụng Android. Thực chất của Android Runtime là một máy ảo java để chạy các ứng dụng Java được lập trình.

*c. Application Framework*

Application Framework là thành phần phía trên của Library và Android Runtime với chức năng cung cấp một nền tảng mở và mạnh mẽ để phát triển ứng dụng Java: - View UI: để xây dựng layout của ứng dụng bao gồm các đối tượng như ListView, TextView, EditText, Button, Dialog,..

- Content Providers: cho phép các ứng dụng truy cập dữ liệu từ các ứng dụng khác hoặc để chia sẻ dữ liệu của riêng ứng dụng.

- Resource Manager: cung cấp cách thức truy cập đến non-code resources như các asset, graphic, image, music, video…

- Notification Manager: cho phép tất cả các ứng dụng hiển thị thông báo trên hệ điều hành.

- Activity Manager: có chức năng quản lý vòng đời của các ứng dụng.

*d. Applications*

Applications là tầng trên cùng trong kiến trúc hệ điều hành Android. Applications là hệ thống phần mềm ứng dụng chạy trên Android. Các phần mềm ứng dụng này viết bằng ngôn ngữ Java và chạy trên máy ảo Dalvik Virtual Machine. Applications còn tích hợp sẵn một số ứng dụng cơ bản như danh bạ, lịch, gọi điện thoại.

### **1.1.3. Chu kỳ sống của ứng dụng**

Trong một ứng dụng Android có chứa nhiều thành phần và mỗi thành phần đều có một chu trình sống riêng. Ứng dụng chỉ được gọi là kết thúc khi tất cả các thành phần trong ứng dụng kết thúc. Activity là một thành phần cho phép người dùng giao tiếp với ứng dụng, còn các thành phần không có khả năng tương tác với người dùng như là Service, BroadcastReceiver.

Nếu một Activity được tạm dừng hoặc dừng hẳn, hệ thống có thể bỏ thông tin của nó từ vùng nhớ bởi việc finish(), hoặc đơn giản “kill” tiến trình của nó. Khi nó được hiển thị lần nữa, nó sẽ phải được restart và phục hồi lại trạng thái trước. Khi một Activity chuyển đổi qua lại giữa các trạng thái, nó phải báo việc chuyển của nó bằng việc gọi hàm transition.

Tất cả các phương thức có liên quan chặt chẽ với nhau, có thể ghi đè lên nhau (override) để làm tương thích công việc trong ứng dụng khi thay đổi trạng thái. Tất cả các Activity bắt buộc phải có phương thức onCreate() để khởi tạo ứng dụng. Nhiều Activity sẽ cũng thực hiện onPause() ví dụ như để xác nhận việc thay đổi dữ liệu hoặc chuẩn bị dừng hoạt động.

### **1.1.4. Các tiến trình**

Android có cơ chế quản lý các tiến trình (process) theo chế độ ưu tiên (priority). Các tiến trình có độ ưu tiên thấp sẽ bị giải phóng mà không hề có cảnh báo nhằm đảm bảo tài nguyên.

- Foreground process: là tiến trình của ứng dụng hiện thời đang được người dùng tương tác.

- Visible process: là tiến trình của ứng dụng mà Activity đang hiển thị đối với người dùng (phương thức onPause() của Activity được gọi).

- Service process: là Service đang hoạt động (running).

- Background process: là tiến trình của ứng dụng mà các Activity của nó không hiển thị với người dùng (phương thức onStoped() của Activity được gọi).

- Empty process: tiến trình không có bất cứ thành phần nào hoạt động. Theo chế độ ưu tiên thì khi cần tài nguyên, Android sẽ tự động đóng tiến trình (kill process), trước tiên là các Empty process.

### **1.1.5. Công cụ lập trình**

Ngôn ngữ lập trình: Ứng dụng Android đa số được viết trên ngôn ngữ Java, bên cạnh đó, ứng dụng trên hệ điều hành Android cũng có thể được viết bằng một số ngôn ngữ khác như C/C++, C#, Python và Kotlin.

Quy trình phát triển phần mềm ứng dụng:

- Bước 1. Thiết kế (Design).

- Bước 2. Viết mã (Code).

- Bước 3. Biên dịch (Compile).

- Bước 4. Tiên kiểm tra (Preverify).

- Bước 5. Đóng gói (Package): Sinh ra tập tin APK.

- Bước 6. Triển khai (Deploy): Để các tài nguyên của ứng dụng dưới sự quản lý của bộ giả lập (emulator).

- Bước 7. Thực thi (Execute): Chạy ứng dụng với bộ giả lập.

- Bước 8. Gỡ lỗi (Debug): Nhận ra các lỗi và sửa lại mã nguồn.

- Bước 9. Phát hành (Publish): Phát hành ứng dụng tới người dùng.

Android SDK là công cụ phát triển ứng dụng Android. Android SDK có thể được sử dụng để chạy và kiểm tra ứng dụng được phát triển. Thiết bị giả lập Android SDK được trang bị đầy đủ hầu hết các tính năng của một thiết bị thật. Tuy nhiên cũng có một số tính năng không được hỗ trợ như kết nối USB, camera, quay video, tai nghe, giả lập pin và bluetooth. Ngoài ra tốc độ khởi động của thiết bị giả lập thường rất chậm gây mất nhiều thời gian.

Công cụ lập trình: Android Studio.

## **1.2. Tổng quan về mạng máy tính**

### **1.2.1. Khái niệm**

Mạng máy tính là mạng của hai hay nhiều máy tính được nối lại với nhau bằng một kiến trúc vật lý nào đó.

### **1.2.2. Các thành phần**

Mạng máy tính bao gồm 3 thành phần: đường biên mạng, mạng đường trục và mạng truy cập.

- Đường biên mạng (Network Edge): bao gồm các máy tính và ứng dụng được tổ chức theo mô hình Client-Server, Peer2Peer.

- Mạng đường trục (Network Core): là mạng nối các router đảm bảo truyền tin thông suốt giữa hai máy cách xa nhau.

- Mạng truy cập (Access Network): nối các máy tính vào các router ngoài bìa.

### **1.2.3. Mô hình OSI**

Mô hình OSI (Open System Interconnection Model) là mô hình tham khảo được phát triển bởi tổ chức tiêu chuẩn thế giới ISO (International Standard Organization).

Theo mô hình OSI, mạng máy tính được chia ra làm 7 tầng:

- Tầng vật lý (Physical layer): đơn vị truyền nhận dữ liệu là bit (Bits), truyền tải các bit thô trên một kênh truyền vật lý.

- Tầng liên kết dữ liệu (Data link layer): đơn vị truyền nhận dữ liệu là khung (Frames). Tầng liên kết dữ liệu là kênh truyền nối trực tiếp hai máy tính, có vai trò thiết lập cơ chế phát hiện và xử lý lỗi, điều khiển dòng, giải quyết tranh chấp đường truyền. - Tầng mạng (Network layer): đơn vị truyền nhận dữ liệu là gói tin (Packets), có chức năng chính là định tuyến và chuyển tiếp các gói tin, kiểm tra khắc phục tình trạng nghẽn đường truyền.

- Tầng vận chuyền (Transport layer): đơn vị truyền nhận dữ liệu là đoạn (Segments), truyền dữ liệu từ điểm tới điểm, kiểm tra gói tin mất, trùng lắp, đa hợp và phân hợp.

- Tầng giao dịch: đơn vị truyền nhận dữ liệu là dữ liệu (Data), quản lý các giao dịch, đồng bộ hóa dữ liệu. - Tầng trình bày: đơn vị truyền nhận dữ liệu là dữ liệu (Data), chuẩn hóa dữ liệu trao đổi giữa các hệ thống khác nhau. - Tầng ứng dụng: đơn vị truyền nhận dữ liệu là dữ liệu (Data), là các chương trình ứng dụng phần mềm dịch vụ. Ví dụ: email, web, ftp…

### **1.2.4. Giao thức TCP/IP**

*a. Giới thiệu*

Mặc dù mô hình OSI được phân chia rất cụ thể và là mô hình rất nổi tiếng về mạng máy tính nhưng nó chỉ là mô hình tham khảo, không được áp dụng vào thực tế. Như vậy, các mạng máy tính ngày nay làm sao có thể trao đổi dữ liệu, đó là nhờ giao thức TCP/IP – Bộ giao thức chuẩn của mạng máy tính hiện nay. Giao thức TCP/IP (Transmission Control Protocol/Internet Protocol) được phát triển từ mạng ARPANET và Internet và được dùng như giao thức mạng và vận chuyển trên mạng Internet. Họ giao thức TCP/IP hiện nay là giao thức nền tảng được sử dụng rộng rãi nhất để liên kết các máy tính và các mạng.

*b. Các tầng của giao thức TCP/IP*

Giao thức TCP/IP được chia làm 4 tầng:

- Tầng truy cập mạng (Network Access layer) bao gồm một số giao thức như: ATM (Asynchronous Transfer Mode), Ethernet, Token Ring, Frame Relay,…

- Tầng Internet (Internet layer) gồm 4 giao thức:

+ IP (Internet Protocol): có chức năng gán địa chỉ cho dữ liệu trước khi truyền và định tuyến chúng tới đích.

+ ARP (Address Resolution Protocol): là giao thức phân giải địa chỉ IP thành địa chỉ MAC.

+ ICMP (Internet Control Message Protocol): có chức năng thông báo.

+ IGMP (Internet Group Management Protocol): có chức năng điều khiển truyền đa hướng (Multicast).

- Tầng vận chuyển (Transport layer) gồm 2 giao thức chính: TCP và UDP.

+ TCP (Tranmission Control Protocol): là giao thức cung cấp dịch vụ vận chuyển tin cậy bằng cách sử dụng cơ chế báo nhận, hướng nối kết theo kiểu truyền thông tin bằng cách phân luồng các byte. Bên cạnh đó, TCP còn là giao thức truyền song công, hỗ trợ cơ chế đa hợp.

+ UDP (User Datagram Protocol): là giao thức cung cấp dịch vụ vận chuyển không nối kết không tin cậy.

- Tầng ứng dụng (Application layer): gồm nhiều giao thức ứng dụng như Tenet, FTP, SMTP, POP, IMAP, HTTP, DNS, SMNP…

## **1.3. Giới thiệu về lập trình socket**

### **1.3.1. Khái niệm về socket**

Xét về góc độ mạng, Socket là 1 trong 2 điểm cuối của đường nối kết 2 chiều giữa 2 chương trình thực thi trên mạng.

Xét ở góc độ người lập trình, Socket là giao diện lập trình ứng dụng (API) hay bộ thư viện hàm hỗ trợ, dùng để nối kết chương trình ứng dụng với lớp mạng trong hệ thống mạng TCP/IP.

Socket được giới thiệu lần đầu dưới hệ điều hành UNIX version 4.3 BSD.

### **1.3.2. Cơ chế giao tiếp**

Trong cơ chế giao tiếp thông qua Socket, một trong hai quá trình phải công bố số hiệu cổng của Socket mà mình sử dụng để nhận và gởi dữ liệu.

Các quá trình khác có thể giao tiếp với quá trình đã công bố cổng cũng bằng cách tạo ra một Socket.

Trong Java, lớp Socket và ServerSocket hỗ trợ việc lập trình mạng giao tiếp giữa máy Client và Server.

Các bước cần có để tạo kết nối ở phía Client:

- Tạo một Socket để kết nối với Server.

- Định nghĩa các dòng nhập/xuất với Socket.

- Sử dụng các dòng nhập xuất để nhận/gởi dữ liệu.

- Đóng các dòng nhập/xuất và Socket khi không còn trao đổi.

Các bước cần có để tạo kết nối ở phía Server:

- Tạo ServerSocket lắng nghe ỏ một nào đó.

- Tạo một Socket accept nối kết với Client.

- Kết nối các dòng nhập/xuất với Socket.

- Sử dụng các dòng nhập xuất để nhận/gởi dữ liệu.

- Đóng các dòng nhập/xuất, Socket, ServerSocket.

### **1.3.3. Các chế độ giao tiếp**

Trong lập trình Socket, có 2 chế độ giao tiếp, đó là TCP (Transmission Control Protocol) và UDP(User Datagram Protocol).

## **1.4. Lập trình luồng (Thread)**

Trong hệ điều hành hiện đại, đơn vị sử dụng CPU là luồng (thread). Luồng cung cấp cơ chế tính toán và xử lý song song trong các ứng dụng. Mỗi luồng có riêng bộ đếm chương trình, trạng thái thanh ghi và ngăn xếp. Luồng sẽ chia sẻ không gian địa chỉ, do đó dễ dàng chia sẻ file, hiệu báo.

Trong ngôn ngữ lập trình Java, hỗ trợ cơ chế lập trình đa luồng bởi việc thừa kế lớp java.lang.Thread hoặc cài đặt lớp Runnable.

## **1.5. Giới thiệu về cờ tướng**

Cờ tướng là loại cờ có từ khoảng thế kỷ VII. Cờ tướng được bắt nguồn từ Saturanga, một loại cờ cổ được phát minh ở Ấn Độ từ thế kỷ V đến thế kỷ VI (trước cờ tướng khoảng 200 năm). Saturanga được phát minh từ Ấn Độ, sau đó dần phát triển và phân hóa thành 2 loại cờ khác nhau. Đó là cờ vua khi đi về phía Tây (Châu Âu) và cờ tướng khi đi về phía Đông (Trung Quốc).

Bàn cờ là một hình chữ nhật do 9 đường dọc và 10 đường ngang cắt nhau vuông góc tại 90 điểm hợp thành. Một khoảng trống gọi là sông nằm ngang giữa bàn cờ, chia bàn cờ thành hai phần đối xứng bằng nhau. Mỗi bên có một cung Tướng hình vuông do 4 ô hợp thành tại các đường dọc 4, 5, 6 kể từ đường ngang cuối của mỗi bên, trong 4 ô này có vẽ hai đường chéo xuyên qua.

Mỗi ván cờ lúc bắt đầu phải có đủ 32 quân, chia đều cho mỗi bên gồm 16 quân Trắng (Đỏ) và 16 quân Đen (Xanh). Quân cờ gồm 7 loại quân: Tướng, Sỹ, Tượng, Xe, Pháo, Mã, Tốt. Mỗi bên có thể viết khác nhau theo chữ hán nhưng giá trị và cách đi thì hoàn toàn giống nhau.

# **Chương 2. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH**

## **2.1. Server**

Server chịu trách nhiệm xử lý thông điệp gửi từ Client, tính toán và lưu trữ cơ sở dữ liệu. Một cách cụ thể:

- Database Server: lưu trữ cơ sở dữ liệu của người dùng bao gồm thông tin đăng ký, đăng nhập, thông tin người chơi như điểm, cấp bậc, bàn cờ. - Game Server: đóng vai trò là nơi trung gian để giao tiếp giữa Client với nhau, quản lý kết nối, đồng bộ thời gian, nhận thông tin từ Client, tính toán nước đi hợp lệ, thông tin thắng thua và gửi thông tin đến Client. Game Server sẽ áp dụng kĩ thuật lập trình song song MPI để tính toán giải thuật thắng thua, kiểm tra chiếu tướng.

- MPI Node là nơi giao tiếp với Game Server để tính toán giải thuật thắng thua, kiểm tra chiếu tướng.

## **2.2. Client**

Client đóng vai trò là người chơi. Cụ thể Client ở đây là sử dụng điện thoại có sử dụng hệ điều hành Android.

- Client có thể đăng ký, đăng nhập và tiến hành chơi cờ.

- Nhận thông tin cập nhật người chơi đang online được gửi từ Server.

- Gửi thông tin nước đi cho Game Server.

- Chờ nhận thông tin gửi từ Game Server và cập nhật lại đường đi cũng như giao diện đồ họa.

- Nhận kết quả thắng thua và hiển thị kết quả cho người chơi.

## **2.3. Xây dựng bàn cờ**

Áp dụng giải thuật, bàn cờ sẽ được chuyển sang ma trận 9x10. Mỗi quân cờ sẽ có một giá trị khác nhau trên bàn cờ, giá trị của quân cờ chính là giá trị của các phần tử trong ma trận.

|  |  |  |
| --- | --- | --- |
| **Tên quân cờ** | **Client 1** | **Client 2** |
| Tướng | 5 | -5 |
| Sỹ | 4 | -4 |
| Tượng | 3 | -3 |
| Mã | 2 | -2 |
| Xe | 1 | -1 |
| Pháo | 6 | -6 |
| Tốt | 7 | -7 |

Vị trí nào trên bàn cờ không có quân cờ sẽ mang giá trị 0. Ví dụ, bàn cờ tướng lúc đầu sẽ được chuyển sang ma trận A[9][10] như sau:

1 2 3 4 5 4 3 2 1

0 0 0 0 0 0 0 0 0

0 6 0 0 0 0 0 6 0

7 0 7 0 7 0 7 0 7

0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0

- 7 0 - 7 0 - 7 0 - 7 0 - 7

0 - 6 0 0 0 0 0 -6 0

0 0 0 0 0 0 0 0 0

- 1 - 2 - 3 - 4 - 5 - 4 - 3 - 2 - 1

## **2.4. Giải thuật**

Đang phát triển